K.S.INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COURSE FILE

NAME OF THE STAFF	:	**Mrs.Vaneeta M**
SUBJECT CODE/NAME	:	**18CS43/ Operating Systems**
SEMESTER/YEAR	:	**IV/II**
ACADEMIC YEAR	:	**2020-21**
BRANCH	:	**Computer Science and Engineering**

FACULTY IN-CHARGE

HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

Course file Contents - Check List

Sl. No.	Particulars
1	Vision, Mission of Institute and Department
2	PEO's, PSO's and PO's
3	CO PO PSO Mapping
4	Calendar of Events of Department & College
5	Student Details
6	Individual and class Time Table
7	Syllabus
8	Lesson Plan
9	Assignment Questions with Scheme
10	IA question Paper with Scheme (both sets)
11	All IA marks and final AVG marks
12	Pedagogy Report and Proofs (Proof of usage of ICT Tools)
13	Question Bank for each Module
14	Previous year VTU Question papers, Scheme for evaluation
15	Course end Survey
16	VTU Results
17	CO PO attainment
18	Teaching Learning Resources (Ex: Notes, PPT etc.)

K.S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Vision of the Institute

To impart quality technical education with ethical values, employable skills and research to achieve excellence

Mission of the Institute

- To attract and retain highly qualified, experienced & committed faculty.
- To create relevant infrastructure.
- Network with industry & premier institutions to encourage emergence of new ideas by providing research & development facilities to strive for academic excellence.
- To inculcate the professional & ethical values among young students with employable skills & knowledge acquired to transform the society.

Vision of the Department

To create competent professionals in Computer Science and Engineering with adequate skills to drive the IT industry

Mission of the Department

- Impart sound technical knowledge and quest for continuous learning.
- To equip students to furnish Computer Applications for the society through experiential learning and research with professional ethics.
- Encourage team work through inter-disciplinary project and evolve as leaders with social concerns.

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Program Educational Objectives

PEO1: Excel in professional career by acquiring knowledge in cutting edge technology and contribute to the society as an excellent employee or as an entrepreneur in the field of Computer Science & Engineering.

PEO2: Continuously enhance their knowledge on par with the development in IT industry and pursue higher studies in Computer Science & Engineering.

PEO3: Exhibit professionalism, cultural awareness, team work, ethics, and effective communication skills with their knowledge in solving social and environmental problems by applying computer technology.

Program Specific Outcomes (PSO)

PSO1: Ability to understand, analyze problems and implement solutions in programming languages, as well to apply concepts in core areas of Computer Science in association with professional bodies and clubs.

PSO2: Ability to use computational skills and apply software knowledge to develop effective solutions and data to address real world challenges.

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru-560 109

K.S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Program Outcomes

PO1: **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K. S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course: Operating Systems			
Type: Core	**Course Code:18CS43**	**Academic Year :2020-2021**	
Faculty : Mrs. Vaneeta M		**Sem/Sec: IV A**	

No of Hours per week

Theory (Lecture Class)	Practical/Field Work/Allied Activities	Total/Week	Total teaching hours
3	0	3	40

Marks

Internal Assessment	Examination	Total	Credits
40	60	100	3

Aim/Objective of the Course:

1. To describe services operating system provides to users, processes and other systems.
2. To describe the various features of processes, including scheduling, creation and termination, and communication.
3. To describe how operating system manages all the resources efficiently.

Course Learning Outcomes:

After completing the course, the students will be able to understand the need of operating system, different types of operating systems and apply suitable technique for management of different resources.

18CS43.1	Identify the need and services of various types of Operating Systems.	Applying (K3)
18CS43.2	Apply suitable techniques for process scheduling, synchronization and thread management.	Applying (K3)
18CS43.3	Make use of deadlock and memory management schemes for managing the operating Systems.	Applying (K3)
18CS43.4	Determining the need of demand paging, file and directory management.	Applying (K3)
18CS43.5	Apply suitable technique for disk scheduling and protection in operating system.	Applying (K3)

Syllabus Content:

MODULE 1: Introduction to operating systems, System structures, Process Management : Introduction to operating systems, System structures : What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. **Operating System Structures**: OS Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and	CO1 10hrs PO1-2 PO2-2 PO3-1 PSO1-1 PSO2-1

implementation; Operating System structure; Virtual machines; Operating System generation; System boot. **Process Management** Process concept; Process scheduling; Operations on processes; Inter process communication LO: At the end of this session the student will be able to, 1. Identify the need of Operating system 2. Understand the services provided by OS to user and processes 3. Discuss basics process concepts	

MODULE 2: Multi-threaded Programming, Process Scheduling, Process Synchronization **Multi-threaded Programming:** Overview; Multithreading models; Thread Libraries; Threading issues. **Process Scheduling**: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization:** Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. LO: At the end of this session the student will be able to, 1. Classify the different multithreading models 2. Classify various process scheduling algorithms 3. Understand methods of process synchronization	CO2 10hrs. PO1-3 PO2-3 PO3-2 PO5-2 PO9-2 PSO1-2 PSO2-2
MODULE 3: Deadlocks, Memory Management **Deadlocks** : Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management**: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. LO: At the end of this session the student will be able to, 1. Describe the methods for handling deadlocks 2. Explain memory management strategies 3. Distinguish between paging and segmentation.	CO3 10hrs PO1-3 PO2-3 PO3-2 PSO1-2 PSO2-2
MODULE 4: Virtual Memory Management, File System And Implementation of File System : Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **File System, Implementation of File System: File system:** File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management. LO: At the end of this session the student will be able to, 1. Understand the benefits of virtual memory management. 2. Study the file and directory system structure, implementation and protection	CO4 10hrs PO1-3 PO2-3 PO3-2 PSO1-2 PSO2-2

3. Understand the different allocation and free space management.	
MODULE 5: Secondary Storage Structures, Protection, Case Study: The Linux Operating System **Secondary Storage Structures, Protection**: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. **Protection**: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study: The Linux Operating System**: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter-process communication. LO: At the end of this session the student will be able to, 1. Learn the secondary storage structure and scheduling. 2. Understand the goals and principles of protection. 3. Understand the design principles and management of Linux operating system	CO5 10hrs PO1-3 PO2-3 PO3-2 PSO1-2 PSO2-2

Text Books: - (specify minimum two foreign authors text books)

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006.

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGrawHill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson

Useful Websites

- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-033-computer-system-engineering-spring-2009
- http://nptel.ac.in/courses/106108101
- https://www.elsevier.com/books/operating-systems/tsichritzis/978-0-12-701750-1

Useful Journals

- http://stmjournals.com/Journal-of-Operating-Systems-development-and-Trends.html

Teaching and Learning Methods:

1. Lecture class: 58 hrs.
2. Self-study: -
3. Field visits/Group Discussions/Seminars: 1hr.
4. Practical classes: Nil

Assessment:

Type of test/examination: Written examination

Continuous Internal Evaluation(CIE) : 20 marks (Average of best two of total three tests will be considered)

Semester End Exam(SEE) : 80 marks (students have to answer all main questions)

Test duration: 1 :30 hr

Examination duration: 3 hrs

CO to PO Mapping

PO1: Science and engineering Knowledge	**PO7:**Environment and Society **PO8:**Ethics

PO2: Problem Analysis **PO3:** Design & Development **PO4:**Investigations of Complex Problems **PO5:** Modern Tool Usage **PO6:** Engineer & Society	**PO9:**Individual & Team Work **PO10:** Communication **PO11:**Project Mngmt & Finance **PO12:**Life long Learning

PSO1: Ability to understand, analyze problems and implement solutions in programming languages, as well to apply concepts in core areas of Computer Science in association with professional bodies and clubs.

PSO2: Ability to use computational skills and apply software knowledge to develop effective solutions and data to address real world challenges.

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
18CS43.1	2	2	1	-	-	-	-	-	-	-	--	-
18CS43.2	3	3	2	-	2	-	-	-	2	-	-	-
18CS43.3	3	3	2	-	-	-	-	-	-	-	-	-
18CS43.4	3	3	2	-	-	-	-	-	-	-	-	-
18CS43.5	3	3	2	-	-	-	-	-	-	-	-	-
18CS43	2.8	2.8	1.8	-	2	-	-	-	2	-	-	-

CO	PSO1	PSO2
18CS43.1	1	1
18CS43.2	2	2
18CS43.3	2	2
18CS43.4	2	2
18CS43.5	2	2
18CS43	1.8	1.8

3	Substantial (High) Correlation
2	Moderate (Medium) Correlation
1	Slight (Low) Correlation
-	No correlation.

Course In-charge **Module Coordinator** **HOD**

Head of the Depart...
Dept. of Computer Sci...
K.S. Institute of Te...
Bengaluru -56...

K. S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course: Operating Systems			
Type: Core	**Course Code:** 18CS43	**Academic Year :2020-2021**	
Faculty : Mrs. Vaneeta M		**Sem/Sec: IV B**	

No of Hours per week

Theory (Lecture Class)	Practical/Field Work/Allied Activities	Total/Week	Total teaching hours
3	0	3	40

Marks

Internal Assessment	Examination	Total	Credits
40	60	100	3

Aim/Objective of the Course:

To describe services operating system provides to users, processes and other systems. To describe the various features of processes, including scheduling, creation and termination, and communication. To describe how operating system manages all the resources efficiently.

Course Learning Outcomes:

After completing the course, the students will be able to understand the need of operating system, different types of operating systems and apply suitable technique for management of different resources.

18CS43.1	Identify the need and services of various types of Operating Systems.	Applying (K3)
18CS43.2	Apply suitable techniques for process scheduling, synchronization and thread management.	Applying (K3)
18CS43.3	Make use of deadlock and memory management schemes for managing the operating Systems.	Applying (K3)
18CS43.4	Determining the need of demand paging, file and directory management.	Applying (K3)
18CS43.5	Apply suitable technique for disk scheduling and protection in operating system.	Applying (K3)

Syllabus Content:

MODULE 1: Introduction to operating systems, System structures, Process Management : Introduction to operating systems, System structures : What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. **Operating System Structures**: OS Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System	CO1 10hrs PO1-2 PO2-2 PO3-1 PSO1-1 PSO2-1

generation; System boot. **Process Management** Process concept; Process scheduling; Operations on processes; Inter process communication LO: At the end of this session the student will be able to, 1. Identify the need of Operating system 2. Understand the services provided by OS to user and processes 3. Discuss basics process concepts	

MODULE 2: Multi-threaded Programming, Process Scheduling, Process Synchronization **Multi-threaded Programming:** Overview; Multithreading models; Thread Libraries; Threading issues. **Process Scheduling**: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization:** Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. LO: At the end of this session the student will be able to, 1. Classify the different multithreading models 2. Classify various process scheduling algorithms 3. Understand methods of process synchronization	CO2 10hrs. PO1-3 PO2-3 PO3-2 PO5-2 PO9-2 PSO1-2 PSO2-2
MODULE 3: Deadlocks, Memory Management **Deadlocks** : Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management**: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. LO: At the end of this session the student will be able to, 1. Describe the methods for handling deadlocks 2. Explain memory management strategies 3. Distinguish between paging and segmentation.	CO3 10hrs PO1-3 PO2-3 PO3-2 PSO1-2 PSO2-2
MODULE 4: Virtual Memory Management, File System And Implementation of File System : Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **File System, Implementation of File System: File system:** File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management. LO: At the end of this session the student will be able to, 1. Understand the benefits of virtual memory management. 2. Study the file and directory system structure, implementation and protection 3. Understand the different allocation and free space management.	CO4 10hrs PO1-3 PO2-3 PO3-2 PSO1-2 PSO2-2
MODULE 5: Secondary Storage Structures, Protection, Case Study: The Linux Operating System **Secondary Storage Structures, Protection**: Mass storage structures; Disk structure;	CO5

Disk attachment; Disk scheduling; Disk management; Swap space management. **Protection**: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study: The Linux Operating System**: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter-process communication. LO: At the end of this session the student will be able to, 1. Learn the secondary storage structure and scheduling. 2. Understand the goals and principles of protection. 3. Understand the design principles and management of Linux operating system	10hrs PO1-3 PO2-3 PO3-2 PSO1-2 PSO2-2

Text Books: - (specify minimum two foreign authors text books)

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006.

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGrawHill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson

Useful Websites

- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-033-computer-system-engineering-spring-2009
- http://nptel.ac.in/courses/106108101
- https://www.elsevier.com/books/operating-systems/tsichritzis/978-0-12-701750-1

Useful Journals

- http://stmjournals.com/Journal-of-Operating-Systems-development-and-Trends.html

Teaching and Learning Methods:

1. Lecture class: 60 hrs.
2. Self-study: -
3. Field visits/Group Discussions/Seminars: 1hr.
4. Practical classes: Nil

Assessment:

Type of test/examination: Written examination

Continuous Internal Evaluation (CIE) : 20 marks (Average of best two of total three tests will be considered)

Semester End Exam (SEE) : 80 marks (students have to answer all main questions)

Test duration: 1 :30 hr

Examination duration: 3 hrs

CO to PO Mapping

PO1: Science and engineering Knowledge **PO2:** Problem Analysis **PO3:** Design & Development	**PO7:**Environment and Society **PO8:**Ethics **PO9:**Individual & Team Work **PO10:** Communication

PO4: Investigations of Complex Problems **PO5:** Modern Tool Usage **PO6:** Engineer & Society	**PO11:**Project Mngmt & Finance **PO12:**Life long Learning

PSO1: Ability to understand, analyze problems and implement solutions in programming languages, as well to apply concepts in core areas of Computer Science in association with professional bodies and clubs.

PSO2: Ability to use computational skills and apply software knowledge to develop effective solutions and data to address real world challenges.

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
18CS43.1	2	2	1	-	-	-	-	-	-	-	--	-
18CS43.2	3	3	2	-	2	-	-	-	2	-	-	-
18CS43.3	3	3	2	-	-	-	-	-	-	-	-	-
18CS43.4	3	3	2	-	-	-	-	-	-	-	-	-
18CS43.5	3	3	2	-	-	-	-	-	-	-	-	-
18CS43	2.8	2.8	1.8	-	2	-	-	-	2	-	-	-

CO	PSO1	PSO2
18CS43.1	1	1
18CS43.2	2	2
18CS43.3	2	2
18CS43.4	2	2
18CS43.5	2	2
18CS43	1.8	1.8

3	Substantial (High) Correlation
2	Moderate (Medium) Correlation
1	Slight (Low) Correlation
-	No correlation.

Course incharge

Module Coordinator

HOD

Head of the Department
Dept. of Computer Science & Eng
K.S. Institute of Technology
Bengaluru -560 109

K. S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

TENTATIVE CALENDAR OF EVENTS: EVEN SEMESTER (2020-2021)

SESSION: APR 2021 - AUG2021

Week No.	Month	Day						Days	Activities
		Mon	Tue	Wed	Thu	Fri	Sat		
1	APR	19 *	20	21	22	23	24	6	19*-Commencement of Higher Semester 24 Wednersday Time Table
2	APR/MAY	26	27	28	29	30	1H	5	1 May Day
3	MAY	3	4	5	6	7	8	6	8 Monday Time Table
4	MAY	10	11	12	13H	14H	15DH	3	13 Idul Fitr 14Basava Jayanti
5	MAY	17	18	19	20	21	22TA	6	22 Tuesday Time Table
6	MAY	24 T1	25T1	26T1	27	28	29DH	5	
7	MAY/JUN	31	1	2	3	4	5ASD	6	5 Wednersday Tme Table
8	JUN	7	8	9	10	11	12DH	5	
9	JUN	14	15	16	17	18	19	6	19 Monday Time Table
10	JUN	21	22	23	24	25TA	26DH	5	
11	JUN/JUL	28 T2	29T2	30T2	1	2	3	6	3 Thursday time Table
12	JUL	5	6	7	8	9ASD	10DH	5	
13	JUL	12	13	14	15	16	17	6	17 Tuesday Time Table
14	JUL	19	20	21H	22	23	24DH	4	21 Bakrid / Eid al Adha
15	JUL	26	27	28TA	29 T3	30T3	31T3	6	
16	AUG	2LT	3LT	4LT	5LT	6	7*ASD	6	7 Wednersday Tme Table 7* Last working day
Total No of Working Days : 86									

Total Number of working days (Excluding holidays and Tests)=73

H	Holiday
T1,T2, T3	Tests 1,2, 3
ASD	Attendance & Sessional Display
DH	Declared Holiday
LT	Lab Test
TA	Test attendance

Monday	15
Tuesday	15
Wednesday	15
Thursday	14
Friday	14
Total	**73**

PRINCIPAL
K.S. INSTITUTE OF TECHNOLOGY
BENGALURU - 560 109

K. S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING -CALENDAR OF EVENTS: EVEN SEMESTER (2020-2021)

SESSION: APR 2021 - AUG2021

Week No.	Month	Day						Days	Activities	Department Activities
		Mon	Tue	Wed	Thu	Fri	Sat			
1	APR	19 *	20	21	22	23	24	6	19*-Commencement of Higher Semester 24 Wednersday Time Table	
2	APR/MAY	26	27	28	29	30	1H	5	1 May Day	
3	MAY	3	4	5	6	7	8	6	8 Monday Time Table	
4	MAY	10	11	12	13H	14H	15DH	3	13 Idul Fitr 14 Basava Jayanti	Internship Opportunities: 12-5-21
5	MAY	17	18	19	20	21	22TA	6	22 Tuesday Time Table	Internship talk on skill Development:18-05-2021
6	MAY	24 T1	25T1	26T1	27	28	29DH	5		Project II phase Review 1 presentation:21-5-21,22-5-21,Internship awareness program:28-5-21
7	MAY/JUN	31	1	2	3	4	5ASD	6	5 Wednersday Tme Table	"Artificial Intelligence & Machine Learning": 3-6-21.
8	JUN	7	8	9	10	11	12DH	5		
9	JUN	14	15	16	17	18	19	6	19 Monday Time Table	
10	JUN	21	22	23	24	25TA	26DH	5		Project II phase Review 2 presentation:21-6-21,24-6-21,25-6-21,26-6-21
11	JUN/JUL	28 T2	29T2	30T2	1	2	3	6	3 Thursday time Table	
12	JUL	5	6	7	8	9ASD	10DH	5		
13	JUL	12	13	14	15	16	17	6	17 Tuesday Time Table	Project Phase II Review 3 Presentation:16-7-21, 17-7-21, Webinar on Design Innovation for Successful Career in the field of Aviation :17-7-21
14	JUL	19 T3(VIII)	20 * T3(VIII)	21H	22	23	24DH	4	20 *VIII Sem Last working day 21 Bakrid / Eid al Adha	
15	JUL	26	27	28TA	29	30	31	6		
16	AUG	2	3	4	5 T3	6 T3	7*T3	6	7 Wednersday Tme Table 7* IV &VI Last working day	
Total No of Working Days : 86										

Total Number of working days (Excluding holidays and Tests)=73

H	Holiday
T1,T2, T3	Tests 1,2, 3
ASD	Attendance & Sessional Display
DH	Declared Holiday
LT	Lab Test
TA	Test attendance

Monday	15
Tuesday	15
Wednesday	15
Thursday	14
Friday	14
Total	73

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru -560 109

K. .S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IV A STUDENT DETAILS

Sl. No	USN	Student Name	Email ID	Mobile Number
1	1KS19CS001	AAKRITI	aaki96798nov@gmail.com	8102100946
2	1KS19CS002	ABHISHEK B	abhibgowda2001@gmail.com	8296520760
3	1KS19CS003	ABHISHEK YADAV	abhishekyaadav1998@gmail.com	6394788936
4	1KS19CS004	ADITH KARTHIK RAJU	adithkarthik10@gmail.com	9164875273
5	1KS19CS005	AJAY S KALBURGI	kalburgiajay@gmail.com	7090663930
6	1KS19CS006	AKASH A S	akashlisp@gmail.com	8088125905
7	1KS19CS007	AMAN KUSHWAHA	amankushwaha365@gmail.com	8081239899
8	1KS19CS009	AMIT K B	baluamit@gmail.com	9483354792
9	1KS19CS010	AMOGHA H S	amogha.hs@gmail.com	9845006222
10	1KS19CS011	AMRUTHA K H	amruthaharshan8347@gmail.com	7204219550
11	1KS19CS012	ANAGHA A HEBBAR	anaghaah01@gmail.com	9148571180
12	1KS19CS014	ANUSHA B	vbabuvbabu222@gmail.com	9342582372
13	1KS19CS015	AQSA AQEEL	aqsa.aqeel17@gmail.com	8707566256
14	1KS19CS016	ASHIKA H N	ashikananjegowda869@gmail.com	7975007345
15	1KS19CS017	BHOOMIKA A M	mohan_iist@yahoo.co.in	9980150041
16	1KS19CS018	BHOOMIKA K	bhoomikak17@gmail.com	9986598304
17	1KS19CS019	BHUMIKA M	ravikumar19721736@gmail.com	9964151736
18	1KS19CS020	C N SHREYAS	sheryascn611@gmail.com	6366306846
19	1KS19CS021	CHAITANYA SHIVARAJU	chaitanyashivaraju@gmail.com	8197730165
20	1KS19CS022	CHANDAN B V	chandanbv9123@gmail.com	7338568147
21	1KS19CS023	DEEKSHA NAIDU R	deeksha12082001@gmail.com	8095122655
22	1KS19CS024	DEEPA G	deepa1012gandhi@gmail.com	9148483322
23	1KS19CS025	DEEPTHI.N.K	deepthink400@gmail.com	9148487058
24	1KS19CS026	DEVI PRASAD N	deviprasad4556@gmail.com	7639444564
25	1KS19CS028	DHEEMANTH G	dheemanthgirish03@gmail.com	9108250916
26	1KS19CS029	DINESH M	denishdinu1431@gmail.coom	6366055811
27	1KS19CS030	G PRERITHA	prerithagowtham@gmail.com	8197458531
28	1KS19CS031	GAGAN REDDY S	gagan19200@gmail.com	8277884917
29	1KS19CS032	GAGANDEEP K	gagandeepk6549@gmail.com	7619129339
30	1KS19CS033	GEETHANJALI B K PRASAD	geethanjali9840@gmail.com	9632806412
31	1KS19CS034	GULSHAN KUMAR S	kumargulshan2702@gmail.com	8050991314
32	1KS19CS035	HARSHITHA J	harshithaj2306@gmail.com	6360781608
33	1KS19CS036	INDRAJITH H M	indrajithlikith@gmail.com	9972503057
34	1KS19CS038	K KISHAN	kishanuk2002@gmail.com	9686480814
35	1KS19CS039	K R VAGEESH	vageesh2001@gmail.com	9742578725
36	1KS19CS040	KALYAN CHOWDHARY B	munirathnam931@gmail.com	9008838499
37	1KS19CS041	KARTHIK S MORAJKAR	karthik.morajkar26@gmail.com	9113629181
38	1KS19CS042	KAVYASHREE.S.L	kavyashreesl599@gmail.com	9113274206
39	1KS19CS043	KEERTHAN GOWDA S	keerthangowdas222@gmail.com	9845871477
40	1KS19CS044	KOTHAPALLI SREEJA	sreeja16112002@gmail.com	9491465760
41	1KS19CS045	KOTTALA SAIVENKATASUCHITH	suchithkottala@gmail.com	9059089926
42	1KS19CS046	KRISHNA K R	krkrishna2626@gmail.com	7358478255
43	1KS19CS047	KUMAR S	chittihoney225@gmail.com	9591084850
44	1KS19CS048	LIKITH G	likhithgk01@gmail.com	9448116636
45	1KS19CS049	LISHA C	lishachoudhary2612@gmail.com	9036831198
46	1KS19CS050	MAHAK SHREE	shreemahak01@gmail.com	9973580148
47	1KS19CS051	MALLIPALLI SPURTHI REDDY	mspurthi49@gmail.com	9440090436
48	1KS19CS052	MANASA G L	manasagl@gmail.com	9901599887
49	1KS19CS053	MOHAMMED NOOR AMAN	mohammednooraman309@gmail.com	9880672867
50	1KS19CS054	MUKESH KUMAR	mukeshkeshri114@gmail.com	7991448599
51	1KS19CS055	MYTHREYI U	shobha.hande8@gmail.com	9741052966
52	1KS19CS056	N ASHOK	ashoknashok132000@gmail.com	8073952496
53	1KS19CS057	N BHAVYA	bhavyanarra56@gmail.com	8790762175
54	1KS19CS058	N P SHASHANKA RAO	shashanknp02@gmail.com	9606573173
55	1KS18CS011	BHARATH R	barathrahul100@gmail.com	7795844959
56	1KS19CS116	VRATHIKA BILLAVA	vrathi.billava@gmail.com	9620109468
57	1KS20CS400	AKIF DELVI	-	9845038115
58	1KS20CS404	PRANAV CHANDRAN P	chandanbv9123@gmail.com	9591810824
59	1KS20CS402	KEERTHI KUMAR V	keerthikumar6317@gmail.com	8317422454

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K. .S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IV B STUDENT DETAILS

Sl. No	USN	Student Name	Email ID	Mobile Number
1	1KS19CS059	NAVEEN K M	naveenkmacharya@gmail.com	8095009620
2	1KS19CS060	NEELAMMA SALI	neelammasali123@gmail.com	9740477266
3	1KS19CS061	NETHRA R	shiva012972@gmail.com	8296322369
4	1KS19CS062	NIKHIL CHOWDARY V R	nikhilchowdary2001@gmail.com	6361010810
5	1KS19CS063	NITHYA S N	nithyojetty116@gmail.com	9380484140
6	1KS19CS064	PAVAN B INDRESH	pavanindresh15@gmail.com	8884899481
7	1KS19CS065	PAVAN L	pavanlakkanna@gmail.com	9611361221
8	1KS19CS066	PAVITHRA S P	pavithrasppalli2000@gmail.com	9632688308
9	1KS19CS067	POOJA J	poojajairaj2002@gmail.com	9916103120
10	1KS19CS068	POTHURU SAI PRIYANKA	saipriyankapothuru0129@gmail.com	9014533171
11	1KS19CS069	PRAJWAL G	ggp.prasad@gmail.com	8088091887
12	1KS19CS070	PRAJWAL KULKARNI	parashuramk237@gmail.com	7899924738
13	1KS19CS071	PREETHI K P	preethikp189@gmail.com	6366013491
14	1KS19CS072	PRIYA E	priya.e0704@gmail.com	9901588915
15	1KS19CS073	PRIYANKA R	priyanka4ac@gmail.com	8088158233
16	1KS19CS074	PUVAN KUMAR V	puvanvenkatraman@gmail.com	9008347759
17	1KS19CS075	R V YASHVANTH	yasvanth27502@gmail.com	6364427502
18	1KS19CS076	RAJ KUMAR	rajmishra8651@gmail.com	8709895679
19	1KS19CS077	RAKESH M J	rakeshrana06@gmail.com	9449458715
20	1KS19CS078	RAKSHA T M	anithanmgowda1212@gmail.com	6362511050
21	1KS19CS079	RANJITHA D V	ranjushetty831@gmail.com	8317398272
22	1KS19CS080	SAHANA S	sahanapranu143@gmail.com	9380860544
23	1KS19CS081	SAI SHIRISHA S B	shirishabharathi1786@gmail.com	9731285109
24	1KS19CS082	SANGHARSH KUMAR RAI	sangharshrai1@gmail.com	7239026893
25	1KS19CS083	SANJEEV MYSORE	sanjeevmysore1@gmail.com	9632612823
26	1KS19CS084	SANKALP KESTI	sankalpkesti1008@gmail.com	8431687833
27	1KS19CS085	SATISH V	sy4529630@gmail.com	6360959158
28	1KS19CS086	SHANVI B P	shanviprakash266@gmail.com	8277448566
29	1KS19CS087	SHIVANI G K	shivanigksonu10@gmail.com	8762001512
30	1KS19CS088	SHREESHA S	shreesha.082@gmail.com	8549010769
31	1KS19CS089	SHREYA R KIRAN	shreyakiran551@gmail.com	9480613920
32	1KS19CS090	SHRI HARSHA S	shriharsha40@gmail.com	8431282887
33	1KS19CS091	SINDHU S	sindhuhiremath3@gmail.com	9380938743
34	1KS19CS092	SMRITHI SHEKAR	smrithishekar1220@gmail.com	9110664900
35	1KS19CS093	SRUJAN K	ksrujan2001@gmail.com	9480792363
36	1KS19CS094	SRUSTI S GOWDA	srusti2706@gmail.com	9945832865

37	1KS19CS096	SUHAS S	suhassairam0@gmail.com	7795534529
38	1KS19CS097	SUSHMITHA K	sushvinutha740@gmail.com	7406030664
39	1KS19CS098	SWETHA M KULKARNI	smk10032002@gmail.com	9353799412
40	1KS19CS099	SYED ZAINUL ABIDIN	szahere619@gmail.com	7795524743
41	1KS19CS100	TALLURU MAURYA	tallurimaurya0@gmail.com	9845888399
42	1KS19CS101	TANUSHREE R	tanugowda1971@gmail.com	8861756048
43	1KS19CS102	TEJAS N	tejasnreddy14@gmail.com	9663150926
44	1KS19CS103	TEJAS P	tejasp0401@gmail.com	8095337382
45	1KS19CS104	TEJASWINI NAYAKA S	tejaswinis2901@gmail.com	8660869256
46	1KS19CS105	THANUSHREE S	siddarajutanushree@gmail.com	7406002195
47	1KS19CS106	THIRUMAL R	thirumal2000@gmail.com	8088236701
48	1KS19CS107	THRIVENI U	thriveeniu@gmail.com	6363356653
49	1KS19CS108	UDHAY KUMAR G	uk382002@gmail.com	9632883339
50	1KS19CS109	VAISHNAVI G	gvaishnavireddy23@gmail.com	9019096739
51	1KS19CS110	VARSHA BAI R	varshabair2001@gmail.com	9480050450
52	1KS19CS111	VARUN KAMBALI	kambalivarun@gmail.com	7975459220
53	1KS19CS112	VISHAL GUPTA	vishalguptaforever@gmail.com	9625853132
54	1KS19CS113	YASHWANTH S R	sryashwanth2001@gmail.com	8088227342
55	1KS18CS021	G.SARAYU CHOWDARY	gsarayu2014@gmail.com	9972693883
56	1KS18CS068	PRAJWAL.N	prajwal.n.12@gmail.com	9845709915
57	1KS19CS115	AKSHAY R	akshayrgowda2001@gmail.com	9066670733
58	1KS20CS403	NITHIN S	nithingowda474@gmail.com	7760397469
59	1KS20CS401	ANUSHA A R	anushaar86693@gmail.com	7349581484

Mary

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU-109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIVIDUAL ONLINE TIME TABLE FOR THE YEAR 2020-2021 (EVEN SEMESTER)

W.E.F: 19/04/2021

NAME OF THE FACULTY: Mrs. VANEETA.M **DESIGNATION: ASSOC. PROF.**

PERIOD	1	2	11.00 AM-11.15 AM	3	4	1.15 PM-1.45 PM	5	6
TIME / DAY	9:00 AM-10:00 AM	10.00 AM-11.00 AM		11.15 AM-12.15 PM	12.15 PM-1.15 PM		1.45 PM-2.45 PM	2.45 PM-3.45 PM
MON	OS (B) (18CS43)		TEA BREAK	OS (A) (18CS43)		LUNCH BREAK	PROJECT WORK PHASE II (17CSP85/15CSP85)	
TUE					OS (B) (18CS43)		TECHNICAL SEMINAR (17CSS86/ 15CSS86)	
WED		OS (B) (18CS43)		OS (A) (18CS43)			INTERNSHIP (17CS84/15CS84)	
THUR	OS (A) (18CS43)							
FRI				OS (B) (18CS43)			OS (A) (18CS43)	

TIME TABLE INCHARGE

17/4/21
HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU-109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IV SEMESTER ONLINE TIME TABLE FOR THE YEAR 2020-21 (EVEN SEMESTER)

W.E.F: 19-04-2021
SEC: 'A'

CLASS TEACHER: VANEETA. M
CLASS ROOM: CN LAB

PERIOD	1	2	11.00 AM-11.15 AM	3	4	1.15 PM-1.45 PM	5	6
TIME / DAY	9:00 AM-10:00 AM	10.00 AM-11.00 AM		11.15 AM-12.15 PM	12.15 PM-1.15 PM		1.45 PM-2.45 PM	2.45 PM-3.45 PM
MON	DC (18CS46)	MC&ES (18CS44)	TEA BREAK	OS (18CS43)	OOC (18CS45)	LUNCH BREAK	DAA LAB (18CSL47) (A1,A2 &A3)	
TUE	DAA (18CS42)	DC (18CS46)		OOC (18CS45)	CAP & SM (18MAT41)		MC&ES LAB (18CSL48) (A1,A2 &A3)	
WED	OS (18CS43)	OOC (18CS45)		DAA (18CS42)	DC (18CS46)		MC&ES (18CS44)	CAP & SM (18MAT41)
THUR	MC&ES (18CS44)	CAP & SM (18MAT41)		DAA (18CS42)	OS (18CS43)		OOC (18CS45)	DC (18CS46)
FRI	CIP (18CPH49)	DAA (18CS42)		MC&ES (18CS44)	CAP & SM (18MAT41)		OS (18CS43)	DIP MATHS

Subject Code	Subject Name	Faculty Name
18MAT41	COMPLEX ANALYSIS, PROBABILITY AND STATISTICAL METHODS	Dr. JALAJA. P
18CS42	DESIGN AND ANALYSIS OF ALGORITHMS	Mr. VENKATA RAO K
18CS43	OPERATING SYSTEMS	Mrs. VANEETA.M
18CS44	MICROCONTROLLER AND EMBEDDED SYSTEMS	Mr. SANJOY DAS
18CS45	OBJECT ORIENTED CONCEPTS	Mr. RAGHAVENDRACHAR.S
18CS46	DATA COMMUNICATION	Mr. KUSHAL KUMAR B.N
18CSL47	DESIGN AND ANALYSIS OF ALGORITHM LABORATORY	Mrs. VIJAYALAXMI. M & Mrs. RANJITHA.K.N
18CSL48	MICROCONTROLLER AND EMBEDDED SYSTEMS LABORATORY	Mr. SANJOY DAS & Mr. ROOPESH KUMAR . B.N
18CPH49	CONSTITUTION OF INDIA, PROFESSIONAL ETHICS AND HUMAN RIGHTS	Mrs. ANURADHA.M.V

B.N 17/4/21 S.Raghu 17/04/2021
TIME TABLE INCHARGE

17/4/21
HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

PRINCIPAL
PRINCIPAL
K.S. INSTITUTE OF TECHNOLOGY
BENGALURU - 560 109.

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU-109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IV SEMESTER ONLINE TIME TABLE FOR THE YEAR 2020-21 (EVEN SEMESTER)

W.E.F: 19-04-2021
SEC: 'B'

CLASS TEACHER: Mrs. VIJAYALAXMI .M
CLASS ROOM: DAA LAB

PERIOD	1	2	11.00 AM-11.15 AM	3	4	1.15 PM-1.45 PM	5	6
TIME / DAY	9:00 AM-10:00 AM	10.00 AM-11.00 AM		11.15 AM-12.15 PM	12.15 PM-1.15 PM		1.45 PM-2.45 PM	2.45 PM-3.45 PM
MON	OS (18CS43)	OOC (18CS45)	TEA BREAK	DC (18CS46)	DAA (18CS42)	LUNCH BREAK	CAP & SM (18MAT41)	MC&ES (18CS44)
TUE	OOC (18CS45)	MC&ES (18CS44)		DC (18CS46)	OS (18CS43)		DAA LAB (18CSL47) (B1,B2 & B3)	
WED	DAA (18CS42)	CIP (18CPH49)		CAP & SM (18MAT41)	OOC (18CS45)		DC (18CS46)	DAA (18CS42)
THUR	CAP & SM (18MAT41)	OS (18CS43)		OOC (18CS45)	MC&ES (18CS44)		MC&ES LAB (18CSL48) (B1,B2 &B3)	
FRI	MC&ES (18CS44)	DAA (18CS42)		OS (18CS43)	CAP & SM (18MAT41)		DC (18CS46)	DIP MATHS

Subject Code	Subject Name	Faculty Name
18MAT41	COMPLEX ANALYSIS, PROBABILITY AND STATISTICAL METHODS	Mr. VENKATARAMANA B S
18CS42	DESIGN AND ANALYSIS OF ALGORITHMS	Mrs. VIJAYALAXMI . M
18CS43	OPERATING SYSTEMS	Mrs. VANEETA.M
18CS44	MICROCONTROLLER AND EMBEDDED SYSTEMS	Mr. ROOPESH KUMAR .B.N
18CS45	OBJECT ORIENTED CONCEPTS	Mr. RAGHAVENDRACHAR.S
18CS46	DATA COMMUNICATION	Mr. KUSHAL KUMAR . B.N
18CSL47	DESIGN AND ANALYSIS OF ALGORITHM LABORATORY	Mrs. RANJITHA.K.N & Mrs. VIJAYALAXMI. M
18CSL48	MICROCONTROLLER AND EMBEDDED SYSTEMS LABORATORY	Mr. ROOPESH KUMAR.B.N & Mr. KRISHNA GUDI
18CPH49	CONSTITUTION OF INDIA, PROFESSIONAL ETHICS AND HUMAN RIGHTS	Mrs. ANURADHA.M.V

R.N 17/4/21 S.Rao 17/04/2021
TIME TABLE INCHARGE

17/4/21
HOD

Head of the Department
Dept of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

PRINCIPAL
PRINCIPAL
K.S. INSTITUTE OF TECHNOLOGY
BENGALURU - 560 109.

OPERATING SYSTEMS (Effective from the academic year 2018 -2019) SEMESTER – IV			
Course Code	**18CS43**	**CIE Marks**	40
Number of Contact Hours/Week	3:0:0	**SEE Marks**	60
Total Number of Contact Hours	40	**Exam Hours**	03
CREDITS –3			

Course Learning Objectives: This course (18CS43) will enable students to:

- Introduce concepts and terminology used in OS
- Explain threading and multithreaded systems
- Illustrate process synchronization and concept of Deadlock
- Introduce Memory and Virtual memory management, File system and storage techniques

Module 1	**Contact Hours**
Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. **Operating System Services;** User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot. **Process Management** Process concept; Process scheduling; Operations on processes; Inter process communication **Text book 1: Chapter 1, 2.1, 2.3, 2.4, 2.5, 2.6, 2.8, 2.9, 2.10, 3.1, 3.2, 3.3, 3.4** **RBT: L1, L2, L3**	08
Module 2	
Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization:** Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. **Text book 1: Chapter 4.1, 4.2, 4.3, 4.4, 5.1, 5.2, 5.3, 5.4, 5.5, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7** **RBT: L1, L2, L3**	08
Module 3	
Deadlocks : Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management:** Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. **Text book 1: Chapter 7, 8.1 to 8.6** **RBT: L1, L2, L3**	08
Module 4	
Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **File System, Implementation of File System:** File system: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management. **Text book 1: Chapter 91. To 9.6, 10.1 to 10.5** **RBT: L1, L2, L3**	08

Module 5	
Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study: The Linux Operating System:** Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter-process communication. **Text book 1: Chapter 12.1 to 12.6, 21.1 to 21.9** **RBT: L1, L2, L3**	08
Course Outcomes: The student will be able to :	
• Demonstrate need for OS and different types of OS • Apply suitable techniques for management of different resources • Use processor, memory, storage and file system commands • Realize the different concepts of OS in platform of usage through case studies	
Question Paper Pattern:	
• The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.	
Textbooks:	
1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006	
Reference Books:	
1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition 2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013. 3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014. 4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.	

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K S INSTITUTE OF TECHNOLOGY BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NAME OF THE STAFF :Mrs. VANEETA M

SUBJECT CODE/NAME : 18CS43/ OPERATING SYSTEMS

SEMESTER/YEAR : II / IV 'A'

ACADEMIC YEAR : 2020-2021

Sl. No.	Topic to be covered	Mode of Delivery	Teaching Aid	No. of Periods	Cumulative No. of Periods	Proposed Date
	MODULE 1					
1	What operating systems do; Computer System organization	L+D	PPT using MS Teams Platform	1	1	19-4-2021
2	Computer System architecture; Operating System structure	L+D	PPT using MS Teams Platform	1	2	21-4-2021
3	Operating System operations; Process management	L+ D	PPT using MS Teams Platform	1	3	22-4-2021
4	Memory management; Storage management; Protection and Security	L+ D	PPT using MS. Teams Platform	1	4	23-4-2021
5	Distributed system; Special-purpose systems; Computing environments.	L+D	PPT using MS Teams Platform	1	5	24-4-2021
6	Operating System Services; User - Operating System interface; System calls	L+D	PPT using MS Teams Platform	1	6	26-4-2021
7	Types of system calls; System programs	L+D	PPT using MS Teams Platform	1	7	28-4-2021
8	Operating system design and implementation; Operating System structure	L+D	PPT using MS Teams Platform	1	8	29-4-2021
9	Virtual machines; Operating System generation; System boot	L+D	PPT using MS Teams Platform	1	9	30-4-2021

10	Process Management Process concept	L+D	PPT using MS Teams Platform	1	10	3-5-2021
11	Process scheduling	L+D	PPT using MS Teams Platform	1	11	5-5-2021
12	Operations on processes	L+D	PPT using MS Teams Platform	1	12	6-5-2021
13	Inter process communication	L+D	PPT using MS Teams Platform	1	13	7-5-2021
MODULE 2						
14	Overview; Multithreading models;	L+ D	PPT using MS Teams Platform	1	14	10-5-2021
15	Thread Libraries; Threading issues.	L+D	PPT using MS Teams Platform	1	15	12-5-2021
16	Process Scheduling: Basic concepts; Scheduling Criteria	L+D	PPT using MS Teams Platform	1	16	17-5-2021
17	Scheduling Algorithms	L+D	PPT using MS Teams Platform	2	18	19-5-2021, 20-5-2021
18	Multiple-processor scheduling; Thread scheduling.	L+D	PPT using MS Teams Platform	1	19	21-5-2021
19	**Internal Test Assessment I**				20	24-5-2021
20	Process Synchronization: Synchronization: The critical section problem	L+D	PPT using MS Teams Platform	1	21	27-5-2021
21	Peterson's solution; Synchronization hardware	L+D	PPT using MS Teams Platform	1	22	28-5-2021
22	Semaphores	L+D	PPT using MS Teams Platform	1	23	31-5-2021
23	Classical problems of synchronization	L+D	PPT using MS Teams Platform	1	24	2-6-2021
24	Pedagogy	L+D	PPT using MS Teams Platform	1	25	3-6-2021
25	Monitors	L+D	PPT using MS Teams Platform	1	26	4-6-2021
	MODULE 3					
26	System model; Deadlock characterization;	L+D	PPT using MS Teams Platform	1	27	5-6-2021

27	Methods for handling deadlocks; Deadlock prevention	L+D	PPT using MS Teams Platform	1	28	7-6-2021
28	Deadlock avoidance	L+D	PPT using MS Teams Platform	2	30	9-6-2021, 10-6-2021
29	Deadlock detection and recovery from deadlock	L+D	PPT using MS Teams Platform	1	31	11-6-2021
30	Memory Management: Memory management strategies: Background	L+D	PPT using MS Teams Platform	1	32	14-6-2021
31	Swapping; Contiguous memory allocation;	L+D	PPT using MS Teams Platform	1	33	16-6-2021
32	Paging	L+D	PPT using MS Teams Platform	2	35	17-6-2021, 18-6-2021
33	Structure of page table	L+D	PPT using MS Teams Platform	1	36	19-6-2021
34	Structure of page table	L+D	PPT using MS Teams Platform	1	37	21-6-2021
35	Segmentation	L+D	PPT using MS Teams Platform	1	38	23-6-2021
	MODULE 4					
36	Background; Demand paging;	L+D	PPT using MS Teams Platform	2	40	24-6-2021, 25-6-2021
37	Copy-on-write	L+D	PPT using MS Teams Platform	1	41	1-7-2021
38	**Internal Test Assessment II**				42	28-6-2021
39	Page replacement	L+D	PPT using MS Teams Platform	2	44	2-7-2021, 3-7-2021
40	Allocation of frames; Thrashing	L+D	PPT using MS Teams Platform	1	45	5-7-2021
41	File System, Implementation of File System: File system: File concept	L+D	PPT using MS Teams Platform	1	46	7-7-2021
42	Access methods;	L+D	PPT using MS Teams Platform	1	47	8-7-2021
43	Directory structure	L+D	PPT using MS Teams Platform	1	48	9-7-2021
44	File system mounting; File sharing; Protection	L+D	PPT using MS Teams Platform	1	49	12-7-2021

45	Implementing File system: File system structure; File system implementation	L+D	PPT using MS Teams Platform	1	50	14-7-2021
46	Directory implementation; Allocation methods	L+D	PPT using MS Teams Platform	1	51	15-7-2021
47	Free space management	L+D	PPT using MS Teams Platform	1	52	16-7-2021
MODULE 5						
48	Mass storage structures; Disk structure; Disk attachment	L+D	PPT using MS Teams Platform	1	53	19-7-2021
49	Disk scheduling	L+D	PPT using MS Teams Platform	1	54	22-7-2021
50	Disk management; Swap space management	L+D	PPT using MS Teams Platform	1	55	23-7-2021
51	Protection: Goals of protection, Principles of protection, Domain of protection,	L+D	PPT using MS Teams Platform	1	56	26-7-2021
52	Access matrix, Implementation of access matrix, Access control,	L+D	PPT using MS Teams Platform	1	57	28-7-2021
53	Revocation of access rights, Capability- Based systems	L+D	PPT using MS Teams Platform	1	58	29-7-2021
54	Case Study: The Linux Operating System: Linux history; Design principles; Kernel modules	L+D	PPT using MS Teams Platform	1	59	30-7-2021
55	Process management; Scheduling; Memory Management	L+D	PPT using MS Teams Platform	1	60	2-8-2021
56	File systems, Input and output; Inter-process communication	L+D	PPT using MS Teams Platform	1	61	4-8-2021
57	**Internal Test Assessment III**				62	5-8-2021

Text Books: -

1 Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006.

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGrawHill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson

Web Materials

- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-033-computer-system-engineering-spring-2009
- http://nptel.ac.in/courses/106108101
- https://www.elsevier.com/books/operating-systems/tsichritzis/978-0-12-701750-1

Details of Teaching Aids:
Power point Presentation using Zoom Platform

Course incharge | **Module Coordinator** | **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K S INSTITUTE OF TECHNOLOGY BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NAME OF THE STAFF : Mrs.VANEETA M

SUBJECT CODE/NAME : 18CS43/ OPERATING SYSTEMS

SEMESTER/YEAR : II / IV 'B'

ACADEMIC YEAR : 2020-2021

Sl. No.	Topic to be covered	Mode of Delivery	Teaching Aid	No. of Periods	Cumulative No. of Periods	Proposed Date
	MODULE 1					
1	What operating systems do; Computer System organization	L+D	PPT using MS Teams Platform	1	1	19-4-2021
2	Computer System architecture; Operating System structure	L+D	PPT using MS Teams Platform	1	2	20-4-2021
3	Operating System operations; Process management	L+ D	PPT using MS Teams Platform	1	3	22-4-2021
4	Memory management; Storage management; Protection and Security	L+ D	PPT using MS Teams Platform	1	4	23-4-2021
5	Distributed system; Special-purpose systems; Computing environments.	L+D	PPT using MS Teams Platform	1	5	26-4-2021
6	Operating System Services; User - Operating System interface; System calls	L+D	PPT using MS Teams Platform	1	6	27-4-2021
7	Types of system calls; System programs	L+D	PPT using MS Teams Platform	1	7	29-4-2021
8	Operating system design and implementation; Operating System structure	L+D	PPT using MS Teams Platform	1	8	3-5-2021

9	Virtual machines; Operating System generation; System boot	L+D	PPT using MS Teams Platform	1	9	4-5-2021
10	Process Management Process concept	L+D	PPT using MS Teams Platform	1	10	6-5-2021
11	Process scheduling	L+D	PPT using MS Teams Platform	1	11	7-5-2021
12	Operations on processes	L+D	PPT using MS Teams Platform	1	12	8-5-2021
13	Inter process communication	L+D	PPT using MS Teams Platform	1	13	10-5-2021
	MODULE 2					
14	Overview; Multithreading models;	L+ D	PPT using MS Teams Platform	1	14	11-5-2021
15	Thread Libraries; Threading issues.	L+D	PPT using MS Teams Platform	1	15	17-5-2021
16	Process Scheduling: Basic concepts; Scheduling Criteria	L+D	PPT using MS Teams Platform	1	16	18-5-2021
17	Scheduling Algorithms	L+D	PPT using MS Teams Platform	2	18	20-5-2021
18	Multiple-processor scheduling; Thread scheduling.	L+D	PPT using MS Teams Platform	1	19	21-5-2021
19	Process Synchronization: Synchronization: The critical section problem		PPT using MS Teams Platform	1	20	22-5-2021
20	**Internal Test Assessment I**				21	24-5-2021
21	Process Synchronization: Synchronization: The critical section problem	L+D	PPT using MS Teams Platform	1	22	27-5-2021
22	Peterson's solution; Synchronization hardware	L+I	PPT using MS Teams Platform	1	23	28-5-2021
23	Semaphores	L+D	PPT using MS Teams Platform	1	24	31-5-2021
24	Classical problems of synchronization	L+D	PPT using MS Teams Platform	1	25	1-6-2021
25	Pedagogy	L+D	PPT using MS Teams Platform	1	26	3-6-2021
26	Monitors	L+D	PPT using MS Teams Platform	1	27	4-6-2021
	MODULE 3					

27	System model; Deadlock characterization;	L+D	PPT using MS Teams Platform	1	28	5-6-2021
28	Methods for handling deadlocks; Deadlock prevention	L+D	PPT using MS Teams Platform	1	29	7-6-2021
29	Deadlock avoidance	L+D	PPT using MS Teams Platform	2	31	8-6-2021, 10-6-2021
30	Deadlock detection and recovery from deadlock	L+D	PPT using MS Teams Platform	1	32	11-6-2021
31	Memory Management: Memory management strategies: Background	L+D	PPT using MS Teams Platform	1	33	14-6-2021
32	Swapping; Contiguous memory allocation;	L+D	PPT using MS Teams Platform	1	34	15-6-2021
33	Paging	L+D	PPT using MS Teams Platform	2	36	17-6-2021, 18-6-2021
34	Structure of page table	L+D	PPT using MS Teams Platform	1	37	19-6-2021
35	Structure of page table	L+D	PPT using MS Teams Platform	1	38	21-6-2021
36	Segmentation	L+D	PPT using MS Teams Platform	1	39	22-6-2021
	MODULE 4					
37	Background; Demand paging;	L+D	PPT using MS Teams Platform	2	41	24-6-2021, 25-6-2021
38	**Internal Test Assessment II**				42	28-6-2021
39	Copy-on-write	L+D	PPT using MS Teams Platform	1	43	2-7-2021
40	Page replacement	L+D	PPT using MS Teams Platform	2	45	3-7-2021, 5-7-2021
41	Allocation of frames; Thrashing	L+D	PPT using MS Teams Platform	1	46	6-7-2021
42	File System, Implementation of File System: File system: File concept	L+D	PPT using MS Teams Platform	1	47	8-7-2021
43	Access methods;	L+D	PPT using MS Teams Platform	1	48	9-7-2021
44	Directory structure	L+D	PPT using MS Teams Platform	1	49	12-7-2021

45	File system mounting; File sharing; Protection	L+D	PPT using MS Teams Platform	1	50	13-7-2021
46	Implementing File system: File system structure; File system implementation	L+D	PPT using MS Teams Platform	1	51	15-7-2021
47	Directory implementation; Allocation methods	L+D	PPT using MS Teams Platform	1	52	16-7-2021
48	Free space management	L+D	PPT using MS Teams Platform	1	53	17-7-2021
	MODULE 5					
49	Mass storage structures; Disk structure; Disk attachment	L+D	PPT using MS Teams Platform	1	54	19-7-2021
50	Disk scheduling	L+D	PPT using MS Teams Platform	1	55	20-7-2021
51	Disk management; Swap space management	L+D	PPT using MS Teams Platform	1	56	22-7-2021
52	Protection: Goals of protection, Principles of protection, Domain of protection,	L+D	PPT using MS Teams Platform	1	57	23-7-2021
53	Access matrix, Implementation of access matrix, Access control,	L+D	PPT using MS Teams Platform	1	58	26-7-2021
54	Revocation of access rights, Capability- Based systems	L+D	PPT using MS Teams Platform	1	59	27-7-2021
55	Case Study: The Linux Operating System: Linux history; Design principles; Kernel modules	L+D	PPT using MS Teams Platform	1	60	29-7-2021
56	Process management; Scheduling; Memory Management	L+D	PPT using MS Teams Platform	1	61	30-7-2021
57	File systems, Input and output; Inter-process communication	L+D	PPT using MS Teams Platform	1	62	2-8-2021
58	Inter-process communication	L+D	PPT using MS Teams Platform	1	63	3-8-2021
59	**Internal Test Assessment III**				64	5-8-2021

Text Books: -

1 Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006.**Reference Books:**

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGrawHill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson

Web Materials

- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-033-computer-system-engineering-spring-2009
- http://nptel.ac.in/courses/106108101
- https://www.elsevier.com/books/operating-systems/tsichritzis/978-0-12-701750-1

Details of Teaching Aids:
Power point Presentation using Zoom Platform

Course incharge | **Module Coordinator** | **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K S INSTITUTE OF TECHNOLOGY, Bangalore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ASSIGNMENT QUESTIONS

Academic Year	**2020-2021**		
Batch	**2019-2023**		
Year/Semester/section	**2nd / 4th /A & B**		
Subject Code-Title	**18CS43- Operating Systems**		
Name of the Instructor	**Vaneeta M**	**Dept**	**CSE**

Assignment No: 1 **Total marks:20**
Date of Issue: 20.05.2021 **Date of Submission: 25.05.2021**

Sl. No.	Assignment Questions	K Level	CO	Marks
1.	**Identify** the following types of systems and explain: Multiprogramming systems, Time Sharing, Multiprocessor Systems, Clustered Systems	Apply	CO1	1
2.	**Make use of** relevant diagrams and explain the layered,microkernel and modules approach to structuring the operating system.	Apply	CO1	1
3.	**Determine** the differences among short-term, medium-term, and long term scheduling.	Apply	CO1	1
4.	**Identify** the five major activities of an operating system in regard to process management, memory and Mass-storage management	Apply	CO1	1
5.	**Utilize** process state diagram and explain different states and transition also mention information that is kept in process control block.	Apply	CO1	1
6.	**Build** the sequence of system calls for copying a file to another (new) file.	Apply	CO1	1
7.	**Pedagogy Activity** - Collaborative study through Team wise Execution of programs for CPU Scheduling Algorithms - Total 12 teams 1. Implement a program to demonstrate the working of First Come First Server (FCFS) CPU scheduling algorithm. 2. Implement a program to demonstrate the working of Shortest Job First (SJF) CPU scheduling algorithm. 3. Implement a program to demonstrate the working of Shortest Remaining Time First (SRTF) CPU scheduling algorithm. 4. Implement a program to demonstrate the working of Priority CPU scheduling algorithm. 5. Implement a program to demonstrate the working of Round Robin (RR) CPU scheduling algorithm.	Apply	CO2	4

Mary
Course In charge

Mary
Module Coordinator

HOD-CSE

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
ASSIGNMENT I 2020-21 EVEN SEMESTER

SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**IV A & B**
Branch	:	**Computer Science & Engineering**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**10**

Q.NO	POINTS	Marks
1	**Multiprogramming systems** • Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. • The idea is as follows: • OS keeps several jobs in memory simultaneously. • OS picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete. • OS simply switches to, and executes, another job. • When that job needs to wait, the CPU is switched to another job, and so on. • As long as at least one job needs to execute, the CPU is never idle. **Time Sharing Systems** • Time sharing (or multitasking) is a logical extension of multiprogramming. • The CPU executes multiple jobs by switching between them. • Switching between jobs occur so frequently that the users can interact with each program while it is running. • Many users are allowed to share the computer simultaneously. **Clustered system** • These systems consist of two or more systems coupled together • These systems share storage & closely linked via LAN. • Used to provide high-availability service. High-availability is obtained by adding a level of redundancy in the system	**1M**
2	**Structure of Operating system:** **Layered:** The OS is divided into a number of layers. • Each layer is built on the top of another layer. The bottom layer is the hardware. • A layer is an implementation of an abstract-object. i.e. The object is made up of data and operations that can manipulate the data. The layer consists of a set of routines that can be invoked by higher-layers. • **Higher-layer** → does not need to know how lower-layer operations are implemented → needs to know only what lower-layer operations do. • **Advantage**: 1) Simplicity of construction and debugging. • **Disadvantages**: 1) Less efficient than other types. 2) Appropriately defining the various layers.(„." a layer can use only lower-layers, careful planning is necessary) **Microkernel:** As UNIX expanded, the kernel became large and difficult to manage.	**1M**

• In the mid-1980s, researchers at Carnegie Mellon University developed an operating system called Mach that modularized the kernel using the microkernel approach.
• This method structures the operating system by removing all nonessential components from the kernel and implementing them as system and user-level programs.
• The result is a smaller kernel.
• There is little concern regarding which services should remain in the kernel and which should be implemented in user space.
• Typically, however, microkernels provide minimal process and memory management, in
addition to a communication facility.

Modules

• The kernel has
→ set of core components and
→ dynamic links in additional services during boot time(or run time).
• Seven types of modules in the kernel (Figure 2.9):
1) Scheduling classes, 2) File systems, 3) Loadable system calls, 4) Executable formats

5) STREAMS modules, 6) Miscellaneous, 7) Device and bus drivers

3. Difference between

1) Long-term scheduler — **1M**

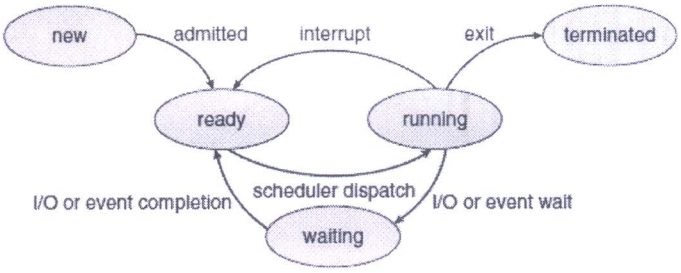
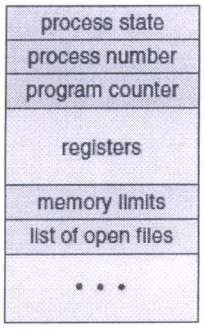
2) Short-term scheduler and

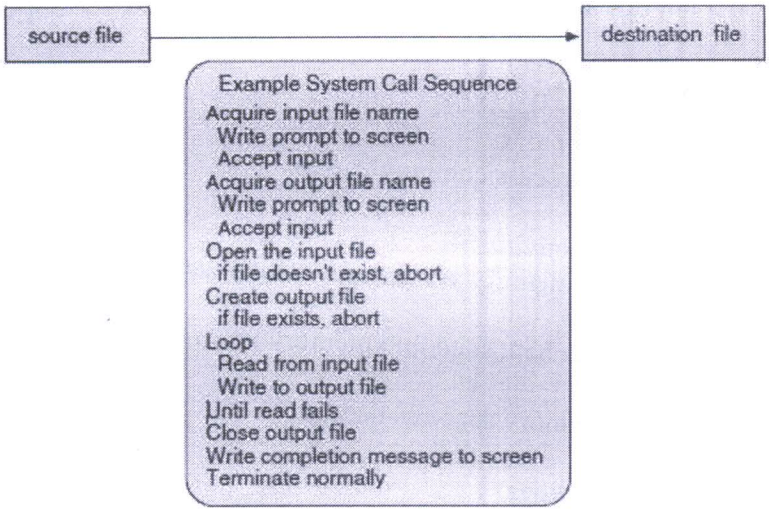
3) Medium-term schedulers

Long-Term Scheduler	Short-Term Scheduler
Also called job scheduler.	Also called CPU scheduler.
Selects which processes should be brought into the ready-queue.	Selects which process should be executed next and allocates CPU.
Need to be invoked only when a process leaves the system and therefore executes much less frequently.	Need to be invoked to select a new process for the CPU and therefore executes much more frequently.
May be slow „„ minutes may separate the creation of one new process and the next.	Must be fast „„ a process may execute for only a few milliseconds.
Controls the degree of multiprogramming.	

1M

• Some time-sharing systems have **medium-term scheduler**

	• The scheduler removes processes from memory and thus reduces the degree of multiprogramming. • Later, the process can be reintroduced into memory, and its execution can be continued where it left off. This scheme is called **swapping**.	
4.	**Process management** 1) Process creation and deletion. 2) Process suspension and resumption. 3) Provision of mechanisms for: a. process synchronization b. process communication **Memory Management** • The OS is responsible for the following activities: 1) Keeping track of which parts of memory are currently being used and by whom 2) Deciding which processes are to be loaded into memory when memory space becomes available 3) Allocating and de-allocating memory space as needed. **Mass System Management 2M** • The OS is responsible for following activities: 1) Free-space management 2) Storage allocation and 3) Disk scheduling. Backing up files on stable (non-volatile) storage media	**1M**
5.	**Process State diagram** : As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of the following states:  • New. The process is being created. • Running. Instructions are being executed. • Waiting. The process is waiting for some event to occur (such as an I/O completion or reception of a signal). • Ready. The process is waiting to be assigned to a processor. Terminated. The process has finished execution. **Process Control Block Explanation and diagram**:  	**1M**

6.		**1M**
7	Pedagogy Activity report – Collaborative study through Team wise Execution of programs for CPU Scheduling Algorithms – Total 12 teams 1. Implement a program to demonstrate the working of First Come First Server (FCFS) CPU scheduling algorithm. 2. Implement a program to demonstrate the working of Shortest Job First (SJF) CPU scheduling algorithm. 3. Implement a program to demonstrate the working of Shortest Remaining Time First (SRTF) CPU scheduling algorithm. 4. Implement a program to demonstrate the working of Priority CPU scheduling algorithm. 5. Implement a program to demonstrate the working of Round Robin (RR) CPU scheduling algorithm.	**4M**

Course incharge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. Institute of Technology, Bangalore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ASSIGNMENT QUESTIONS

Academic Year	2020-21		
Batch	2019-2023		
Year/Semester/section	II/IV/A & B		
Course Code-Title	18CS43- OPERATING SYSTEMS		
Name of the Instructor	Vaneeta M	Dept	CSE

Assignment No: 2 **Total marks:10**
Date of Issue: 22/06/2021 **Date of Submission: 26/06/2021**

Sl. No	Assignment Questions	K Level	CO	Marks
1.	**Build** the solution to Dinning philosopher problem using monitors.	K3	CO2	1
2.	**Apply** TestAndSet() and Swap() instructions for implementation of mutual exclusion.	K3	CO2	1
3.	**Make use of** suitable example and construct resource allocation graph (RAG) for the follows: i) With Deadlock ii) With cycle but no Deadlock ii) No deadlock	K3	CO3	1
4	**Identify** how deadlock occurs and the necessary conditions for deadlock occurrence. Indicate how many of these conditions should occur for deadlock to happen?	K3	CO3	1
5	**Consider** the following snapshot of the system and answer the following questions using Banker's algorithm? Process / Allocation (A B C D) / Max (A B C D) / Available (A B C D) P1: 0 0 1 2 / 0 0 1 2 / 1 5 2 0 P2: 1 0 0 0 / 1 7 5 0 P3: 1 3 5 4 / 2 3 5 6 P4: 0 6 3 2 / 0 6 5 2 P5: 0 0 1 4 / 0 6 5 6 a. Find the need of the allocation? b. Is the system is in safe state? c. If the process P1 request (0,4,2,0) resources cam the request be granted immediately?	K3	CO3	1
6	**Identify** the principle behind paging. Explain its operation using a paging hardware clearly indicating how the logical addresses are converted to physical addresses.	K3	CO3	1
7	**Construct** hardware support for TLB paging scheme with a neat diagram and explain in detail	K3	CO3	1
8	**Build** an example segment table using Segmentation memory management hardware and explain.	K3	CO3	1
9	**Identify** the steps involved in page fault using a neat diagram	K3	CO4	1
10	**Identify** different page replacement algorithms and explain.	K3	CO4	1

May
Course Incharge

May
Module Coordinator

HOD
Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
ASSIGNMENT II 2020-21 EVEN SEMESTER

SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**IV A &B**
Branch	:	**Computer Science & Engineering**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**10**

Q.NO	POINTS	Marks

1	monitor dp
	{
	enum {THINKING, HUNGRY, EATING} state [5]
	condition self [5] ;
	void pickup(int i) {
	state [i] = HUNGRY;
	test (i) ;
	if (state [i] != EATING)
	self [i] .wait() ;
	}
	void putdown(int i) {
	state [i] = THINKING;
	test((i + 4) % 5} ;
	test((i + 1) % 5) ;
	}
	void test(int i) {
	if ((state [(i + 4) % 5] != EATING) && (state [i] == HUNGRY) && (state [(i + 1) % 5] != EATING)) {
	state [i] = EATING;
	self [i].signal() ;
	}
	initialization-code ()
	{
	for (int i = 0; i < 5; i++)
	state [i] = THINKING;
	}
	}
		1M

2	Definition: **test_and_set Instruction** **boolean test_and_set (boolean *target)** **{** **boolean rv = *target;** ***target = TRUE;** **return rv:** **}** 1. Executed atomically 2. Returns the original value of passed parameter 3. Set the new value of passed parameter to "TRUE".	**1M**

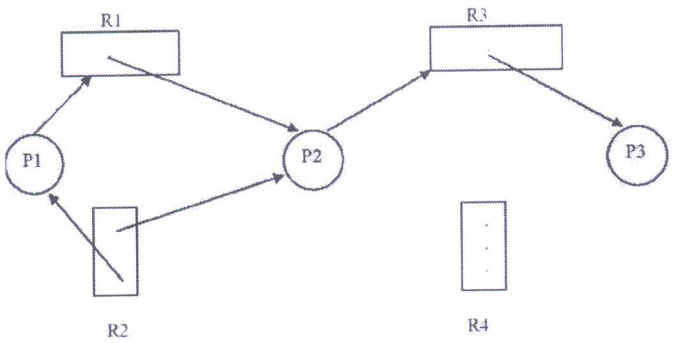
4.

Shared Boolean variable lock, initialized to FALSE

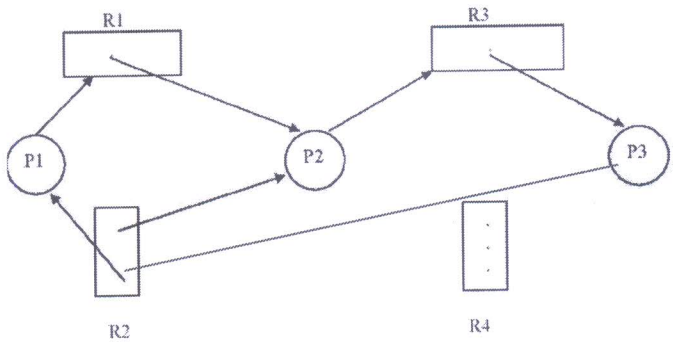
Solution:

```
do {
    while (test_and_set(&lock))
        ; /* do nothing */
            /* critical section */
    lock = false;
            /* remainder section */
} while (true);
```

3. **RAG without deadlock** **1M**



In the above graph if process P3 makes a request for an instance of R2 then deadlock takes place.

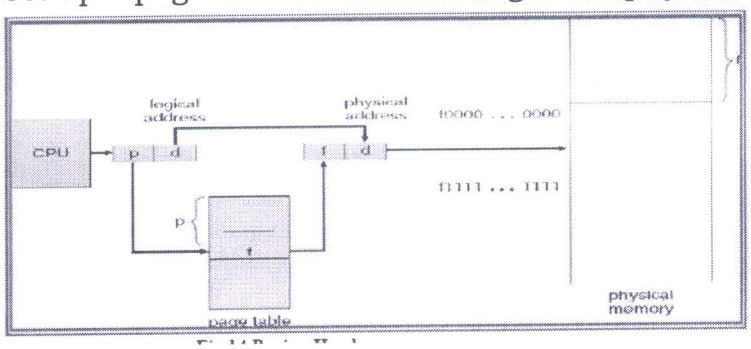


4. Deadlock occurs when there is a cycle in RAG. If first process is requesting for a resource held to second process and second process is waiting for resource held by first process, then deadlock occurs **1M**

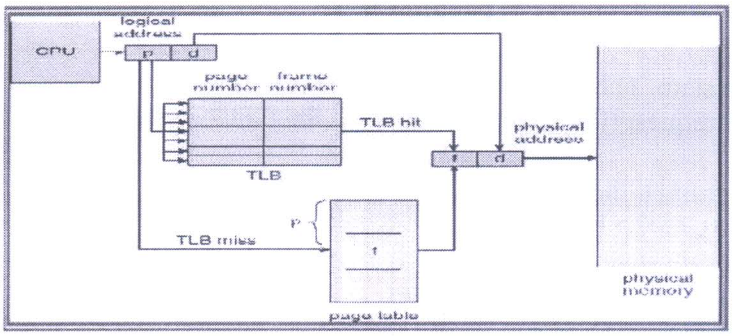
Necessary Conditions:A deadlock situation can occur if the following 4 conditions occur simultaneously in asystem:-

. **Mutual Exclusion**: Only one process must hold the resource at a time. If any other process requestsfor the resource, the requesting process must be delayed until the resource has been released.

. **Hold and Wait**:-A process must be holding at least one resource and

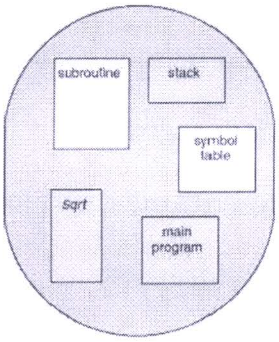
	waiting to acquire additionalresources that are currently being held by the other process. . **No Preemption**:-Resources can't be preempted i.e., only the process holding the resources must release itafter the process has completed its task. . **Circular Wait**:-A set {P0,P1 ..... Pn} of waiting process must exist such that P0 is waiting for a resource i.e., held by P1, P1 is waiting for a resource i.e., held by P2. Pn-1 is waiting for resource held by processPn and Pn is waiting for the resource i.e., held by P1. All the four conditions must hold for a deadlock to occur	
5.	**Need Matrix** P1<0,0,0,0> P2<0,7,5,2>, P3<1,0,0,2>, P4<0,0,2,0>, P5<0,6,4,2> System is in safe state. The safe sequence is <P1,P3,P4,P5,P2> If the process P1 request (0,4,2,0) resources cam the request be granted immediately? If the request is granted the new need matrix is P1<0,0,0,0> P2<0,3,3,2> P3<1,0,0,2> P4<0,0,2,0> P5<0,6,4,2> The system is In safe state with safe sequence as <P1,P3,P4,P5,P2> so the request can be granted.	**1M**
6.	**Principle of Paging** Paging is a memory management scheme that permits the physical address space of a process to be non-contiguous. Support for paging is handled by hardware. It is used to avoid external fragmentation. Paging avoids the considerable problem of fitting the varying sized memory chunks on to the backing store. Divide physical memory into fixed-sized blocks called frames • Size is power of 2, between 512 bytes and 16 Mbytes Divide logical memory into blocks of same size called pages Keep track of all free frames To run a program of size N pages, need to find N free frames and load program Set up a page table to translate logical to physical addresses  	**1M**
7	The hardware implementation of the page table can be done in several ways: 1. The simplest method is that the page table is implemented as a set of dedicated registers. These registers must be built with very high-speed logic for making paging address translation. Every accessed memory must go through paging map. The use of registers for page table is satisfactory if the page table is small.	**1M**

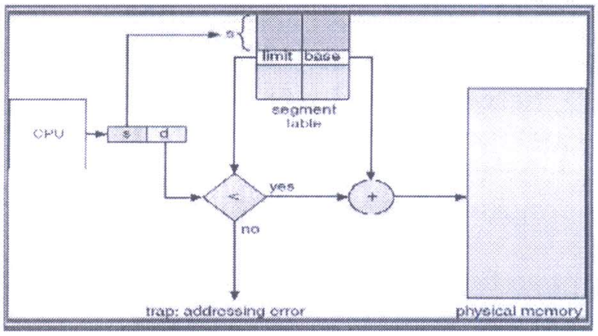
2. If the page table is large then the use of registers is not visible. So, the page table is kept in the main memory and a page table base register [PTBR] points to the page table. Changing the page table requires only one register which reduces the context switching type. The problem with this approach is the time required to access memory location. To access a location [i] first we have to index the page table using PTBR offset. It gives the frame number which is combined with the page offset to produce the actual address. Thus, we need two memory accesses for a byte.
The only solution is to use special, fast, lookup hardware cache called translation look aside buffer [TLB] or associative register



8 • Most users do not think memory as a linear array of bytes rather the users think memory as a collection of variable sized segments which are dedicated to a particular use such as code, data, stack, heap etc. 1M

• A logical address is a collection of segments. Each segment has a name and length. The address specifies both the segment name and the offset within the segments.





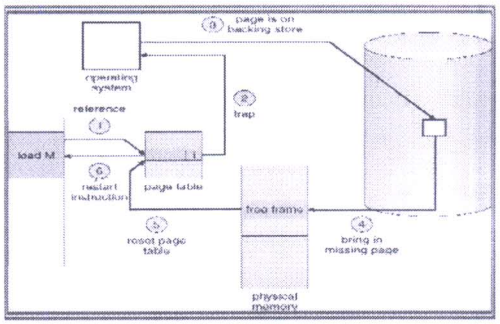
The users specify address by using two quantities: a segment name and an offset.
For simplicity the segments are numbered and referred by a segment number.
So, the logical address consists of <segment number, offset>.

Hardware support: This mapping is affected by a segment table. Each entry in the segment table has a segment base and segment limit. The segment base contains the starting physical address where the segment resides and limit specifies the length of the segment 1M

9 A page fault causes the following sequence to occur: 4M

1. Trap to the OS.
2. Save the user registers and process state.
3. Determine that the interrupt was a page fault.
4. Checks the page references were legal and determine the location of page on disk.
5. Issue a read from disk to a free frame.
6. If waiting, allocate the CPU to some other user.
7. Interrupt from the disk.

8. Save the registers and process states of other users.
9. Determine that the interrupt was from the disk.
10. Correct the page table and other table to show that the desired page is now in memory.
11. Wait for the CPU to be allocated to this process again.
12. Restore the user register process state and new page table then resume the interrupted instruction.
13.



10 **FIFO Algorithm:** **1M**

• This is the simplest page replacement algorithm. A FIFO replacement algorithm associates each page the time when that page was brought into memory.

• When a Page is to be replaced the oldest one is selected.

• We replace the queue at the head of the queue. When a page is brought into memory, we insert it at the tail of the queue.

Example: Consider the following references string with frames initially empty.

Optimal Algorithm Optimal page replacement algorithm is mainly to solve the problem of Belady's Anamoly. Optimal page replacement algorithm has the lowest page fault rate of all

o algorithms. An optimal page replacement algorithm exists and has been called OPT. The working is simple "Replace the page that will not be used for the longest period of time" Example: consider the following reference string

Least Recently Used (LRU) Algorithm

If the optimal algorithm is not feasible, an approximation to the optimal algorithm is possible. The main difference b/w OPTS and FIFO is that;

• used.

•

FIFO algorithm uses the time when the pages was built in and OPT uses the time when a page is to be

The LRU algorithm replaces the pages that have not been used for longest period of time. x The LRU

Associates with its pages with time of that pages last use. x This strategy is the optimal page replacement algorithm looking backward in time rather than forward. Ex: consider the following reference string

Course incharge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. Institute of Technology, Bangalore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ASSIGNMENT QUESTIONS

Academic Year	2020-21		
Batch	2019-2023		
Year/Semester/section	II/IV/A & B		
Course Code-Title	18CS43- OPERATING SYSTEMS		
Name of the Instructor	Vaneeta M	Dept	CSE

Assignment No: 3 **Total marks:10**
Date of Issue: 31/07/2021 **Date of Submission: 07/08/2021**

Sl. No	Assignment Questions	K Level	CO	Marks
1.	Utilize an example and design any three types of directory structures.	K3	CO4	1
2.	Make use of an example to differentiate different file access methods to access file from disk.	K3	CO4	1
3.	Utilize contiguous, Linked and Index methods and explain file allocation.	K3	CO4	1
4	Identify the different attributes of a file and operations performed on files.	K3	CO4	1
5	Utilize an example and distinguish between SSTF, FCFS, SCAN, C-SCAN disk scheduling methods.	K3	CO5	1
6	Construct access matrix of domain and objects with copy, transfer and owner rights.	K3	CO5	1
7	Identify the different components of a Linux system	K3	CO5	1
8	Identify the usage of i) file control block(FCB) and ii) File Mounting in file system implementation.	K3	CO5	1
9	Identify how process is managed on Linux platform	K3	CO5	1
10	Apply any two types of free space management techniques to a list of free blocks on disk and mention its advantages and disadvantages.	K3	CO5	1

Course Incharge

Module Coordinator

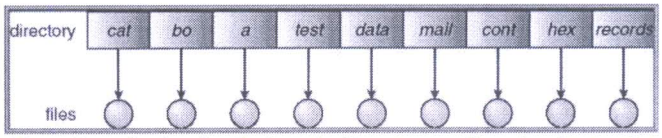
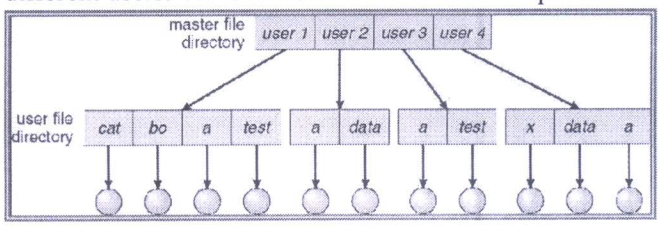
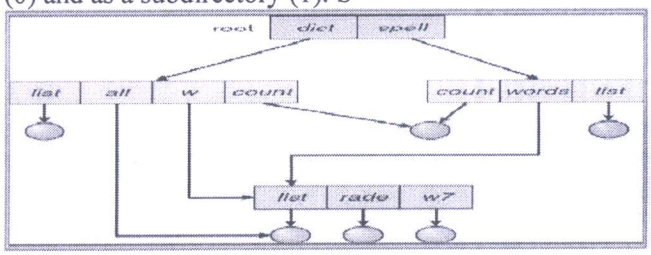
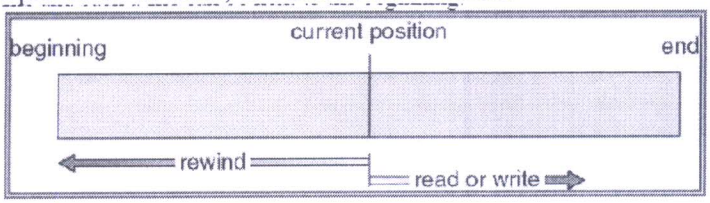
HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
ASSIGNMENT III 2020-21 EVEN SEMESTER

SCHEME AND SOLUTION

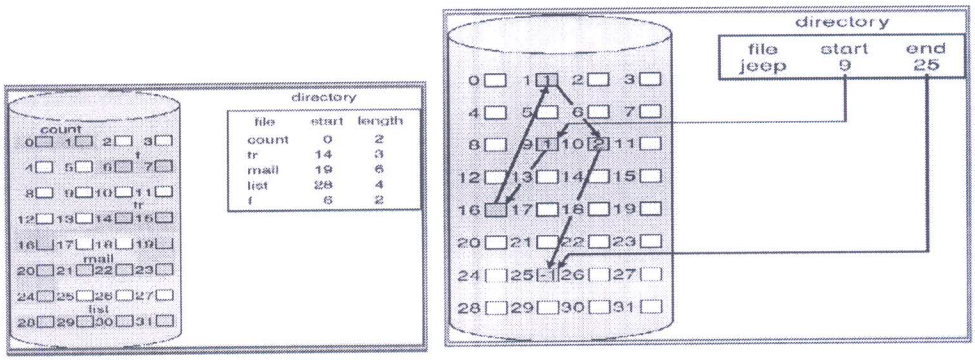
Degree	:	**B.E**	**Semester**	:	**IV A &B**
Branch	:	**Computer Science & Engineering**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**10**

Q.NO	POINTS	Marks
1	Single Level: This is the simplest directory structure. All the files are contained in the same directory which is easy to support and understand  Two Level: A single level directory often leads to the confusion of file names between different users. The solution here is to create separate directory or each user  Tree Structured: MS-DOS use Tree structure directory. It allows users to create their own subdirectory and to organize their files accordingly. A subdirectory contains a set of files or subdirectories. A directory is simply another file, but it is treated in a special way. The entire directory will have the same internal format. One bit in each entry defines the entry as a file (0) and as a subdirectory (1). S 	**1M**
2	**Sequential Access**: Sequential access is the simplest access method. Information in the file is processed in order, one record after another.  **Direct Access**: Direct access allows random access to any file block. This method is based on disk model of a file. A file is made up of fixed length logical records. It allows the program to read and write records rapidly in any order. A direct access file allows arbitrary blocks to be read or written.	**1M**

sequential access	implementation for direct access
reset	*cp = 0;*
read next	*read cp;* *cp = cp + 1;*
write next	*write cp;* *cp = cp + 1;*

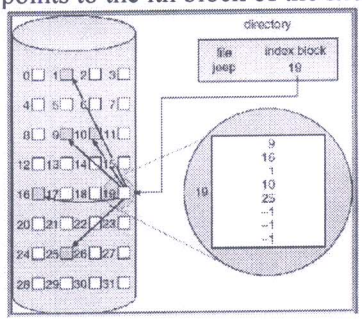
Indexing Method: The index is like an index at the end of a book which contains pointers to various blocks. To find a record in a file, we search the index and then use the pointer to access the file directly and to find the desired record. With large files index file itself can be very large to be kept in memory. One solution to create an index to the index files itself. The primary index file would contain pointer to secondary index files which would point to the actual data items.

3. **Contiguous Allocation**: A single set of blocks is allocated to a file at the time of file creation. This is a pre-allocation strategy that uses portion of variable size. The file allocation table needs just a single entry for each file, showing the starting block and the length of the file. The figure shows the contiguous allocation method. **1M**



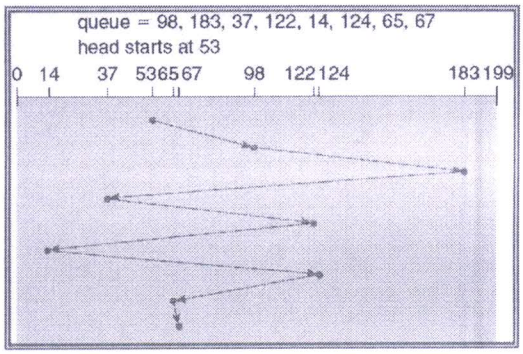
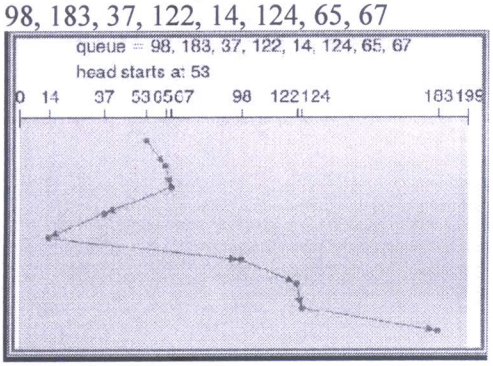
Linked Allocation: It solves the problem of contiguous allocation. This allocation is on the basis of an individual block. Each block contains a pointer to the next block in the chain. x The disk block can be scattered any where on the disk. The directory contains a pointer to the first and the last blocks of the file. The following figure shows the linked allocation. To create anew file, simply create a new entry in the directory.

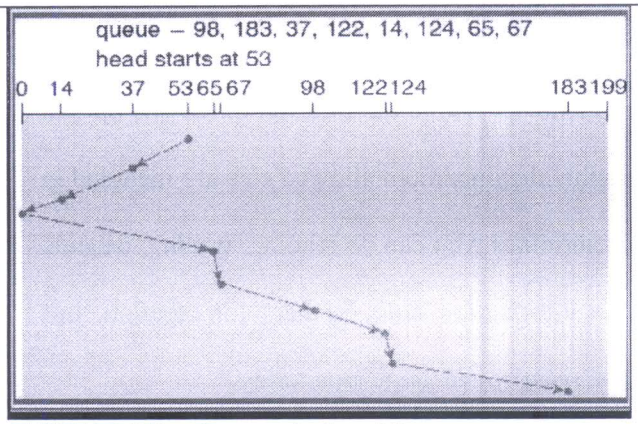
Indexed Allocation: The file allocation table contains a separate one level index for each file. The index has one entry for each portion allocated to the file. The ith entry in the index block points to the ith block of the file. The following figure shows indexed allocation.



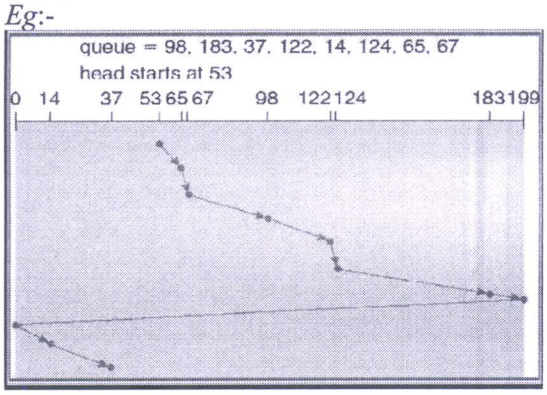
4. File Attributes:-File attributes varies from one OS to other. The common file attributes are: **1M**

Name:-The symbolic file name is the only information kept in human readable form.
Identifier:-The unique tag, usually a number, identifies the file within the file system. It is the non-

	readable name for a file. Type:-This information is needed for those systems that supports different types. Location:-This information is a pointer to a device and to the location of the file on that device. Size:-The current size of the file and possibly the maximum allowed size are included in this attribute. Protection:-Access control information determines who can do reading, writing, execute and so on. Time, data and User Identification:-This information must be kept for creation, last modification and last use. These data are useful for protection, security and usage monitoring.	
5.	**FCFS scheduling algorithm**: This is the simplest form of disk scheduling algorithm. This services the request in the order they are received. This algorithm is fair but do not provide fastest service. It takes no special time to minimize the overall seek time. *Eg:*- consider a disk queue with request for i/o to blocks on cylinders. 98, 183, 37, 122, 14, 124, 65, 67  **SSTF** (Shortest seek time first) **algorithm**:This selects the request with minimum seek time from the current head position. Since seek time increases with the number of cylinders traversed by head, SSTF chooses the pending request closest to the current head position. *Eg:*- :-consider a disk queue with request for i/o to blocks on cylinders. 98, 183, 37, 122, 14, 124, 65, 67  **SCAN algorithm**:In this the disk arm starts at one end of the disk and moves towards the other end, servicing the request as it reaches each cylinder until it gets to the other end of the disk. At the other end, the direction of the head movement is reversed and servicing continues. *Eg:*-:-consider a disk queue with request for i/o to blocks on cylinders. 98, 183, 37, 122, 14, 124, 65, 67	**1M**



C-SCAN (Circular scan) **algorithm**:
C-SCAN is a variant of SCAN designed to provide a more uniform wait time. Like SCAN, C-SCAN moves the head from end of the disk to the other servicing the request along the way. When the head reaches the other end, it immediately returns to the beginning of the disk, without servicing any request on the return. The C-SCAN treats the cylinders as circular list that wraps around from the final cylinder to the first one.
Eg:-



6. **Access Matrix:** **1M**

View protection as a matrix (*access matrix*)
- Rows represent domains
- Columns represent objects
- *Access(i, j)* is the set of operations that a process executing in Domain$_i$ can invoke on Object$_j$

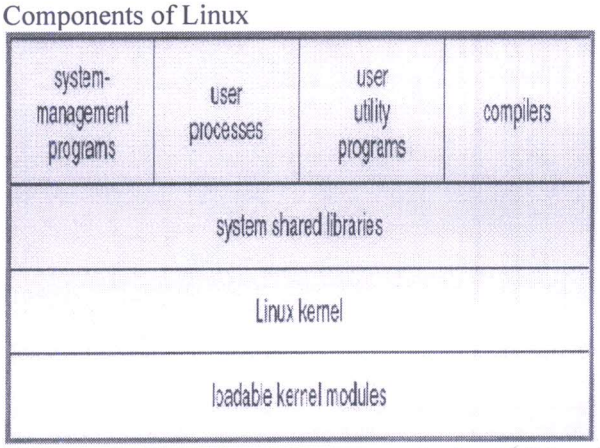
domain \ object	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

object / domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

Owner, copy and transfer rights

object / domain	F_1	F_2	F_3
D_1	owner execute		write
D_2		read* owner	read* owner write
D_3	execute		

Components of Linux



7 The **system libraries** define a standard set of functions through which applications interact with the kernel, and which implement much of the operating-system functionality that does not need the full privileges of kernel code **1M**

The **system utilities** perform individual specialized management tasks x Sections of kernel code that can be compiled, loaded, and unloaded independent of the rest of the kernel

KERNEL MODULES

x A kernel module may typically implement a device driver, a file system, or a networking protocol

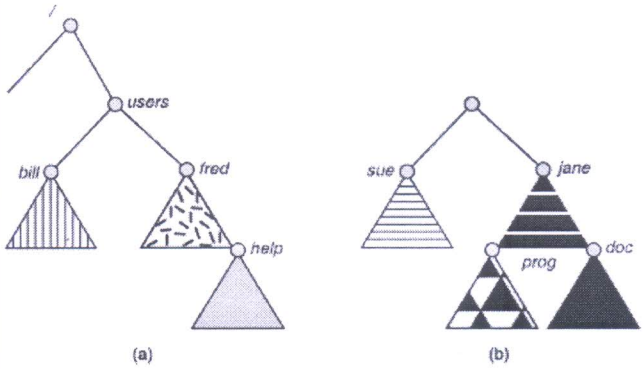
x The module interface allows third parties to write and distribute, on their own terms, device drivers or file systems that could not be distributed under the GPL

x Kernel modules allow a Linux system to be set up with a standard, minimal kernel, without any extra device drivers built in

x Three components to Linux module support:

o module management

o driver registration

8	o conflict resolution FILE SYSTEM MOUNTING The file system must be mounted before it can be available to processes on the system The procedure for mounting the file is: a. The OS is given the name of the device and the location within the file structure at which toattach the file system (mount point).A mount point will be an empty directory at which themounted file system will be attached  	1M
9	A typical file control blocks consists of File Permission ,File dates (create, access, write) File owner, group, acc File size File data blocks **PROCESS MANAGEMENT** o UNIX process management separates the creation of processes and the running of a new program into two distinct operations. The **fork** system call creates a new process. o new program is run after a call to **execve** o Under Linux, process properties fall into three groups: the process's identity, environment, and **Process Identity** Process ID (PID). o Credentials. o Personality.. **Process Environment** **Scheduling**	1M
10	Bit vector- A Vector that maintains the list of free blocks. Linked list of free blocks , Grouping	1M

Course incharge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2020 – 21 EVEN SEMESTER

USN | | | | | | | | | | |

SET-A

Degree : B.E **Semester: IV**

Branch : Computer Science &Engineering **Course Code: 18CS43**

Course Title : Operating Systems **Date: 25-05-2021**

Duration : 90 Minutes **Max Marks: 30**

Note: Answer ONE full question from each part.

Q No.	Question	Marks	CO mapping	K-Leve
	PART-A			
1(a)	**Identify** the essential features of the following: Multiprogramming systems, Time Sharing, Multitasking Systems.	6	CO1	K3
(b)	**Utilize** process state diagram and explain different states and transition also mention information that is kept in process control block.	6	CO1	K3
(c)	**Identify** the services provided by an operating system to user and the system.	6	CO1	K3
	OR			
2(a)	**Identify** the five major activities of an operating system in regard to process management, memory and file system management.	6	CO1	K3
(b)	**Construct** an example program for creating a child process using fork system call and explain how child process is allocated resources and address space.	6	CO1	K3
(c)	**Make use of** relevant diagrams and explain the layered and microkernel approach for structuring the operating system.	6	CO1	K3
	PART-B			
3(a)	**Construct** multithreading models and mention benefits of creating a thread over creating a process.	6	CO2	K3
(b)	Solve for the following set of processes with their burst times and priority as shown **Process** **Burst Time** **Priority** P0 8 4 P1 6 1 P2 1 2 P3 9 2 P4 3 3 Draw a Gantt chart that illustrates the execution of these processes using the non preemptive priority and RR (Time Quantum=1). Find turnaround time for each and average waiting time.	6	CO2	K3
	OR			
4(a)	**Identify** the different process scheduling criteria's.	6	CO2	K3
(b)	**Solve** for the following set of processes given arrival and burst times as show: **Process** **Arrival Time** **Burst Time** P0 0 6 P1 1 3 P2 2 1 P3 3 4 a) Draw a Gantt chart that illustrates the execution of these processes using the SRTF and round robin (Time Quantum=2ms) b) Find turnaround time, waiting time for each process and AWT. Hence show that SRTF is faster the SJF.	6	CO2	K3

Course Incharge Module Coordinator HOD

Head of the Department
Dept. of Computer Science & Engg

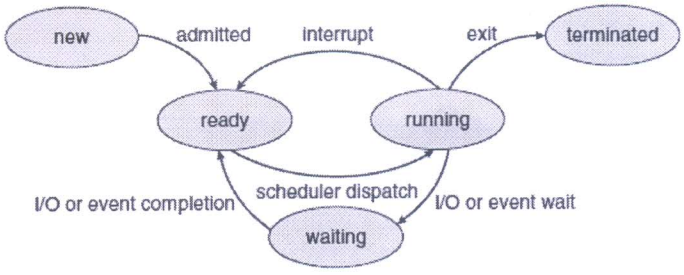
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

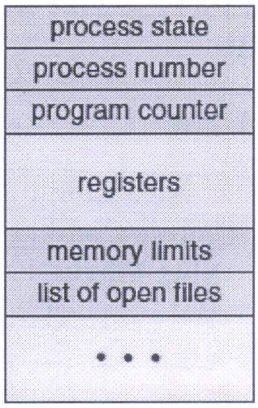
I SESSIONAL TEST QUESTION PAPER 2020-21

EVEN SEMESTER

SCHEME AND SOLUTION –SET A

Degree	:	**B.E**	**Semester**	:	**IV**
Branch	:	**CSE**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
	PART-A	
1(a)	**Multiprogramming systems (2M)** • Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. • The idea is as follows: • OS keeps several jobs in memory simultaneously. • OS picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete. • OS simply switches to, and executes, another job. • When that job needs to wait, the CPU is switched to another job, and so on. • As long as at least one job needs to execute, the CPU is never idle. **Time Sharing Systems (2M)** • Time sharing (or multitasking) is a logical extension of multiprogramming. • The CPU executes multiple jobs by switching between them. • Switching between jobs occur so frequently that the users can interact with each program while it is running. • Many users are allowed to share the computer simultaneously. **Mulitasking system (2M)** Perform multiple tasks in foreground and background Example.	2+2+2=6M
(b)	**Process State diagram : (3M)** As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of the following states:  • New. The process is being created. • Running. Instructions are being executed. • Waiting. The process is waiting for some event to occur (such as an I/O completion or reception of a signal). • Ready. The process is waiting to be assigned to a processor. Terminated. The process has finished execution.	3+3=6M

	Process Control Block Explanation and diagram: (3M)  	
(c)	**Services provided by an operating system to user: (4M)** User interface, Program execution, I/O operations, Communications, Error Detection, File Manipulation **Services provided by an operating system to system: (2M)** Resource Management, Accounting, Protection and Security	4+2-=6M
2(a)	The operating system is responsible for the following activities in connection with **Process management :2M** 1) Process creation and deletion. 2) Process suspension and resumption. 3) Provision of mechanisms for: a. process synchronization b. process communication **Memory Management 2M** • The OS is responsible for the following activities: 1) Keeping track of which parts of memory are currently being used and by whom 2) Deciding which processes are to be loaded into memory when memory space becomes available 3) Allocating and de-allocating memory space as needed. **File System Management 2M** 1) Creating and deleting files. 2) Creating and deleting directories. 3) Supporting primitives for manipulating files & directories. 4) Mapping files onto secondary storage. Backing up files on stable (non-volatile) storage media	2+2+2=6M
(b)	**Program 3M**	3+3=6M

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
pid_t pid;

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
       fprintf(stderr, "Fork Failed");
       return 1;
    }
    else if (pid == 0) { /* child process */
       execlp("/bin/ls","ls",NULL);
    }
    else { /* parent process */
       /* parent will wait for the child to complete */
       wait(NULL);
       printf("Child Complete");
    }

    return 0;
}
```

Explanation 3M

- Child-process may
  → get resources directly from the OS or

  → get resources of parent-process. This prevents any process from overloading the system.
- Two options exist in terms of the address-space of the new process:

  → The child-process is a duplicate of the parent-process (it has the same program and data as the parent).

The child-process has a new program loaded into it

(c) **Layered Approach Diagram and explanation** — 3+3=6M

Diagram 1M

Explanation 2M

The OS is divided into a number of layers. Each layer is built on the top of another layer. The bottom layer is the hardware. A layer is an implementation of an abstract-object. i.e. The object is made up of data and operations that can manipulate the data.

Advantage:

Disadvantages

Micro-Kernels Diagram and explanation

Diagram 1M

Explanation 2M

This method structures the operating system by removing all nonessential components from the kernel and implementing them as system and user-level programs. The result is a smaller kernel.

Advantages

Disadvantages

PART- B

3(a) Multithreading models: **Explanation and Diagram** — 2+2+2=6M

Many to One (2M) **One-to-One Model (2M)** **Many-to-Many Model(2M)**

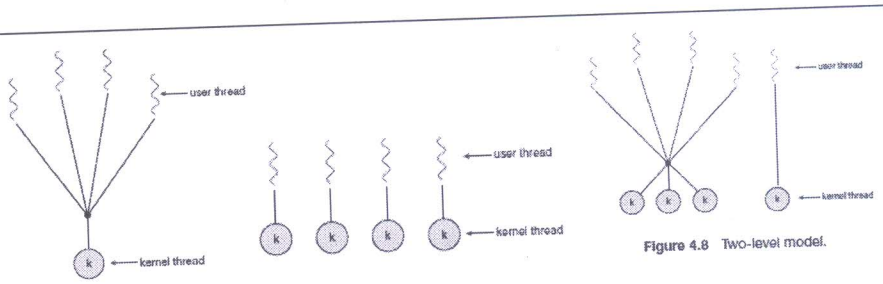


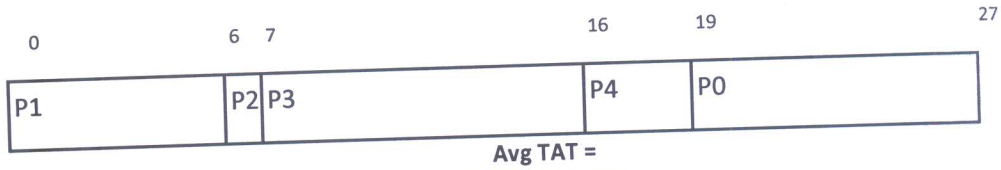
Figure 4.8 Two-level model.

(b) **Non Preemptive**: Gantt Chart-1M, AWT and TAT:1M — 3+3=6M

Process	Burst Time	Priority
P0	8	4
P1	6	1
P2	1	2
P3	9	2
P4	3	3

1=High Priority, 4= Low Priority

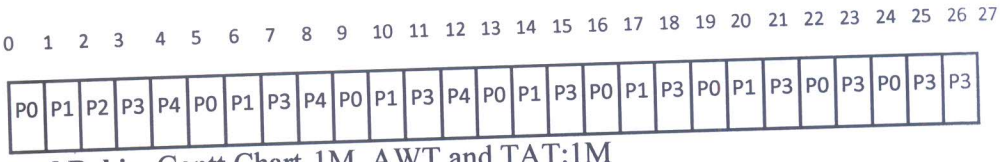
Non-Preemptive Priority **3M**



Avg. Wait Time = 0+6+7+16+19 = 48/5 = 9.6ms

Avg TAT = 6+7+16+19+27 75/5 = 15ms

Round Robin (Quantumm)



Round Robin: Gantt Chart-1M, AWT and TAT:1M
AWT=12.4ms, Avg TAT=17.8ms

4(a) **Scheduling Criteria** : Criteria used to compare CPU-scheduling algorithms: CPU Utilization, Throughput, Turnaround Time, Waiting time, Response Time — 6M

(b) — 3+2+1=6M

Process	Arrival Time	Burst Time
P0	0	6
P1	1	3
P2	2	1
P3	3	4

Round Robin: Gantt Chart-2M, AWT and TAT:1M
AWT = 5.25ms
Turn-around time -1M TAT: P0=14ms, P1=9ms, P2=3ms, P4=9ms.

	SRTF: Gantt Chart-1M, AWT and TAT:1M AWT = 4.25ms TAT: P0=6ms, P1=9ms, P2=5ms, P4=11ms. SRTF is faster than round robin 1M	

Course incharge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2020 – 21 EVEN SEMESTER

USN | | | | | | | | | | |

SET-B

Degree : B.E
Branch : Computer Science &Engineering
Course Title : Operating Systems
Duration : 90 Minutes

Semester: IV
Course Code: 18CS43
Date: 25-05-2021
Max Marks: 30

Note: Answer ONE full question from each part.

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Determine** the differences among short-term, medium-term, and long term scheduling.	6	CO1	K3
(b)	**Build** the sequence of system calls for copying a file to another (new) file.	6	CO1	K3
(c)	**Identify** the essential features of the following: Multiprogramming systems, Symmetric and Asymmetric Multiprocessor Systems.	6	CO1	K3
	OR			
2(a)	**Identify** the five major activities of an operating system in regard to process management, memory and Mass-storage management.	6	CO1	K3
(b)	**Utilize** process state diagram and explain different states and transition also mention information that is kept in process control block.	6	CO1	K3
(c)	**Make use of** relevant diagrams and explain the layered, microkernel and modules approach to structuring the operating system.	6	CO1	K3
	PART-B			
3(a)	**Solve** for the following set of processes, with the length of CPU burst in milliseconds. Process PI P2 P3 P4 P5 Arrival time 00 02 03 06 30 Burst time 10 12 14 16 05 Draw a Gantt chart that illustrates the execution of these processes using the preemptive shortest job first (SJF) algorithm and preemptive priority scheduling algorithm. Given priority of each process is PI = 4, P2=3, P3=5, P4= 1 and P5= 1. Also find the average waiting time. Hence find the average waiting time.	6	CO2	K3
(b)	**Determine** the benefits of multithreading also explain with a diagram multithread.	6	CO2	K3
	OR			
4(a)	**Solve** for the following set of processes, with the length of CPU burst in milliseconds. Process PI P2 P3 P4 Arrival time 00 01 02 03 Burst time 10 29 03 07 Draw a Gantt chart and calculate waiting and turnaround time for SJF, RR for time quantum of 10. Also calculate average waiting time.	6	CO2	K3
(b)	**Identify** the different process scheduling criteria's.	6	CO2	K3

Course in charge

Module Coordinator

HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

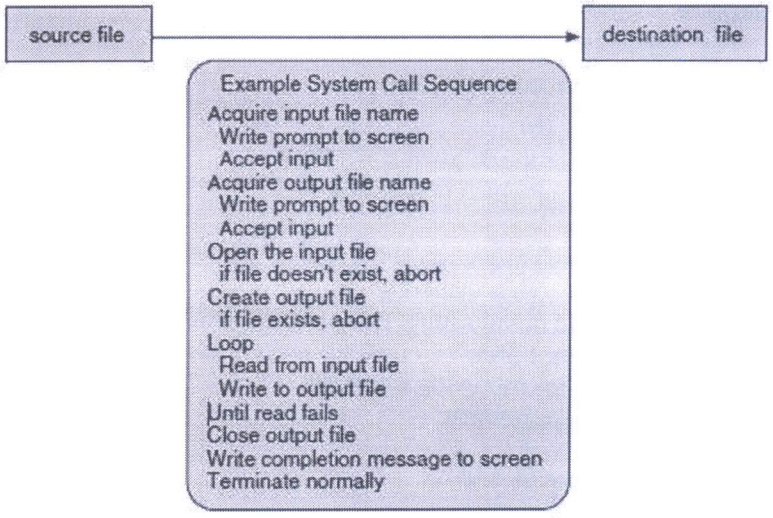
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

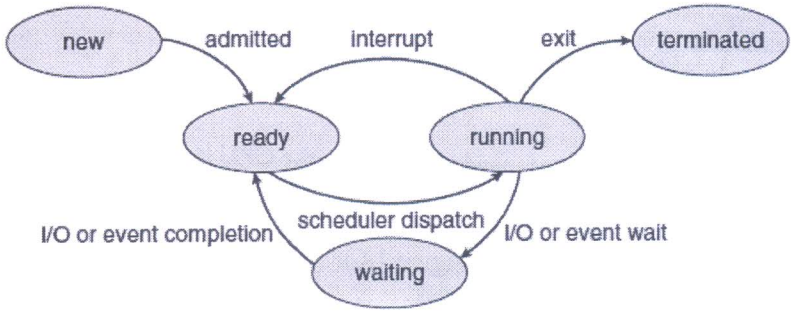
I SESSIONAL TEST QUESTION PAPER 2020-21

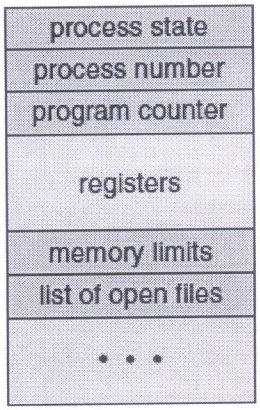
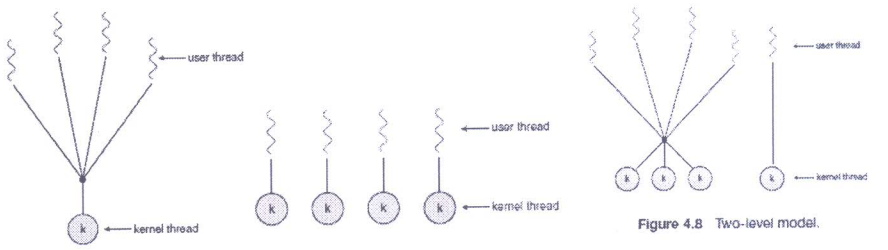
EVEN SEMESTER

SCHEME AND SOLUTION –SET B

Degree	:	**B.E**	**Semester**	:	**IV**
Branch	:	**CSE**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
	PART-A	
1(a)	Long Term Scheduler: 2M Short Term Scheduler: 2M Medium Term Scheduler:2M	2+2+2=6M
(b)	Sequence of system calls for copying a file to another (new) file. 	6M
(c)	**Multiprogramming systems (2M)** • Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. • The idea is as follows: 1) OS keeps several jobs in memory simultaneously. 2) OS picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete. 3) OS simply switches to, and executes, another job. 4) When that job needs to wait, the CPU is switched to another job, and so on. 5) As long as at least one job needs to execute, the CPU is never idle. **Asymmetric Multiprocessing 2M** • This uses master-slave relationship (Figure 1.6). • Each processor is assigned a specific task.	2+2+2=6M

	• A master-processor controls the system. • The master-processor schedules and allocates work to the slave-processors. **Symmetric Multiprocessing 2M** • Each processor runs an identical copy of S. • All processors are peers; no master-slave relationship exists between processors. • Advantages: 1) Many processes can run simultaneously. 2) Processes and resources are shared dynamically among the various processors. • Disadvantage: Since CP s are separate, one CPU may be sitting idle while another CPU is overloaded. This results in inefficiencies	
2(a)	The operating system is responsible for the following activities in connection with **Process management :2M** • Process creation and deletion. • Process suspension and resumption. 1) Provision of mechanisms for: process synchronization b. process communication **Memory Management** 2M • The OS is responsible for the following activities: 1) Keeping track of which parts of memory are currently being used and by whom 2) Deciding which processes are to be loaded into memory when memory space becomes available 3) Allocating and de-allocating memory space as needed. **Mass System Management 2M** • The OS is responsible for following activities: 1) Free-space management 2) Storage allocation and 3) Disk scheduling. Backing up files on stable (non-volatile) storage media	2+2+2=6M
(b)	Process State diagram 2M Explanation 4M As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of the following states:  	2+4=6M

	• New. The process is being created. • Running. Instructions are being executed. • Waiting. The process is waiting for some event to occur (such as an I/O completion or reception of a signal). • Ready. The process is waiting to be assigned to a processor. Terminated. The process has finished execution. Process Control Block 3M 	
(c)	Layered Approach Diagram and explanation 3M Modules approach Diagram and explanation 3M	3+3=6M
	PART- B	
3(a)	**Preemptive Shortest Job First:** Gantt Chart-2M, AWT:1M Process PI P2 P3 P4 P5 Arrival time 00 02 03 06 30 Burst time 10 12 14 16 05 AWT = 13.4ms **Preemptive Priority:** Gantt Chart-2M, AWT:1M AWT = 17.8ms	3+3=6M
(b)	Multithreading models. **Explanation** **Diagram** Many to One (2M) **One-to-One Model (2M) Many-to-Many Model(2M)**  Figure 4.8 Two-level model.	2+2+2=6M
4(a)	Process PI P2 P3 P4 Arrival time 00 01 02 03	3+3=6M

	Burst time 10 29 03 07 **Shortest Job First:** Gantt Chart-1M, AWT:1M and TAT:1M AWT = 7.75ms Turn-around time -1M TAT: P0=20ms, P1=48ms, P2=3ms, P4=9ms. **Round Robin:** Gantt Chart-1M, AWT and TAT:1M AWT = 14.25ms TAT: P0=10ms, P1=48ms, P2=21ms, P4=27ms.	
(b)	**Scheduling Criteria**: Criteria used to compare CPU-scheduling algorithms: CPU Utilization, Throughput, Turnaround Time, Waiting time, Response Time Any 6: 1M each	6M

Course incharge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
II SESSIONAL TEST QUESTION PAPER 2020-21 EVEN SEMESTER

SET – A

USN | | | | | | | | | | |

Degree : B.E
Branch : Computer Science & Engineering
Course Title : Operating Systems
Duration : 90 Minutes

Semester : IV
Course Code : 18CS43
Date : 27-06-2021
Max Marks : 30

Note: Answer ONE full question from each part.

<table>
<tr><th>Q No.</th><th>Question</th><th>Marks</th><th>CO mapping</th><th>K-Level</th></tr>
<tr><td colspan="5">PART-A</td></tr>
<tr><td>1(a)</td><td>Develop comparison study between the following:
a) Logical and physical address space. b) Internal and External fragmentation</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td>(b)</td><td>Construct hardware support for TLB paging scheme with a neat diagram and explain in detail</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td>(C)</td><td>Consider the following snapshot of resource allocation at time t0.
<table>
<tr><th rowspan="2">Process</th><th colspan="3">Allocation</th><th colspan="3">Request</th><th colspan="3">Available</th></tr>
<tr><th>A</th><th>B</th><th>C</th><th>A</th><th>B</th><th>C</th><th>A</th><th>B</th><th>C</th></tr>
<tr><td>P0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>P1</td><td>2</td><td>0</td><td>0</td><td>2</td><td>0</td><td>2</td><td colspan="3" rowspan="5"></td></tr>
<tr><td>P2</td><td>3</td><td>0</td><td>3</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>P3</td><td>2</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>P4</td><td>0</td><td>0</td><td>2</td><td>0</td><td>0</td><td>2</td></tr>
</table>
a) Show that the system is not deadlocked by generating one safe sequence.
b) At instance t1, P2 makes one additional request for instance of type C. Show that the system is deadlocked if the request is granted. Write down the deadlocked processes</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>2(a)</td><td>Identify how deadlock occurs and the necessary conditions for deadlock occurrence. Indicate how many of these conditions should occur for deadlock to happen?</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td>(b)</td><td>Build an example segment table using Segmentation memory management hardware and explain.</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td>(C)</td><td>The operating system contains three resources. The numbers of instances of each resource type are (7, 7,10). The current allocation state is given below.
<table>
<tr><th rowspan="2">Process</th><th colspan="3">Allocation</th><th colspan="3">Max</th></tr>
<tr><th>R1</th><th>R2</th><th>R3</th><th>R1</th><th>R2</th><th>R3</th></tr>
<tr><td>P1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>6</td><td>8</td></tr>
<tr><td>P2</td><td>2</td><td>0</td><td>3</td><td>4</td><td>3</td><td>3</td></tr>
<tr><td>P3</td><td>1</td><td>2</td><td>4</td><td>3</td><td>4</td><td>4</td></tr>
</table>
a. Find need matrix?
b. Is the current allocation is safe?
c. Can the request made by the process P1(1,1,0) can be granted?</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td colspan="5">PART-B</td></tr>
<tr><td>3(a)</td><td>Identify the use of binary and counting semaphore and write the C structure of wait () and signal ( ) operations.</td><td>6</td><td>CO2</td><td>K3</td></tr>
</table>

(b)	**Apply** FCFS, LRU, Optimal for following reference strings 1,2,3,4,1,2,5,1,2,3,4,5 and find the number of page faults.	**6**	**CO4**	**K3**
	OR			
4(a)	**Build** the solution to Dinning philosopher problem using monitors.	**6**	**CO2**	**K3**
(b)	**Identify** the steps involved page fault using a neat diagram.	**6**	**CO4**	**K3**

Course In-charge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

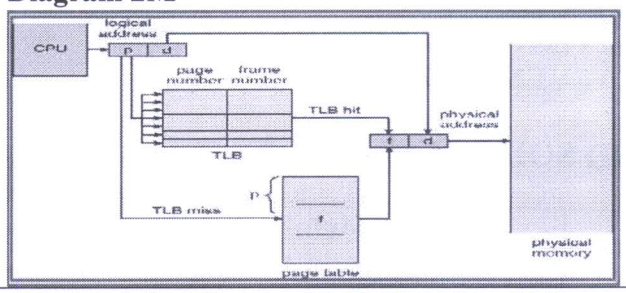
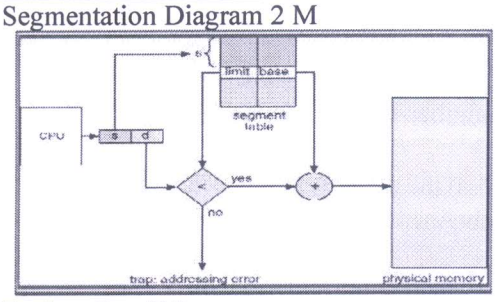
II SESSIONAL TEST QUESTION PAPER 2020-21

EVEN SEMESTER

SCHEME AND SOLUTION –SET A

Degree	:	**B.E**	**Semester**	:	**IV**
Branch	:	**CSE**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
	PART-A	
1(a)	• **Internal and External Address Space (3M)** • **Internal Fragmentation** there is wasted space internal to a portion due to the fact that block of data loaded is smaller than the partition. *Eg*:-If there is a block of 50kb and if the process requests 40kb and if the block is allocated to the process then there will be 10kb of memory left. • **External Fragmentation** exists when there is enough memory space exists to satisfy the request, but it not contiguous i.e., storage is fragmented in to large number of small holes. External Fragmentation may be either minor or a major problem **Logical and Physical Address Space (3M)** **Logical Address Space:** The address generated by the CPU is called logical address or virtual address. The address seen by the memory unit i.e., the one loaded in to the memory register is called the physical address. Set of logical address space generated by the programs is the logical address space. **Physical Address Space**: Set of physical address corresponding to these logical addresses is the physical address space. The mapping of virtual address to physical address during run time is done by the hardware device called memory management unit.	3+3=6M
(b)	**Explanation 4M** • When an associative register is presented with an item, it is compared with all the key values, if foundthe corresponding value field is return and searching is fast. TLB is used with the page table as follows: • TLB contains only few page table entries. • When a logicaladdress is generated by the CPU, its page number along with the frame number is added to TLB. If the page number is found its frame memory is used to access the actual memory. • If the page number is not in the TLB (TLB miss) the memory reference to the page table is made. When the frame number is obtained use can use it to access the memory. • If the TLB is full of entries the OS must select anyone for replacement. • Each time a new page table isselected the TLB must be flushed [erased] to ensure that next executing process do not use wrong information. • The percentage of time that a pagenumber is found in the TLB is called HIT ratio.	4+2=6M

	Diagram 2M  	
(c)	System is in safe state <P0,P2,P3,P4,P1> (3M) At time t1 P2 makes additional request for resource C <0,0,1> (2M) System is in deadlock state after additional request. The deadlocked processes are P1,P2,P3,P4 (1M)	3+2+1=6M
2(a)	Deadlock occurs when there is a cycle in RAG. If first process is requesting for a resource held to second process and second process is waiting for resource held by first process, then deadlock occurs 1M Four Conditions 4M **Necessary Conditions**:A deadlock situation can occur if the following 4 conditions occur simultaneously in asystem:- . **Mutual Exclusion**: Only one process must hold the resource at a time. If any other process requestsfor the resource, the requesting process must be delayed until the resource has been released. . **Hold and Wait**:-A process must be holding at least one resource and waiting to acquire additionalresources that are currently being held by the other process. . **No Preemption**:-Resources can't be preempted i.e., only the process holding the resources must release itafter the process has completed its task. . **Circular Wait**:-A set {P0,P1 .....Pn} of waiting process must exist such that P0 is waiting for a resource i.e., held by P1, P1 is waiting for a resource i.e., held by P2. Pn-1 is waiting for resource held by processPn and Pn is waiting for the resource i.e., held by P1. All the four conditions must hold for a deadlock to occur 1M	1+4+1=6M
(b)	Segmentation Diagram 2 M  Explanation 4M	2+4=6M
(c)	Need Matrix P1<1,4,5>, P2<2,3,0>,P3<2,2,0> 2M System is in safe state: <P1,P2,P3> 2M request made by the process P1(1,1,0) can be granted with safe sequence<P1,P2,P3> 2M	2+2+2=6M
	PART- B	

<table>
<tr><td>3(a)</td><td>

- **Counting semaphore** – integer value can range over an unrestricted domain
- **Binary semaphore** – integer value can range only between 0 and 1
  - Same as a **mutex lock**
- Can solve various synchronization problems
- Consider P_1 and P_2 that require S_1 to happen before S_2

Create a semaphore "**synch**" initialized to 0

P1:
S1;
signal(synch);
P2:
wait(synch);
S2;

- Can implement a counting semaphore ***S*** as a binary semaphore
- Mutual-exclusion implementation with semaphores

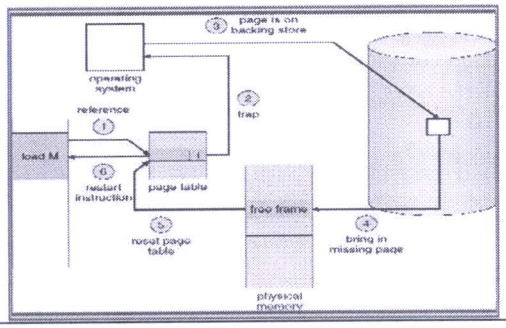
```
do {
        waiting(mutex);
                // critical section
        signal(mutex);
                // remainder section
}while (TRUE);
```

</td><td>6M</td></tr>
<tr><td>3(b)</td><td>No of Page Faults using FIFO for 3 Frames= 9
No of Page Faults using LRU for 3 Frames= 10
No of Page Faults using Optimal for 3 Frames=7</td><td>2+2+2=6M</td></tr>
<tr><td>4(a)</td><td>

```
monitor dp
{
enum {THINKING, HUNGRY, EATING} state [5]
condition self [5] ;
void pickup(int i) {
        state [i] = HUNGRY;
        test (i) ;
        if (state [i] != EATING)
        self [i] .wait() ;
        }
void putdown(int i) {
        state [i] = THINKING;
        test((i + 4) % 5} ;
        test( (i + 1) % 5) ;
        }
void test(int i) {
if ((state [(i + 4) % 5] != EATING) && (state [i] == HUNGRY) && (state [(i +
1) % 5] != EATING)) {
                state [i] = EATING;
                self [i].signal() ;
                }
initialization-code ()
{
        for (int i = 0; i < 5; i++)
```

</td><td>6M</td></tr>
</table>

	state [i] = THINKING;
	}
	}	

(b)	A page fault causes the following sequence to occur: 4M 1. Trap to the OS. 2. Save the user registers and process state. 3. Determine that the interrupt was a page fault. 4. Checks the page references were legal and determine the location of page on disk. 5. Issue a read from disk to a free frame. 6. If waiting, allocate the CPU to some other user. 7. Interrupt from the disk. 8. Save the registers and process states of other users. 9. Determine that the interrupt was from the disk. 10. Correct the page table and other table to show that the desired page is now in memory. 11. Wait for the CPU to be allocated to this process again. 12. Restore the user register process state and new page table then resume the interrupted instruction. Diagram 2M  	4+2=6M

Course incharge

Module Coordinator

HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

II SESSIONAL TEST QUESTION PAPER 2020-21 EVEN SEMESTER

SET – B

USN | | | | | | | | | | |

Degree : B.E
Branch : Computer Science & Engineering
Course Title : Operating Systems
Duration : 90 Minutes

Semester : IV
Course Code : 18CS43
Date : 27-06-2021
Max Marks : 30

Note: Answer ONE full question from each part.

<table>
<tr><th>Q No.</th><th>Question</th><th>Marks</th><th>CO mapping</th><th>K-Level</th></tr>
<tr><td colspan="5">PART-A</td></tr>
<tr><td>1(a)</td><td>Identify how deadlock occurs and the necessary conditions for deadlock occurrence. Indicate how many of these conditions should occur for deadlock to happen?</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td>(b)</td><td>Identify the principle behind paging. Explain its operation using a paging hardware clearly indicating how the logical addresses are converted to physical addresses.</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td>(C)</td><td>The operating system contains three resources. The numbers of instances of each resource type are (7, 7,10). The current allocation state is given below.
<table>
<tr><th>Process</th><th colspan="3">Allocation</th><th colspan="3">Max</th></tr>
<tr><th></th><th>R1</th><th>R2</th><th>R3</th><th>R1</th><th>R2</th><th>R3</th></tr>
<tr><td>P1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>6</td><td>8</td></tr>
<tr><td>P2</td><td>2</td><td>0</td><td>3</td><td>4</td><td>3</td><td>3</td></tr>
<tr><td>P3</td><td>1</td><td>2</td><td>4</td><td>3</td><td>4</td><td>4</td></tr>
</table>
a. Is the current allocation is safe?
b. Find need matrix?
c. Can the request made by the process P1(1,1,0) can be granted?</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>2(a)</td><td>Construct hardware support for TLB paging scheme with a neat diagram and explain in detail.</td><td>6</td><td>CO3</td><td>K3</td></tr>
<tr><td>(b)</td><td>Develop comparison study between the following:
a) Logical and physical address space. b) Internal and External fragmentation</td><td>6</td><td>CO3</td><td></td></tr>
<tr><td>(C)</td><td>A system has 5 processes; P0, P1, P2, P3 and P4.There are 3 types of resources R1, R2 and R3.there are 10 instances of R1, 5 instances of R2 and 7 instances of R3. At time T0, the situation is as follows:
<table>
<tr><th>Process</th><th colspan="3">Allocation</th><th colspan="3">Max Required</th></tr>
<tr><th></th><th>R1</th><th>R2</th><th>R3</th><th>R1</th><th>R2</th><th>R3</th></tr>
<tr><td>P0</td><td>0</td><td>1</td><td>0</td><td>6</td><td>5</td><td>3</td></tr>
<tr><td>P1</td><td>2</td><td>0</td><td>0</td><td>3</td><td>2</td><td>2</td></tr>
<tr><td>P2</td><td>3</td><td>0</td><td>2</td><td>9</td><td>0</td><td>2</td></tr>
<tr><td>P3</td><td>2</td><td>1</td><td>1</td><td>2</td><td>2</td><td>2</td></tr>
<tr><td>P4</td><td>0</td><td>0</td><td>2</td><td>4</td><td>3</td><td>3</td></tr>
</table>
i) Calculate Need and Available matrix.
ii) Is the system in a safe state at time T0? If yes give the safe Sequence.
iii) Suppose at time T1, process P1 requests (1,1,1) additional instance of resource type R1,R2 and R3, supposing the request if granted will the system be in a safe state?</td><td>6</td><td>CO3</td><td>K3</td></tr>
</table>

	PART-B			
3(a)	**Apply** TestAndSet() and Swap() instructions for implementation of mutual exclusion.	**6**	**CO2**	**K3**
(b)	**Identify** the steps involved page fault using a neat diagram.	**6**	**CO4**	**K3**
	OR			
4(a)	**Identify** and explain three requirements' that a solution to critical section problem must satisfy.	**6**	**CO2**	**K3**
(b)	**Solve** and calculate the page faults using FIFO and LRU for 3 and 4 page frames respectively for the following page references that occur 5,4,3,2,1,4,3,5,4,3,2,1,5.	**6**	**CO4**	**K3**

Course in charge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

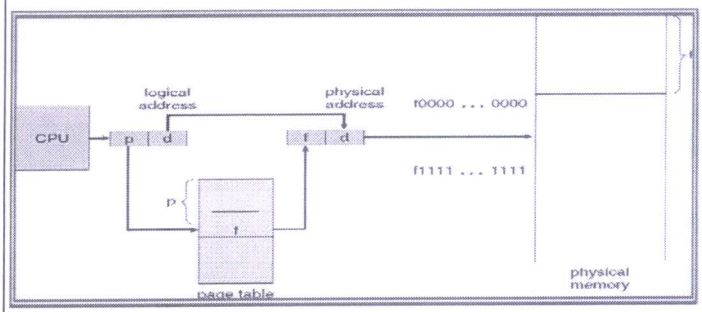
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

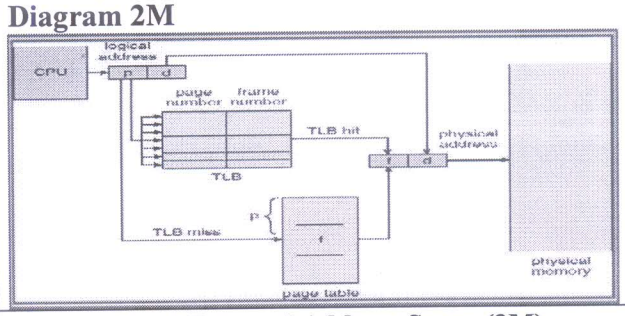
II SESSIONAL TEST QUESTION PAPER 2020-21

EVEN SEMESTER

SCHEME AND SOLUTION –SET B

Degree	:	**B.E**	**Semester**	:	**IV**
Branch	:	**CSE**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
	PART-A	
1(a)	Deadlock occurs when there is a cycle in RAG. If first process is requesting for a resource held to second process and second process is waiting for resource held by first process, then deadlock occurs 1M Four Conditions 4M **Necessary Conditions**: A deadlock situation can occur if the following 4 conditions occur simultaneously in a system:- . **Mutual Exclusion**: Only one process must hold the resource at a time. If any other process requestsfor the resource, the requesting process must be delayed until the resource has been released. . **Hold and Wait**:-A process must be holding at least one resource and waiting to acquire additionalresources that are currently being held by the other process. . **No Preemption**:-Resources can't be preempted i.e., only the process holding the resources must release itafter the process has completed its task. . **Circular Wait**:-A set {P0,P1 ..... Pn} of waiting process must exist such that P0 is waiting for a resource i.e., held by P1, P1 is waiting for a resource i.e., held by P2. Pn-1 is waiting for resource held by processPn and Pn is waiting for the resource i.e., held by P1. All the four conditions must hold for a deadlock tooccur	1+4+1=6M
(b)	Diagram 2M  Explanation 4M	2+4=6M
(c)	• Need Matrix P1<1,4,5>, P2<2,3,0>,P3<2,2,0> 2M • System is in safe state. Safe Sequence is <p2,P3,P1> 3M • The request p1<1,1,0> can not be granted as satisfying the request will lead to deadlock.1M	2+3+1=6M
2(a)	**Explanation 4M** • When an associative register is presented with an item, it is compared with all the key values, if foundthe corresponding value field is return and searching is fast. TLB is used	4+2=6M

	with the page table as follows: • TLB contains only few page table entries. • When a logicaladdress is generated by the CPU, its page number along with the frame number is added to TLB. If the page number is found its frame memory is used to access the actual memory. • If the page number is not in the TLB (TLB miss) the memory reference to the page table is made. When the frame number is obtained use can use it to access the memory. • If the TLB is full of entries the OS must select anyone for replacement. • Each time a new page table isselected the TLB must be flushed [erased] to ensure that next executing process do not use wrong information. • The percentage of time that a pagenumber is found in the TLB is called HIT ratio. **Diagram 2M** 	
(b)	• **Internal and External Address Space (3M)** • **Internal Fragmentation** there is wasted space internal to a portion due to the fact that block of data loaded is smaller than the partition. *Eg*:-If there is a block of 50kb and if the process requests 40kb and if the block is allocated to the process then there will be 10kb of memory left. • **External Fragmentation** exists when there is enough memory space exists to satisfy the request, but it not contiguous i.e., storage is fragmented in to large number of small holes. External Fragmentation may be either minor or a major problem **Logical and Physical Address Space (3M)** **Logical Address Space:** The address generated by the CPU is called logical address or virtual address. The address seen by the memory unit i.e., the one loaded in to the memory register is called the physical address. Set of logical address space generated by the programs is the logical address space. **Physical Address Space**: Set of physical address corresponding to these logical addresses is the physical address space. The mapping of virtual address to physical address during run time is done by the hardware device called memory management unit.	3+3=6M
(c)	Need Matrix P0<6,4,3> P1<1,2,2> P2<6,0,0> P3<0,0,1>P4<4,3,1> 2M System is in safe state and safe sequence is < P1,P3,P4,P0,P2>3M Request P1<1,1,1> can be granted and the safe sequence is < P3,P4,P1,P2,P0> 1M	2+3+1=6M
	PART- B	
3(a)	Definition: **test_and_set Instruction** **boolean test_and_set (boolean *target)** **{** **boolean rv = *target;** ***target = TRUE;** **return rv:** **}** 1. Executed atomically 2. Returns the original value of passed parameter	6M

3. Set the new value of passed parameter to "TRUE".

Shared Boolean variable lock, initialized to FALSE

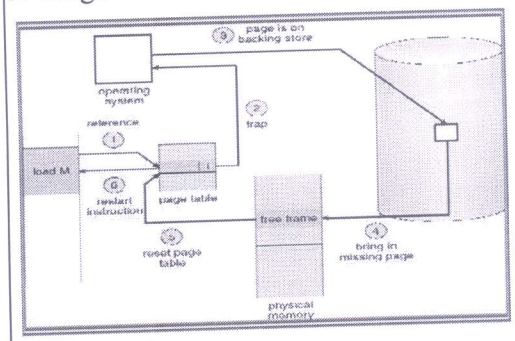
Solution:

```
do {
    while (test_and_set(&lock))
        ; /* do nothing */
            /* critical section */
    lock = false;
            /* remainder section */
} while (true);
```

(b) 4+2=6M

A page fault causes the following sequence to occur: 4M

. Trap to the OS.
. Save the user registers and process state.
. Determine that the interrupt was a page fault.
. Checks the page references were legal and determine the location of page on disk.
. Issue a read from disk to a free frame.
. If waiting, allocate the CPU to some other user.
. Interrupt from the disk.
. Save the registers and process states of other users.
. Determine that the interrupt was from the disk.
0. Correct the page table and other table to show that the desired page is now in memory.
1. Wait for the CPU to be allocated to this process again.
2. Restore the user register process state and new page table then resume the interrupted instruction.
3. Diagram 2M



4(a) 2+2+2=6M

Mutual Exclusion2M -If process Pi is executing in its critical section, then no other processes can be executingin their critical sections

Progress 2M -If no process is executing in its critical section and there exist some processes that wish to entertheir critical section, then the selection of the processes that will enter the critical section next cannot bepostponed indefinitely

	Bounded Waiting 2M-A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.	
(b)	No of Page Faults using FIFO for 3 Frames=10 No of Page Faults using LRU for 3 Frames=11 No of Page Faults using FIFO for 4 Frames=9 No of Page Faults using LRU for 4 Frames=7	6M

Course incharge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

III SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

USN | | | | | | | | | | |

SET – A

Degree	:	B.E	Semester:	IV
Branch	:	Computer Science and Engineering	Subject Code:	18CS43
Course Title	:	OPERATING SYSTEMS	Date:	6-8-2021
Duration	:	90 Minutes	Max Marks:	30

Note: Answer ONE full question from each part.

Q No.	Questions	Marks	CO mapping	K-Level
	PART-A			
1(a)	Identify how process is managed on Linux platform	6	CO5	(Applying) K3
(b)	Construct access matrix of domain and objects with copy, switch and control rights.	6	CO5	(Applying) K3
(c)	Suppose that the disk has 2000 cylinders named 0 to 1999. The R/W head is currently serving at cylinder 125. The queue of pending requests in order: 1270, 1213, 1174, 1148, 1090, 1122, 1250, 1130 starting from the current head position. What is the total distance travelled (in cylinders) by the disk arm to satisfy the requests using algorithms FCFS, C-SCAN and LOOK. Illustrate with figure in each case.	6	CO5	(Applying) K3
	OR			
2(a)	Apply any two types of free space management techniques to a list of free blocks on disk and mention its advantages and disadvantages.	6	CO5	(Applying) K3
(b)	Identify the different components of a Linux system	6	CO5	(Applying) K3
(c)	Suppose that the disk has 500 cylinders named 0 to 1999. The R/W head is currently serving at cylinder 125. The queue of pending requests in order: 270, 213, 174, 148, 90, 122, 250, 130 starting from the current head position. What is the total distance travelled (in cylinders) by the disk arm to satisfy the requests using algorithms FCFS, SCAN and C-LOOK. Illustrate with figure in each case.	6	CO5	(Applying) K3
	PART-B			
3(a)	Make use of an example to differentiate different file access methods to access file from disk.	6	CO4	(Applying) K3
(b)	Utilize an example and design any three types of directory structures.	6	CO4	(Applying) K3
	OR			
4(a)	Identify the different attributes of a file and operations performed on files.	6	CO4	(Applying) K3
(b)	Utilize contiguous, Linked and Index methods and explain file allocation.	6	CO4	(Applying) K3

Course In Charge | Module Coordinator | HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

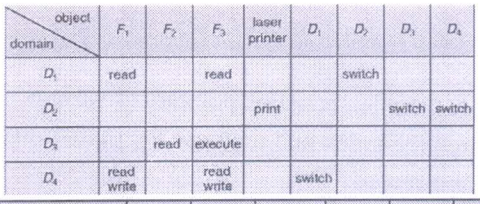
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

III SESSIONAL TEST QUESTION PAPER 2020-21

EVEN SEMESTER

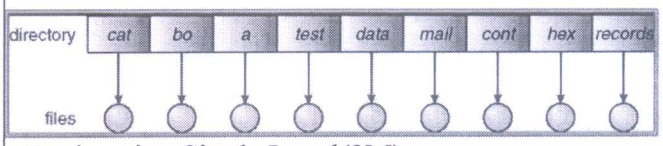
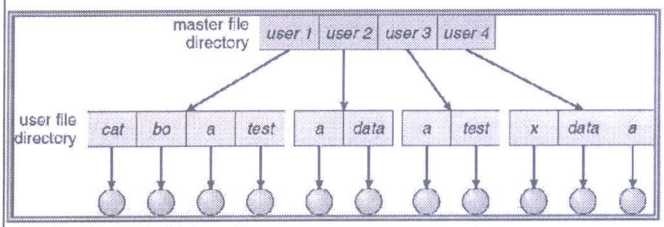
SCHEME AND SOLUTION –SET A

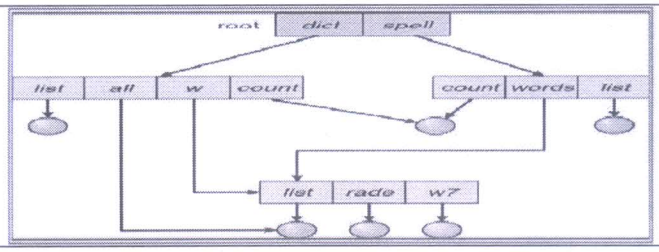
Degree	:	**B.E**	**Semester**	:	**IV**
Branch	:	**CSE**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
	PART-A	
1(a)	o UNIX process management separates the creation of processes and the running of a new program into two distinct operations. **Process Creation and execution 2M** **Process identity 2M** **Process environment 2M**	2+2+2=6M
(b)	• View protection as a matrix (*access matrix*). Rows represent domains. Columns represent objects *Access(i, j)* is the set of operations that a • process executing in Domain$_i$ can invoke on Object$_j$ • Access-right= where rights-set is a subset of all valid operations that can be performed on the object.  Copy-2M Switch-2M Control-2M	2+2+2=6M
(c)	FCFS= 1594(2M), C-SCAN=1934(2M), LOOK=1934(2M)	2+2+2=6M
2(a)	Bit vector- A Vector that maintains the list of free blocks, indicating if bit[i]=1 page in memory or else on disk. Advantages and Disadvantages 2M Linked list of free blocks 2M, Grouping 2M	2+2+2=6M
(b)	Diagram 2M Explanation 4M The system libraries define a standard set of functions through which applications interact with the kernel, and which implement much of the	2+4=6M

object / domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

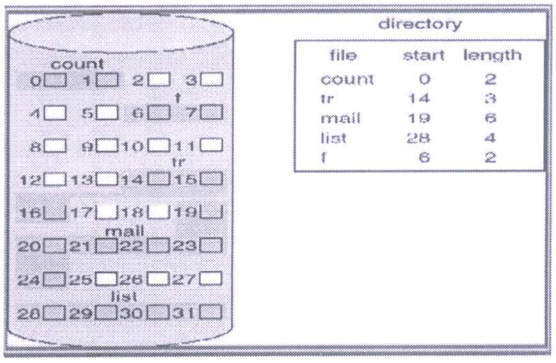
object / domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch control
D_3		read	execute					
D_4	write		write		switch			

	operating-system functionality that does not need the full privileges ofkernel code The system utilities perform individual specialized management tasks x Sections of kernel code that can be compiled, loaded, and unloaded independent of the rest of the kernel KERNEL MODULES Three components to Linux module support: module management driver registration conflict resolution	
(c)	FCFS=605 Head movement 2M SSTF=331 Head movement 2M SCAN= 395 2M	2+2+2=6M
	PART- B	
3(a)	Sequential Access 2M Sequential access is the simplest access method. Information in the file is processed in order, one record after another. A read operation reads the next portion of the file and automatically advances a file pointer. Write operation appends to the end of the file and such a file can be next to the beginning. Example Direct Access 2M Direct access allows random access to any file block. This method is based on disk model of a file.A file is made up of fixed length logical records. It allows the program to read and write records rapidly in any order.A direct access file allows arbitrary blocks to be read or written. Example Indexed Access 2M The index is like an index at the end of a book which contains pointers to various blocks. x To find a record in a file, we search the index and then use the pointer to access the file directly and to find the desired record	2+2+2=6M
(b)	 Explanation Single Level(2M)  Two Level 2M Tree Level directory Structure 2M	2+2+2=6M

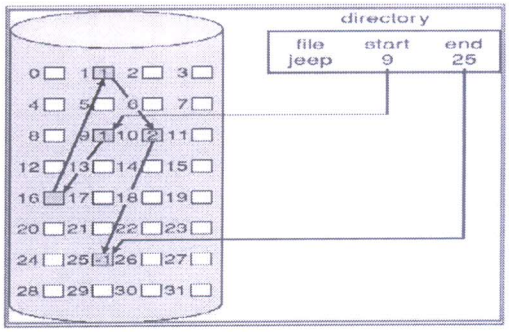
<table>
<tr><td></td><td> 
</td><td></td></tr>
<tr><td>4(a)</td><td><u>File Attributes</u>:-File attributes varies from one OS to other. The common file attributes are:

<u>Name</u>:-The symbolic file name is the only information kept in human readable form.
<u>Identifier</u>:-The unique tag, usually a number, identifies the file within the file system. It is the non-
readable name for a file.

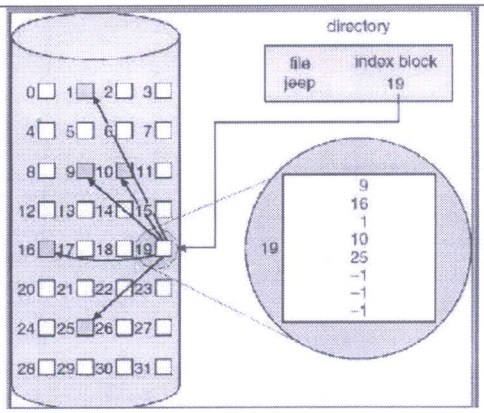
<u>Type</u>:-This information is needed for those systems that supports different types.
<u>Location</u>:-This information is a pointer to a device and to the location of the file on that device.
<u>Size</u>:-The current size of the file and possibly the maximum allowed size are included in this attribute.
<u>Protection</u>:-Access control information determines who can do reading, writing, execute and so on.
<u>Time, data and User Identification</u>:-This information must be kept for creation, last modification and
last use. These data are useful for protection, security and usage monitoring.</td><td>6M</td></tr>
<tr><td>(b)</td><td>Continuous Allocation 2M

Linked Allocation 2M

Indexed Allocation 2M</td><td>2+2+2=6M</td></tr>
</table>



Course incharge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

III SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

USN | | | | | | | | | | |

SET – B

Degree	:	B.E	Semester:	IV
Branch	:	Computer Science and Engineering	Subject Code:	18CS43
Course Title	:	OPERATING SYSTEMS	Date:	6-8-2021
Duration	:	90 Minutes	Max Marks:	30

Note: Answer ONE full question from each part.

Q No.	Questions	Marks	CO mapping	K-Level
	PART-A			
1(a)	Construct access matrix of domain and objects with copy, transfer and owner rights.	6	CO5	(Applying) K3
(b)	Identify how process is managed on Linux platform	6	CO5	(Applying) K3
(c)	Suppose that the disk has 5000 cylinders named 0 to 4999. The R/W head is currently serving at cylinder 125. The queue of pending requests in order: 1470, 913, 1774, 948, 1509, 1022, 1750, 130 starting from the current head position. What is the total distance travelled (in cylinders) by the disk arm to satisfy the requests using algorithms FCFS, SCAN and LOOK. Illustrate with figure in each case.	6	CO5	(Applying) K3
	OR			
2(a)	Identify the different components of a Linux system	6	CO5	(Applying) K3
(b)	Apply any two types of free space management techniques to a list of free blocks on disk and mention its advantages and disadvantages.	6	CO5	(Applying) K3
(c)	Suppose that the disk has 300 cylinders named 0 to 299. The R/W head is currently serving at cylinder 125. The queue of pending requests in order: 270, 213, 174, 148, 90, 122, 250, 130 starting from the current head position. What is the total distance travelled (in cylinders) by the disk arm to satisfy the requests using algorithms FCFS, SSTF and SCAN. Illustrate with figure in each case.	6	CO5	(Applying) K3
	PART-B			
3(a)	Make use of an example to differentiate different file access methods to access file from disk.	6	CO4	(Applying) K3
(b)	Utilize an example and design any three types of directory structures.	6	CO4	(Applying) K3
	OR			
4(a)	Utilize contiguous, Linked and Index methods and explain file allocation.	6	CO4	(Applying) K3
(b)	Make use of relevant diagram and explain the thrashing phenomenon and working set model.	6	CO4	(Applying) K3

Course Incharge | Module Coordinator | HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

III SESSIONAL TEST QUESTION PAPER 2020-21

EVEN SEMESTER

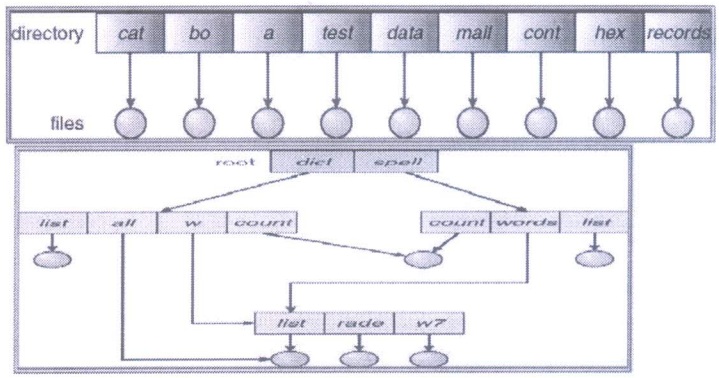
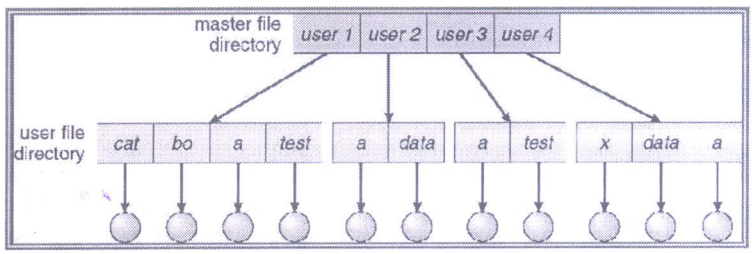
SCHEME AND SOLUTION –SET B

Degree	:	**B.E**	**Semester**	:	**IV**
Branch	:	**CSE**	**Course Code**	:	**18CS43**
Course Title	:	**Operating Systems**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
	PART-A	
1(a)	• View protection as a matrix (*access matrix*). Rows represent domains. Columns represent objects *Access(i, j)* is the set of operations that a process executing in Domain$_i$ can invoke on Object$_j$ • Access-right= where rights-set is a subset of all valid operations that can be performed on the object. • Copy 2M • Transfer 2M • Owner 2M	2+2+2=6M

object / domain	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch control
D_3		read	execute					
D_4	write		write		switch			

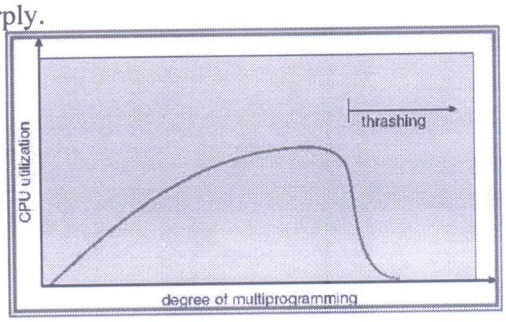
(b)	Bit vector- A Vector that maintains the list of free blocks, indicating if bit[i]=1 page in memory or else on disk. Advantages and Disadvantages 2M Linked list of free blocks 2M, Grouping 2M	2+2+2=M
(c)	FCFS=6985 2M SCAN=1649 2M LOOK=1649 2M	2+2+2=6M
2(a)	Diagram 2M Explanation 4M The system libraries define a standard set of functions through which applications interact with the kernel, and which implement much of the operating-system functionality that does not need the full privileges ofkernel code The system utilities perform individual specialized management tasks x Sections of kernel codethat can be compiled, loaded, and unloaded independent of the rest of the kernel KERNEL MODULES Three components to Linux module support: module management	2+4=6M

	driver registration conflict resolution	
(b)	UNIX process management separates the creation of processes and the running of a new program into two distinct operations. Process Creation and execution 2M Process identity 2M Process environment 2M	2+2+2=6M
(c)	FCFS=605 2M SSTF=328 2M SCAN=328 2M	2+2+2=6M
	PART- B	
3(a)	Sequential Access 2M Direct Access 2M Indexed Access 2M	2+2+2=6M
(b)	Continuous Allocation 2M Linked Allocation 2M Index Allocation 2M	2+2+2=6M
4(a)	  Explanation and diagram for Single Level 2M Two Level 2M Tree Level directory Structure 2M	2+2+2=6M
(b)	<u>Cause of Thrashing</u> 2M Thrashing results in severe performance problem. The operating system monitors the cpu utilization is low. We increase the	2+4=6M

degree of multiprogramming by introducing new process to the system.

A global page replacement algorithm replaces pages with no regards to the process to whichthey belong.The figure shows the thrashing.

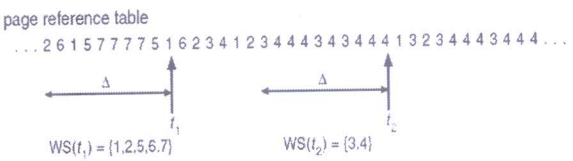
As the degree of multi programming increases, more slowly until a maximum is reached. If the degree of multi programming is increased further thrashing sets in and the cpu utilization drops sharply.



Working set model 4M

Working set model algorithm uses the current memory requirements to determine the number of page framesto allocate to the process, an informal definition is "the collection of pages that a process is working with andwhich must be resident if the process to avoid thrashing". The idea is to use the recent needs of a process to predict its future reader. The working set is an approximation of programs locality. Ex: given a sequence of memory reference, if the working set window size to memory references, then working set at time t1 is

{1,2,5,6,7} and at t2 is changed to {3,4}



Course incharge

Module Coordinator

HOD

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV/A
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	Final Consolidated IA

Sl. NO.	USN	NAME	IA 1 (30M)	Assign 1 (10M)	IA 2 (30M)	Assign 2 (10M)	IA 3 (30M)	Assign 3 (10M)	Avg IA (30M)	Avg Assign (10M)	Final IA (30M)	Final Assign (10)	Final (40M)	Sign
1	1KS19CS001	AAKRITI	20	10	29	10	27	10	25.33	10.00	26	10	36	
2	1KS19CS002	ABHISHEK B	27	10	26	10	27	10	26.67	10.00	27	10	37	
3	1KS19CS003	ABHISHEK YADAV	24	10	26	10	21	10	23.67	10.00	24	10	34	
4	1KS19CS004	ADITH KARTHIK RAJU	26	10	26	10	27	10	26.33	10.00	27	10	37	
5	1KS19CS005	AJAY S KALBURGI	28	10	26	10	27	10	27.00	10.00	27	10	37	
6	1KS19CS006	AKASH A S	27	10	26	10	26	10	26.33	10.00	27	10	37	
7	1KS19CS007	AMAN KUSHWAHA	26	10	26	10	27	10	26.33	10.00	27	10	37	
8	1KS19CS009	AMIT K B	19	10	26	10	30	10	25.00	10.00	25	10	35	
9	1KS19CS010	AMOGHA H S	27	10	26	10	26	10	26.33	10.00	27	10	37	
10	1KS19CS011	AMRUTHA K H	29	10	26	10	29	10	28.00	10.00	28	10	38	
11	1KS19CS012	ANAGHA A HEBBAR	25	10	27	10	26	10	26.00	10.00	26	10	36	
12	1KS19CS014	ANUSHA B	25	10	29	10	30	10	28.00	10.00	28	10	38	

Sl. NO.	USN	NAME	IA 1 (30M)	Assign 1 (10M)	IA 2 (30M)	Assign 2 (10M)	IA 3 (30M)	Assign 3 (10M)	Avg IA (30M)	Avg Assign (10M)	Final IA (30M)	Final Assign (10)	Final (40M)	Sign
13	1KS19CS015	AQSA AQEEL	28	10	25	10	30	10	27.67	10.00	28	10	38	
14	1KS19CS016	ASHIKA H N	30	10	27	10	29	10	28.67	10.00	29	10	39	
15	1KS19CS017	BHOOMIKA A M	30	10	29	10	30	10	29.67	10.00	30	10	40	
16	1KS19CS018	BHOOMIKA K	28	10	25	10	30	10	27.67	10.00	28	10	38	
6	1KS19CS019	BHUMIKA M	27	10	29	10	30	10	28.67	10.00	29	10	39	
18	1KS19CS020	C N SHREYAS	27	10	29	10	27	10	27.67	10.00	28	10	38	
19	1KS19CS021	CHAITANYA SHIVARAJU	29	10	29	10	30	10	29.33	10.00	30	10	40	
20	1KS19CS022	CHANDAN B V	19	10	25	10	23	10	22.33	10.00	23	10	33	
21	1KS19CS023	DEEKSHA NAIDU R	28	10	29	10	27	6	28.00	6.67	28	7	35	
22	1KS19CS024	DEEPA G	27	10	26	10	27	10	26.67	10.00	27	10	37	
23	1KS19CS025	DEEPTHI.N.K	25	10	27	10	27	10	26.33	10.00	27	10	37	
24	1KS19CS026	DEVI PRASAD N	27	10	28	10	29	10	28.00	10.00	28	10	38	
25	1KS19CS028	DHEEMANTH G	28	10	27	10	27	10	27.33	10.00	28	10	38	
26	1KS19CS029	DINESH M	23	10	26	10	27	10	25.33	10.00	26	10	36	
27	1KS19CS030	G PRERITHA	26	10	26	10	27	10	26.33	10.00	27	10	37	
28	1KS19CS031	GAGAN REDDY S	19	10	24	10	23	10	22.00	10.00	22	10	32	
29	1KS19CS032	GAGANDEEP K	26	10	25	10	26	10	25.67	10.00	26	10	36	
30	1KS19CS033	GEETHANJALI B K PRASAD	26	10	27	10	30	10	27.67	10.00	28	10	38	
31	1KS19CS034	GULSHAN KUMAR S	27	10	26	10	27	10	26.67	10.00	27	10	37	
32	1KS19CS035	HARSHITHA J	29	10	26	10	30	10	28.33	10.00	29	10	39	
33	1KS19CS036	INDRAJITH H M	27	10	28	10	30	10	28.33	10.00	29	10	39	
34	1KS19CS038	K KISHAN	26	10	29	8	30	10	28.33	9.33	29	10	39	
35	1KS19CS039	K R VAGEESH	27	10	29	10	24	10	26.67	10.00	27	10	37	
36	1KS19CS040	KALYAN CHOWDHARY B	20	10	26	10	27	10	24.33	10.00	25	10	35	
37	1KS19CS041	KARTHIK S MORAJKAR	30	10	26	10	30	10	28.67	10.00	29	10	39	

Sl. NO.	USN	NAME	IA 1 (30M)	Assign 1 (10M)	IA 2 (30M)	Assign 2 (10M)	IA 3 (30M)	Assign 3 (10M)	Avg IA (30M)	Avg Assign (10M)	Final IA (30M)	Final Assign (10)	Final (40M)	Sign
38	1KS19CS042	KAVYASHREE.S.L	28	10	29	10	28	10	28.33	10.00	29	10	39	Kavya
39	1KS19CS043	KEERTHAN GOWDA S	25	10	27	10	30	10	27.33	10.00	28	10	38	
40	1KS19CS044	KOTHAPALLI SREEJA	21	10	28	10	30	10	26.33	10.00	27	10	37	
41	1KS19CS045	KOTTALA SAIVENKATASUCHITH	26	10	26	0	22	10	24.67	6.67	25	7	32	Suchith
42	1KS19CS046	KRISHNA K R	26	10	26	10	24	10	25.33	10.00	26	10	36	K.R.K
43	1KS19CS047	KUMAR S	16	10	22	10	24	10	20.67	10.00	21	10	31	Kumar
44	1KS19CS048	LIKITH G	16	10	21	10	24	10	20.33	10.00	21	10	31	Likith
45	1KS19CS049	LISHA C	28	10	27	10	27	10	27.33	10.00	28	10	38	Lisha. C.
46	1KS19CS050	MAHAK SHREE	29	10	28	10	26	10	27.67	10.00	28	10	38	Mahak
47	1KS19CS051	MALLIPALLI SPURTHI REDDY	28	10	29	10	30	10	29.00	10.00	29	10	39	Spurthi.
48	1KS19CS052	MANASA G L	28	10	29	10	30	10	29.00	10.00	29	10	39	MANASA G L
49	1KS19CS053	MOHAMMED NOOR AMAN	24	10	27	10	27	10	26.00	10.00	26	10	36	MDNoor.
50	1KS19CS054	MUKESH KUMAR	21	10	26	10	27	10	24.67	10.00	25	10	35	Mukesh
51	1KS19CS055	MYTHREYI U	28	10	29	10	30	10	29.00	10.00	29	10	39	Mythreyi
52	1KS19CS056	N ASHOK	26	10	28	10	30	10	28.00	10.00	28	10	38	ASHOK
53	1KS19CS057	N BHAVYA	27	10	27	10	30	10	28.00	10.00	28	10	38	
54	1KS19CS058	N P SHASHANKA RAO	26	10	27	10	27	10	26.67	10.00	27	10	37	Shashank
55	1KS18CS011	BHARATH R	26	10	23	10	23	0	24.00	6.67	24	7	31	
56	1KS19CS116	VRATHIKA BILLAVA	19	10	26	10	24	2	23.00	7.33	23	8	31	Vrathika
57	1KS20CS400	AKIF DELVI	26	10	24	10	29	10	26.33	10.00	27	10	37	
58	1KS20CS404	PRANAV CHANDRAN P	16	10	21	10	28	10	21.67	10.00	22	10	32	Pranav C
59	1KS20CS402	KEERTHI KUMAR V	16	10	24	10	24	10	21.33	10.00	22	10	32	Keerthi

Mary

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV/A
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	I

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2		CO1	CO2
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	12 M
1	1KS19CS001	AAKRITI	5	5	4				6				20	14	6
2	1KS19CS002	ABHISHEK B	6	6	4						5	6	27	16	11
3	1KS19CS003	ABHISHEK YADAV	2	6	4						6	6	24	12	12
4	1KS19CS004	ADITH KARTHIK RAJU	6	6	4						4	6	26	16	10
5	1KS19CS005	AJAY S KALBURGI	6	5	6						5	6	28	17	11
6	1KS19CS006	AKASH A S	6	5	4						6	6	27	15	12
7	1KS19CS007	AMAN KUSHWAHA	5	6	5						6	4	26	16	10
8	1KS19CS009	AMIT K B	4	6	2				5	2			19	12	7
9	1KS19CS010	AMOGHA H S	6	6	4						5	6	27	16	11
10	1KS19CS011	AMRUTHA K H	6	6	5						6	6	29	17	12
11	1KS19CS012	ANAGHA A HEBBAR	6	6	4				6	3			25	16	9
12	1KS19CS014	ANUSHA B	4	6	4						5	6	25	14	11
13	1KS19CS015	AQSA AQEEL	6	6	5						6	5	28	17	11
14	1KS19CS016	ASHIKA H N	6	6	6						6	6	30	18	12

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2		CO1	CO2
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	12 M
15	1KS19CS017	BHOOMIKA A M	6	6	6						6	6	**30**	18	12
16	1KS19CS018	BHOOMIKA K	6	6	5						6	5	**28**	17	11
6	1KS19CS019	BHUMIKA M	6	6	3						6	6	**27**	15	12
18	1KS19CS020	C N SHREYAS	6	6	3						6	6	**27**	15	12
19	1KS19CS021	CHAITANYA SHIVARAJU	6	6	5						6	6	**29**	17	12
20	1KS19CS022	CHANDAN B V	6	4	5						4		**19**	15	4
21	1KS19CS023	DEEKSHA NAIDU R	6	6	4						6	6	**28**	16	12
22	1KS19CS024	DEEPA G	6	5	4						6	6	**27**	15	12
23	1KS19CS025	DEEPTHI.N.K	4	5	4						6	6	**25**	13	12
24	1KS19CS026	DEVI PRASAD N	6	6	4						6	5	**27**	16	11
25	1KS19CS028	DHEEMANTH G	6	5	5						6	6	**28**	16	12
26	1KS19CS029	DINESH M	6	3	4						4	6	**23**	13	10
27	1KS19CS030	G PRERITHA	6	4	4						6	6	**26**	14	12
28	1KS19CS031	GAGAN REDDY S	6	5	4						2	2	**19**	15	4
29	1KS19CS032	GAGANDEEP K	6	6	4						4	6	**26**	16	10
30	1KS19CS033	GEETHANJALI B K PRASAD	6	5	4						5	6	**26**	15	11
31	1KS19CS034	GULSHAN KUMAR S	6	6	4						5	6	**27**	16	11
32	1KS19CS035	HARSHITHA J	6	5	6						6	6	**29**	17	12
33	1KS19CS036	INDRAJITH H M	6	6	3						6	6	**27**	15	12
34	1KS19CS038	K KISHAN	6	6	2						6	6	**26**	14	12
35	1KS19CS039	K R VAGEESH	6	6	3						6	6	**27**	15	12
36	1KS19CS040	KALYAN CHOWDHARY B	4	4	4						6	2	**20**	12	8
37	1KS19CS041	KARTHIK S MORAJKAR	6	6	6						6	6	**30**	18	12
38	1KS19CS042	KAVYASHREE.S.L	6	6	5						5	6	**28**	17	11

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2		CO1	CO2
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	12 M
39	1KS19CS043	KEERTHAN GOWDA S	4	6	4				6	5			25	14	11
40	1KS19CS044	KOTHAPALLI SREEJA				1	3	6			5	6	21	10	11
41	1KS19CS045	KOTTALA SAIVENKATASUCHITH	4	6	4						6	6	26	14	12
42	1KS19CS046	KRISHNA K R	6	4	4						6	6	26	14	12
43	1KS19CS047	KUMAR S	4	5	4						3		16	13	3
44	1KS19CS048	LIKITH G	4	4	3						3		14	11	3
45	1KS19CS049	LISHA C	5	6	5						6	6	28	16	12
46	1KS19CS050	MAHAK SHREE	6	6	5						6	6	29	17	12
47	1KS19CS051	MALLIPALLI SPURTHI REDDY	6	6	4						6	6	28	16	12
48	1KS19CS052	MANASA G L				6	4	6			6	6	28	16	12
49	1KS19CS053	MOHAMMED NOOR AMAN	4	6	6				6	2			24	16	8
50	1KS19CS054	MUKESH KUMAR	4	6	4						5	2	21	14	7
51	1KS19CS055	MYTHREYI U	6	6	5						5	6	28	17	11
52	1KS19CS056	N ASHOK	6	6	4						4	6	26	16	10
53	1KS19CS057	N BHAVYA	6	6	4						5	6	27	16	11
54	1KS19CS058	N P SHASHANKA RAO	6	6	5				6	3			26	17	9
55	1KS18CS011	BHARATH R	5	5	4						6	6	26	14	12
56	1KS19CS116	VRATHIKA BILLAVA				4		4			5	6	19	8	11
57	1KS20CS400	AKIF DELVI	6	6	3						5	6	26	15	11
58	1KS20CS404	PRANAV CHANDRAN P	2	6	3						5		16	11	5
59	1KS20CS402	KEERTHI KUMAR V	6	5							5	0	16	11	5

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	II

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S		
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4		CO3	CO2	CO4
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	6 M	6 M
1	1KS19CS001	AAKRITI	6	6	6						6	5	29	18	6	5
2	1KS19CS002	ABHISHEK B				6	5	4			6	5	26	15	6	5
3	1KS19CS003	ABHISHEK YADAV				6	5	4			6	5	26	15	6	5
4	1KS19CS004	ADITH KARTHIK RAJU				6	5	4			6	5	26	15	6	5
5	1KS19CS005	AJAY S KALBURGI				6	5	4			6	5	26	15	6	5
6	1KS19CS006	AKASH A S				5	6	4			6	5	26	15	6	5
7	1KS19CS007	AMAN KUSHWAHA				6	5	4			6	5	26	15	6	5
8	1KS19CS009	AMIT K B				6	5	4			6	5	26	15	6	5
9	1KS19CS010	AMOGHA H S				6	5	4			6	5	26	15	6	5
10	1KS19CS011	AMRUTHA K H				6	5	4			6	5	26	15	6	5
11	1KS19CS012	ANAGHA A HEBBAR				5	5	6			6	5	27	16	6	5
12	1KS19CS014	ANUSHA B				6	6	6			6	5	29	18	6	5
13	1KS19CS015	AQSA AQEEL				6	5	4			5	5	25	15	5	5

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S		
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4		CO3	CO2	CO4
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	6 M	6 M
14	1KS19CS016	ASHIKA H N				6	5	4			6	6	27	15	6	6
15	1KS19CS017	BHOOMIKA A M				6	6	6			6	5	29	18	6	5
16	1KS19CS018	BHOOMIKA K	6	6	5				5	3			25	17	5	3
6	1KS19CS019	BHUMIKA M				6	6	6			6	5	29	18	6	5
18	1KS19CS020	C N SHREYAS				6	6	6			6	5	29	18	6	5
19	1KS19CS021	CHAITANYA SHIVARAJU				6	6	6			6	5	29	18	6	5
20	1KS19CS022	CHANDAN B V				6	5	4			5	5	25	15	5	5
21	1KS19CS023	DEEKSHA NAIDU R				6	6	6			6	5	29	18	6	5
22	1KS19CS024	DEEPA G				6	5	4			6	5	26	15	6	5
23	1KS19CS025	DEEPTHI.N.K				6	5	5			6	5	27	16	6	5
24	1KS19CS026	DEVI PRASAD N				6	5	6			6	5	28	17	6	5
25	1KS19CS028	DHEEMANTH G				6	6	4			6	5	27	16	6	5
26	1KS19CS029	DINESH M				5	6	4			6	5	26	15	6	5
27	1KS19CS030	G PRERITHA				6	5	4			6	5	26	15	6	5
28	1KS19CS031	GAGAN REDDY S				5	5	3			6	5	24	13	6	5
29	1KS19CS032	GAGANDEEP K				6	5	3			6	5	25	14	6	5
30	1KS19CS033	GEETHANJALI B K PRASAD				6	6	4			6	5	27	16	6	5
31	1KS19CS034	GULSHAN KUMAR S				6	5	4			6	5	26	15	6	5
32	1KS19CS035	HARSHITHA J				6	5	4			6	5	26	15	6	5
33	1KS19CS036	INDRAJITH H M				6	5	6			6	5	28	17	6	5
34	1KS19CS038	K KISHAN				6	6	6			6	5	29	18	6	5
35	1KS19CS039	K R VAGEESH				6	6	6			6	5	29	18	6	5
36	1KS19CS040	KALYAN CHOWDHARY B				6	5	4			6	5	26	15	6	5
37	1KS19CS041	KARTHIK S MORAJKAR				6	5	4			6	5	26	15	6	5

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S		
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4		CO3	CO2	CO4
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	6 M	6 M
38	1KS19CS042	KAVYASHREE.S.L				6	6	6			6	5	29	18	6	5
39	1KS19CS043	KEERTHAN GOWDA S				6	5	5			6	5	27	16	6	5
40	1KS19CS044	KOTHAPALLI SREEJA				6	6	4			6	6	28	16	6	6
41	1KS19CS045	KOTTALA SAIVENKATASUCHITH				5	5	5			6	5	26	15	6	5
42	1KS19CS046	KRISHNA K R				6	5	4			6	5	26	15	6	5
43	1KS19CS047	KUMAR S				6	5				6	5	22	11	6	5
44	1KS19CS048	LIKITH G				6	5				6	4	21	11	6	4
45	1KS19CS049	LISHA C				6	5	5			6	5	27	16	6	5
46	1KS19CS050	MAHAK SHREE				5	6	6			6	5	28	17	6	5
47	1KS19CS051	MALLIPALLI SPURTHI REDDY				6	6	6			6	5	29	18	6	5
48	1KS19CS052	MANASA G L				6	6	6			6	5	29	18	6	5
49	1KS19CS053	MOHAMMED NOOR AMAN				6	5	5			6	5	27	16	6	5
50	1KS19CS054	MUKESH KUMAR				5	6	4			6	5	26	15	6	5
51	1KS19CS055	MYTHREYI U				6	6	6			6	5	29	18	6	5
52	1KS19CS056	N ASHOK				6	6	5			6	5	28	17	6	5
53	1KS19CS057	N BHAVYA				6	5	4			6	6	27	15	6	6
54	1KS19CS058	N P SHASHANKA RAO				6	5	5			6	5	27	16	6	5
55	1KS18CS011	BHARATH R				6	4	2			6	5	23	12	6	5
56	1KS19CS116	VRATHIKA BILLAVA				6	5	2			6	5	26	13	6	5
57	1KS20CS400	AKIF DELVI	6	5	6						2	5	24	17	2	5
58	1KS20CS404	PRANAV CHANDRAN P				6	4				6	5	21	10	6	5
59	1KS20CS402	KEERTHI KUMAR V				6	5	2			6	5	24	13	6	5

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV/A
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	III

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO5	CO5	CO5	CO5	CO5	CO5	CO4	CO4	CO4	CO4		CO4	CO5
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	12 M	18 M
1	1KS19CS001	AAKRITI	6	6	3				6	6			27	12	15
2	1KS19CS002	ABHISHEK B	6	6	3				6	6			27	12	15
3	1KS19CS003	ABHISHEK YADAV	6	6	3						6		21	6	15
4	1KS19CS004	ADITH KARTHIK RAJU	6	6	3				6	6			27	12	15
5	1KS19CS005	AJAY S KALBURGI	6	6	3				6	6			27	12	15
6	1KS19CS006	AKASH A S	6	6	3				6	5			26	11	15
7	1KS19CS007	AMAN KUSHWAHA	6	6	3				6	6			27	12	15
8	1KS19CS009	AMIT K B			3	6	6	6	6	6			30	12	18
9	1KS19CS010	AMOGHA H S	6	6	3				6	5			26	11	15
10	1KS19CS011	AMRUTHA K H	6	6	5						6	6	29	12	17
11	1KS19CS012	ANAGHA A HEBBAR	6	6	3				6	5			26	11	15
12	1KS19CS014	ANUSHA B				6	6	6	6	6			30	12	18
13	1KS19CS015	AQSA AQEEL				6	6	6	6	6			30	12	18
14	1KS19CS016	ASHIKA H N				6	6	6	6	5			29	11	18
15	1KS19CS017	BHOOMIKA A M				6	6	6	6	6			30	12	18

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO5	CO5	CO5	CO5	CO5	CO5	CO4	CO4	CO4	CO4		CO4	CO5
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	12 M	18 M
16	1KS19CS018	BHOOMIKA K				6	6	6	6	6			30	12	18
6	1KS19CS019	BHUMIKA M	6	6	6				6	6			30	12	18
18	1KS19CS020	C N SHREYAS				6	6	3	6	6			27	12	15
19	1KS19CS021	CHAITANYA SHIVARAJU				6	6	6	6	6			30	12	18
20	1KS19CS022	CHANDAN B V	6	6					6	5			23	11	12
21	1KS19CS023	DEEKSHA NAIDU R	6	6	3				6	6			27	12	15
22	1KS19CS024	DEEPA G	6	6	3				6	6			27	12	15
23	1KS19CS025	DEEPTHI.N.K	6	6	3				6	6			27	12	15
24	1KS19CS026	DEVI PRASAD N				6	6	6	6	5			29	11	18
25	1KS19CS028	DHEEMANTH G	6	6	3				6	6			27	12	15
26	1KS19CS029	DINESH M	6	6	3				6	6			27	12	15
27	1KS19CS030	G PRERITHA	6	6	3				6	6			27	12	15
28	1KS19CS031	GAGAN REDDY S	6	6					6	5			23	11	12
29	1KS19CS032	GAGANDEEP K	6	6	3				6	5			26	11	15
30	1KS19CS033	GEETHANJALI B K PRASAD				6	6	6	6	6			30	12	18
31	1KS19CS034	GULSHAN KUMAR S	6	6	3				6	6			27	12	15
32	1KS19CS035	HARSHITHA J				6	6	6	6	6			30	12	18
33	1KS19CS036	INDRAJITH H M				6	6	6	6	6			30	12	18
34	1KS19CS038	K KISHAN				6	6	6	6	6			30	12	18
35	1KS19CS039	K R VAGEESH	6	6	0						6	6	24	12	12
36	1KS19CS040	KALYAN CHOWDHARY B	6	6	3				6	6			27	12	15
37	1KS19CS041	KARTHIK S MORAJKAR				6	6	6	6	6			30	12	18
38	1KS19CS042	KAVYASHREE.S.L				6	6	4			6	6	28	12	16
39	1KS19CS043	KEERTHAN GOWDA S				6	6	6	6	6			30	12	18
40	1KS19CS044	KOTHAPALLI SREEJA				6	6	6	6	6			30	12	18

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO5	CO5	CO5	CO5	CO5	CO5	CO4	CO4	CO4	CO4		CO4	CO5
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	12 M	18 M
41	1KS19CS045	KOTTALA SAIVENKATASUCHITH	6	6					5	5			22	10	12
42	1KS19CS046	KRISHNA K R	6	6					6	6			24	12	12
43	1KS19CS047	KUMAR S	6	6					6	6			24	12	12
44	1KS19CS048	LIKITH G	6	6					6	6			24	12	12
45	1KS19CS049	LISHA C	6	6	3				6	6			27	12	15
46	1KS19CS050	MAHAK SHREE	6	6	3				6	5			26	11	15
47	1KS19CS051	MALLIPALLI SPURTHI REDDY				6	6	6	6	6			30	12	18
48	1KS19CS052	MANASA G L				6	6	6	6	6			30	12	18
49	1KS19CS053	MOHAMMED NOOR AMAN				6	6	4			5	6	27	11	16
50	1KS19CS054	MUKESH KUMAR	6	6	3				6	6			27	12	15
51	1KS19CS055	MYTHREYI U				6	6	6	6	6			30	12	18
52	1KS19CS056	N ASHOK				6	6	6	6	6			30	12	18
53	1KS19CS057	N BHAVYA				6	6	6	6	6			30	12	18
54	1KS19CS058	N P SHASHANKA RAO	6	6	2						6	6	26	12	14
55	1KS18CS011	BHARATH R	6	6					6	5			23	11	12
56	1KS19CS116	VRATHIKA BILLAVA				6	6	0			6	6	24	12	12
57	1KS20CS400	AKIF DELVI				6	6	6	6	5			29	11	18
58	1KS20CS404	PRANAV CHANDRAN P				6	6	6	6	4			28	10	18
59	1KS20CS402	KEERTHI KUMAR V	6	6					6	6			24	12	12

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV/B
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	FINAL MARKS

Sl. NO.	USN	NAME	IA 1 (30M)	Assign 1 (10M)	IA 2 (30M)	Assign 2 (10M)	IA 3 (30M)	Assign 3 (10M)	Avg IA (30M)	Avg Assign (10M)	Final IA (30M)	Final Assign (10M)	Final (40M)	Sign
1	1KS19CS059	NAVEEN K M	29	10	25	10	28	10	27.33	10.00	28	10	38	NAVEEN
2	1KS19CS060	NEELAMMA SALI	29	10	28	10	27	10	28.00	10.00	28	10	38	
3	1KS19CS061	NETHRA R	26	10	27	10	29	3	27.33	7.67	28	8	36	Preethi.K.P.
4	1KS19CS062	NIKHIL CHOWDARY V R	19	10	26	10	21	10	22.00	10.00	22	10	32	
5	1KS19CS063	NITHYA S N	28	10	26	10	30	10	28.00	10.00	28	10	38	
6	1KS19CS064	PAVAN B INDRESH	26	10	26	10	20	10	24.00	10.00	24	10	34	Pavi
7	1KS19CS065	PAVAN L	25	10	27	10	26	10	26.00	10.00	26	10	36	~~Pavan~~ L
8	1KS19CS066	PAVITHRA S P	26	10	26	10	27	10	26.33	10.00	27	10	37	Preethi.K.P.
9	1KS19CS067	POOJA J	29	10	30	10	27	10	28.67	10.00	29	10	39	Pooja J
10	1KS19CS068	POTHURU SAI PRIYANKA	22	10	28	10	24	10	24.67	10.00	25	10	35	P.M
11	1KS19CS069	PRAJWAL G	20	10	22	10	19	10	20.33	10.00	21	10	31	Prajwal
12	1KS19CS070	PRAJWAL KULKARNI	24	10	26	10	30	10	26.67	10.00	27	10	37	
13	1KS19CS071	PREETHI K P	30	10	29	10	30	10	29.67	10.00	30	10	40	Preethi.K.P.
14	1KS19CS072	PRIYA E	27	10	28	10	30	10	28.33	10.00	29	10	39	Priya

15	1KS19CS073	PRIYANKA R	27	10	26	10	27	10	26.67	10.00	27	10	37	Priya
16	1KS19CS074	PUVAN KUMAR V	26	10	27	10	27	10	26.67	10.00	27	10	37	PUVAN
6	1KS19CS075	R V YASHVANTH	29	10	28	10	27	10	28.00	10.00	28	10	38	Yashu
18	1KS19CS076	RAJ KUMAR	18	10	25	10	27	0	23.33	6.67	24	7	31	Raj
19	1KS19CS077	RAKESH M J	25	10	28	10	29	10	27.33	10.00	28	10	38	MJ
20	1KS19CS078	RAKSHA T M	22	10	28	10	30	10	26.67	10.00	27	10	37	Raksha.T.M
21	1KS19CS079	RANJITHA D V	28	10	26	10	29	10	27.67	10.00	28	10	38	Ranjitha
22	1KS19CS080	SAHANA S	30	10	27	10	27	10	28.00	10.00	28	10	38	
23	1KS19CS081	SAI SHIRISHA S B	28	10	26	10	24	10	26.00	10.00	26	10	36	Sai
24	1KS19CS082	SANGHARSH KUMAR RAI	22	10	26	0	25	10	24.33	6.67	25	7	32	Sangarsh
25	1KS19CS083	SANJEEV MYSORE	26	10	27	10	27	10	26.67	10.00	27	10	37	
26	1KS19CS084	SANKALP KESTI	27	10	26	10	27	10	26.67	10.00	27	10	37	
27	1KS19CS085	SATISH V	30	10	28	10	30	10	29.33	10.00	30	10	40	SATHISH
28	1KS19CS086	SHANVI B P	26	10	27	10	29	10	27.33	10.00	28	10	38	
29	1KS19CS087	SHIVANI G K	30	10	27	10	29	10	28.67	10.00	29	10	39	Shivani
30	1KS19CS088	SHREESHA S	25	10	27	10	30	10	27.33	10.00	28	10	38	Shreesha
31	1KS19CS089	SHREYA R KIRAN	29	10	29	10	29	10	29.00	10.00	29	10	39	Shreya
32	1KS19CS090	SHRI HARSHA S	27	10	28	10	27	10	27.33	10.00	28	10	38	
33	1KS19CS091	SINDHU S	28	10	28	10	29	10	28.33	10.00	29	10	39	Sindhu
34	1KS19CS092	SMRITHI SHEKAR	28	10	28	10	29	10	28.33	10.00	29	10	39	
35	1KS19CS093	SRUJAN K	27	10	27	10	29	10	27.67	10.00	28	10	38	
36	1KS19CS094	SRUSTI S GOWDA	27	10	27	10	30	10	28.00	10.00	28	10	38	
37	1KS19CS096	SUHAS S	24	10	26	10	26	10	25.33	10.00	26	10	36	Suhas.S.
38	1KS19CS097	SUSHMITHA K	28	10	28	10	29	10	28.33	10.00	29	10	39	Sushmitha
39	1KS19CS098	SWETHA M KULKARNI	29	10	29	10	26	10	28.00	10.00	28	10	38	
40	1KS19CS099	SYED ZAINUL ABIDIN	26	10	26	8	22	10	24.67	9.33	25	10	35	S.Zain
41	1KS19CS100	TALLURU MAURYA	27	10	26	10	26	0	26.33	6.67	27	7	34	Maurya
42	1KS19CS101	TANUSHREE R	28	10	26	10	24	10	26.00	10.00	26	10	36	Tanu
43	1KS19CS102	TEJAS N	28	10	27	10	26	10	27.00	10.00	27	10	37	Tejas N
44	1KS19CS103	TEJAS P	24	6	22	7	29	10	25.00	7.67	25	8	33	TP
45	1KS19CS104	TEJASWINI NAYAKA S	28	10	28	10	28	10	28.00	10.00	28	10	38	Tejaswini
46	1KS19CS105	THANUSHREE S	27	10	26	10	24	10	25.67	10.00	26	10	36	Tanu

47	1KS19CS106	THIRUMAL R	17	6	27	10	30	10	24.67	8.67	25	9	34	Thirumal.R
48	1KS19CS107	THRIVENI U	29	10	28	10	29	10	28.67	10.00	29	10	39	Thriveni
49	1KS19CS108	UDHAY KUMAR G	21	10	22	10	18	10	20.33	10.00	21	10	31	~~UDA~~
50	1KS19CS109	VAISHNAVI G	28	10	26	10	27	10	27.00	10.00	27	10	37	Vaishnavi
51	1KS19CS110	VARSHA BAI R	26	10	28	10	30	10	28.00	10.00	28	10	38	~~VARSHA~~
52	1KS19CS111	VARUN KAMBALI	27	10	24	10	24	10	25.00	10.00	25	10	35	VK
53	1KS19CS112	VISHAL GUPTA	22	10	26	10	27	10	25.00	10.00	25	10	35	Vishal
54	1KS19CS113	YASHWANTH S R	29	10	28	10	30	10	29.00	10.00	29	10	39	YASHWANTH
55	1KS18CS021	G. SARAYU CHOWDARY	23	10	26	10	24	10	24.33	10.00	25	10	35	Sarayu
56	1KS18CS068	PRAJWAL N	17	10	26	10	30	10	24.33	10.00	25	10	35	Prajwal.N
57	1KS19CS115	AKSHAY R	30	10	28	10	30	10	29.33	10.00	30	10	40	Akshay
58	1KS20CS403	NITHIN S	26	10	26	10	29	10	27.00	10.00	27	10	37	Nithin S
59	1KS20CS401	ANUSHA A R	24	10	26	10	26	10	25.33	10.00	26	10	36	Anusha.AR

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV/B
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	I

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2		CO1	CO2
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	12 M
1	1KS19CS059	NAVEEN K M	6	6	5				6	6			29	17	12
2	1KS19CS060	NEELAMMA SALI	6	5	6						6	6	29	17	12
3	1KS19CS061	NETHRA R	5	6	4						5	6	26	15	11
4	1KS19CS062	NIKHIL CHOWDARY V R	6	6	4				3				19	16	3
5	1KS19CS063	NITHYA S N				5	6	6	4		5	6	28	17	11
6	1KS19CS064	PAVAN B INDRESH	6	6	4				5	5			26	16	10
7	1KS19CS065	PAVAN L	5	6	4				5	5			25	15	10
8	1KS19CS066	PAVITHRA S P	6	6	4						4	6	26	16	10
9	1KS19CS067	POOJA J	6	6	4						5		21	16	5
10	1KS19CS068	POTHURU SAI PRIYANKA	6	6	4				6				22	16	6
11	1KS19CS069	PRAJWAL G	4	6							4	6	20	10	10
12	1KS19CS070	PRAJWAL KULKARNI	6	6	6						0	6	24	18	6
13	1KS19CS071	PREETHI K P	6	6	6						6	6	30	18	12

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2		CO1	CO2
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	12 M
14	1KS19CS072	PRIYA E	6	6	4						6	5	27	16	11
15	1KS19CS073	PRIYANKA R	6	6	3						6	6	27	15	12
16	1KS19CS074	PUVAN KUMAR V	5	6	5						4	6	26	16	10
6	1KS19CS075	R V YASHVANTH	6	6	5						6	6	29	17	12
18	1KS19CS076	RAJ KUMAR	2	6	2						4	2	16	10	6
19	1KS19CS077	RAKESH M J				2	6	5	6	6			25	13	12
20	1KS19CS078	RAKSHA T M	5	6	5						0	6	22	16	6
21	1KS19CS079	RANJITHA D V	6	5	5						6	6	28	16	12
22	1KS19CS080	SAHANA S	6	6	6						6	6	30	18	12
23	1KS19CS081	SAI SHIRISHA S B	6	6	4						6	6	28	16	12
24	1KS19CS082	SANGHARSH KUMAR RAI	6	6	4				6				22	16	6
25	1KS19CS083	SANJEEV MYSORE	6	6	6				6	2			26	18	8
26	1KS19CS084	SANKALP KESTI	6	6	4						5	6	27	16	11
27	1KS19CS085	SATISH V	6	6	6				6	6			30	18	12
28	1KS19CS086	SHANVI B P	6	6	2						6	6	26	14	12
29	1KS19CS087	SHIVANI G K	6	6	6						6	6	30	18	12
30	1KS19CS088	SHREESHA S	5	5	5						6	4	25	15	10
31	1KS19CS089	SHREYA R KIRAN	6	5	6						6	6	29	17	12
32	1KS19CS090	SHRI HARSHA S				6	3	6			6	6	27	15	12
33	1KS19CS091	SINDHU S	6	6	4						6	6	28	16	12
34	1KS19CS092	SMRITHI SHEKAR	6	6	4						6	6	28	16	12
35	1KS19CS093	SRUJAN K	6	6	4						5	6	27	16	11
36	1KS19CS094	SRUSTI S GOWDA	6	6	4				6	5			27	16	11
37	1KS19CS096	SUHAS S	6	6	6				6				24	18	6

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2		CO1	CO2
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	12 M
38	1KS19CS097	SUSHMITHA K	6	5	5						6	6	28	16	12
39	1KS19CS098	SWETHA M KULKARNI	6	5	6						6	6	29	17	12
40	1KS19CS099	SYED ZAINUL ABIDIN	6	6	4						4	6	26	16	10
41	1KS19CS100	TALLURU MAURYA	6	6	4						5	6	27	16	11
42	1KS19CS101	TANUSHREE R	6	5	5						6	6	28	16	12
43	1KS19CS102	TEJAS N	6	6	4						6	6	28	16	12
44	1KS19CS103	TEJAS P	6	6	6						0	6	24	18	6
45	1KS19CS104	TEJASWINI NAYAKA S	6	6	4						6	6	28	16	12
46	1KS19CS105	THANUSHREE S	6	6	5						4	6	27	17	10
47	1KS19CS106	THIRUMAL R				5	3	6	3				17	14	3
48	1KS19CS107	THRIVENI U	6	5	6						6	6	29	17	12
49	1KS19CS108	UDHAY KUMAR G	5	5	5						5		20	15	5
50	1KS19CS109	VAISHNAVI G	5	6	6						5	6	28	17	11
51	1KS19CS110	VARSHA BAI R	6	6	6				6	2			26	18	8
52	1KS19CS111	VARUN KAMBALI	6	6	5						4	6	27	17	10
53	1KS19CS112	VISHAL GUPTA				6	2	4			4	6	22	12	10
54	1KS19CS113	YASHWANTH S R				6	5	6			6	6	29	17	12
55	1KS18CS021	G. SARAYU CHOWDARY	6	6	5						6		23	17	6
56	1KS18CS068	PRAJWAL N	2	6	3						6		17	11	6
57	1KS19CS115	AKSHAY R	6	6	6						6	6	30	18	12
58	1KS20CS403	NITHIN S	4	6	5						5	6	26	15	11
59	1KS20CS401	ANUSHA A R	5	6	2						5	6	24	13	11

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	II

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S		
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4		CO3	CO2	CO4
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	6 M	6 M
1	1KS19CS059	NAVEEN K M				6	6	5			3	5	**25**	17	3	5
2	1KS19CS060	NEELAMMA SALI	5	6	6						6	5	**28**	17	6	5
3	1KS19CS061	NETHRA R				5	6	5			6	5	**27**	16	6	5
4	1KS19CS062	NIKHIL CHOWDARY V R				6	5	4			6	5	**26**	15	6	5
5	1KS19CS063	NITHYA S N				5	6	4			6	5	**26**	15	6	5
6	1KS19CS064	PAVAN B INDRESH				6	5	4			6	5	**26**	15	6	5
7	1KS19CS065	PAVAN L				6	5	6			5	5	**27**	17	5	5
8	1KS19CS066	PAVITHRA S P				6	5	4			6	5	**26**	15	6	5
9	1KS19CS067	POOJA J				6	6	4			6	5	**30**	16	6	5
10	1KS19CS068	POTHURU SAI PRIYANKA				6	5	6			6	5	**28**	17	6	5
11	1KS19CS069	PRAJWAL G				6	5				6	5	**22**	11	6	5
12	1KS19CS070	PRAJWAL KULKARNI				5	5	5			6	5	**26**	15	6	5
13	1KS19CS071	PREETHI K P				6	6	6			6	5	**29**	18	6	5
14	1KS19CS072	PRIYA E				6	5	6			6	5	**28**	17	6	5

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S		
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4		CO3	CO2	CO4
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	6 M	6 M
15	1KS19CS073	PRIYANKA R				6	5	4			6	5	**26**	15	6	5
16	1KS19CS074	PUVAN KUMAR V				6	5	5			6	5	**27**	16	6	5
6	1KS19CS075	R V YASHVANTH	5	6	6						6	5	**28**	17	6	5
18	1KS19CS076	RAJ KUMAR				6	4	4			6	5	**25**	14	6	5
19	1KS19CS077	RAKESH M J				6	5	6			6	5	**28**	17	6	5
20	1KS19CS078	RAKSHA T M				6	5	6			6	5	**28**	17	6	5
21	1KS19CS079	RANJITHA D V				6	5	4			6	5	**26**	15	6	5
22	1KS19CS080	SAHANA S	6	6	4						6	5	**27**	16	6	5
23	1KS19CS081	SAI SHIRISHA S B				6	5	4			6	5	**26**	15	6	5
24	1KS19CS082	SANGHARSH KUMAR RAI				6	5	4			6	5	**26**	15	6	5
25	1KS19CS083	SANJEEV MYSORE				6	5	5			6	5	**27**	16	6	5
26	1KS19CS084	SANKALP KESTI				5	6	4			6	5	**26**	15	6	5
27	1KS19CS085	SATISH V				6	5	6			6	5	**28**	17	6	5
28	1KS19CS086	SHANVI B P				5	5	6			6	5	**27**	16	6	5
29	1KS19CS087	SHIVANI G K				5	5	6			6	5	**27**	16	6	5
30	1KS19CS088	SHREESHA S				5	6	5			6	5	**27**	16	6	5
31	1KS19CS089	SHREYA R KIRAN				6	6	6			6	5	**29**	18	6	5
32	1KS19CS090	SHRI HARSHA S				6	5	6			6	5	**28**	17	6	5
33	1KS19CS091	SINDHU S				6	5	6			6	5	**28**	17	6	5
34	1KS19CS092	SMRITHI SHEKAR				6	5	6			6	5	**28**	17	6	5
35	1KS19CS093	SRUJAN K				5	6	5			6	5	**27**	16	6	5
36	1KS19CS094	SRUSTI S GOWDA				6	5	4			6	6	**27**	15	6	6
37	1KS19CS096	SUHAS S				6	5	4			6	5	**26**	15	6	5
38	1KS19CS097	SUSHMITHA K				6	5	6			6	5	**28**	17	6	5

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S		
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4		CO3	CO2	CO4
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	18 M	6 M	6 M
39	1KS19CS098	SWETHA M KULKARNI	6	6	6						6	5	**29**	18	6	5
40	1KS19CS099	SYED ZAINUL ABIDIN				5	5	5			6	5	**26**	15	6	5
41	1KS19CS100	TALLURU MAURYA				5	5	5			6	5	**26**	15	6	5
42	1KS19CS101	TANUSHREE R				6	5	5			5	5	**26**	16	5	5
43	1KS19CS102	TEJAS N	5	6	5						6	5	**27**	16	6	5
44	1KS19CS103	TEJAS P	5	5	2						5	5	**22**	12	5	5
45	1KS19CS104	TEJASWINI NAYAKA S				6	5	6			6	5	**28**	17	6	5
46	1KS19CS105	THANUSHREE S				6	5	4			6	5	**26**	15	6	5
47	1KS19CS106	THIRUMAL R				6	5	5			6	5	**27**	16	6	5
48	1KS19CS107	THRIVENI U				6	5	6			6	5	**28**	17	6	5
49	1KS19CS108	UDHAY KUMAR G				6	5				6	5	**22**	11	6	5
50	1KS19CS109	VAISHNAVI G				6	5	4			6	5	**26**	15	6	5
51	1KS19CS110	VARSHA BAI R				6	6	5			6	5	**28**	17	6	5
52	1KS19CS111	VARUN KAMBALI				6	5	4			4	5	**24**	15	4	5
53	1KS19CS112	VISHAL GUPTA				6	5	4			6	5	**26**	15	6	5
54	1KS19CS113	YASHWANTH S R	6	6	5						6	5	**28**	17	6	5
55	1KS18CS021	G. SARAYU CHOWDARY				5	6	4			6	5	**26**	15	6	5
56	1KS18CS068	PRAJWAL N				6	5	4			6	5	**26**	15	6	5
57	1KS19CS115	AKSHAY R				6	5	6			6	5	**28**	17	6	5
58	1KS20CS403	NITHIN S				6	5	4			6	5	**26**	15	6	5
59	1KS20CS401	ANUSHA A R				6	5	4			6	5	**26**	15	6	5

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II /IV/B
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021
INTERNALS	III

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO5	CO5	CO5	CO5	CO5	CO5	CO4	CO4	CO4	CO4		CO4	CO5
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	12 M	18 M
1	1KS19CS059	NAVEEN K M				6	6	4	6	6			28	12	16
2	1KS19CS060	NEELAMMA SALI	6	6	3				6	6			27	12	15
3	1KS19CS061	NETHRA R				6	6	6	6	5			29	11	18
4	1KS19CS062	NIKHIL CHOWDARY V R				6	6	3			5	1	21	6	15
5	1KS19CS063	NITHYA S N				6	6	6	6	6			30	12	18
6	1KS19CS064	PAVAN B INDRESH				6	6	3			5		20	5	15
7	1KS19CS065	PAVAN L				6	6	2	6	6			26	12	14
8	1KS19CS066	PAVITHRA S P	6	6	3				6	6			27	12	15
9	1KS19CS067	POOJA J	6	6	3				6	6			27	12	15
10	1KS19CS068	POTHURU SAI PRIYANKA				6	6	0	6	6			24	12	12
11	1KS19CS069	PRAJWAL G	6	6					6	1			19	7	12
12	1KS19CS070	PRAJWAL KULKARNI				6	6	6	6	6			30	12	18
13	1KS19CS071	PREETHI K P				6	6	6	6	6			30	12	18
14	1KS19CS072	PRIYA E				6	6	6	6	6			30	12	18

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO5	CO5	CO5	CO5	CO5	CO5	CO4	CO4	CO4	CO4		CO4	CO5
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	12 M	18 M
15	1KS19CS073	PRIYANKA R	6	6	3				6	6			**27**	12	15
16	1KS19CS074	PUVAN KUMAR V	6	6	3				6	6			**27**	12	15
6	1KS19CS075	R V YASHVANTH	6	6	3				6	6			**27**	12	15
18	1KS19CS076	RAJ KUMAR	6	6	3				6	6			**27**	12	15
19	1KS19CS077	RAKESH M J				6	6	5	6	6			**29**	12	17
20	1KS19CS078	RAKSHA T M				6	6	6	6	6			**30**	12	18
21	1KS19CS079	RANJITHA D V				6	6	6	6	5			**29**	11	18
22	1KS19CS080	SAHANA S	6	6	3				6	6			**27**	12	15
23	1KS19CS081	SAI SHIRISHA S B	6	6					6	6			**24**	12	12
24	1KS19CS082	SANGHARSH KUMAR RAI	6	6	1				6	6			**25**	12	13
25	1KS19CS083	SANJEEV MYSORE	6	6	3				6	6			**27**	12	15
26	1KS19CS084	SANKALP KESTI	6	6	3				6	6			**27**	12	15
27	1KS19CS085	SATISH V				6	6	6	6	6			**30**	12	18
28	1KS19CS086	SHANVI B P				6	6	6	6	5			**29**	11	18
29	1KS19CS087	SHIVANI G K				6	6	6	6	5			**29**	11	18
30	1KS19CS088	SHREESHA S				6	6	6	6	6			**30**	12	18
31	1KS19CS089	SHREYA R KIRAN				6	6	6	6	5			**29**	11	18
32	1KS19CS090	SHRI HARSHA S	6	6	3				6	6			**27**	12	15
33	1KS19CS091	SINDHU S				6	6	6	6	5			**29**	11	18
34	1KS19CS092	SMRITHI SHEKAR				6	6	6	6	5			**29**	11	18
35	1KS19CS093	SRUJAN K				6	6	6	6	5			**29**	11	18
36	1KS19CS094	SRUSTI S GOWDA				6	6	6	6	6			**30**	12	18
37	1KS19CS096	SUHAS S				6	6	2			6	6	**26**	12	14
38	1KS19CS097	SUSHMITHA K				6	6	6	6	5			**29**	11	18

SI. NO.	USN	NAME	Q.N.1A	Q.N.1B	Q.N.1C	Q.N.2A	Q.N.2B	Q.N.2C	Q.N.3A	Q.N.3B	Q.N.4A	Q.N.4B	TOTAL	CO'S	
			CO5	CO5	CO5	CO5	CO5	CO5	CO4	CO4	CO4	CO4		CO4	CO5
			6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	6 M	30 M	12 M	18 M
39	1KS19CS098	SWETHA M KULKARNI	6	6	3				6	5			26	11	15
40	1KS19CS099	SYED ZAINUL ABIDIN	6	6					6	4			22	10	12
41	1KS19CS100	TALLURU MAURYA	6	6	3				6	5			26	11	15
42	1KS19CS101	TANUSHREE R				6	6		6	6			24	12	12
43	1KS19CS102	TEJAS N	6	6	3				6	5			26	11	15
44	1KS19CS103	TEJAS P				6	6	6	6	5			29	11	18
45	1KS19CS104	TEJASWINI NAYAKA S				6	6	5	6	5			28	11	17
46	1KS19CS105	THANUSHREE S				6	6		6	6			24	12	12
47	1KS19CS106	THIRUMAL R				6	6	6	6	6			30	12	18
48	1KS19CS107	THRIVENI U				6	6	6	6	5			29	11	18
49	1KS19CS108	UDHAY KUMAR G	6						6	6			18	12	6
50	1KS19CS109	VAISHNAVI G	6	6	3				6	6			27	12	15
51	1KS19CS110	VARSHA BAI R				6	6	6	6	6			30	12	18
52	1KS19CS111	VARUN KAMBALI	6	6	0				6	6			24	12	12
53	1KS19CS112	VISHAL GUPTA	6	6	3				6	6			27	12	15
54	1KS19CS113	YASHWANTH S R				6	6	6	6	6			30	12	18
55	1KS18CS021	G. SARAYU CHOWDARY				6	6				6	6	24	12	12
56	1KS18CS068	PRAJWAL N				6	6	6	6	6			30	12	18
57	1KS19CS115	AKSHAY R				6	6	6	6	6			30	12	18
58	1KS20CS403	NITHIN S				6	6	5	6	6			29	12	17
59	1KS20CS401	ANUSHA A R				6	6	2	6	6			26	12	14

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

Visvesvaraya Technological University

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

Branch : CS Scheme : 2018 Semester : 4

Sl NO.	USN	18CPC49	18CS42	18CS43	18CS44	18CS45	18CS46	18CSL47	18CSL48	18MAT41	18MATDIP41	STUDENT SIGNATURE
1	1KS18CS011	36	30	31	38	37	30	37	39	38	-	
2	1KS18CS021	36	35	35	40	37	37	37	40	40	-	
3	1KS18CS068	36	37	35	40	40	33	38	40	39	-	
4	1KS19CS001	37	39	36	40	37	35	40	40	40	-	
5	1KS19CS002	35	39	37	39	40	33	39	37	40	-	
6	1KS19CS003	33	32	34	40	39	32	37	38	39	-	
7	1KS19CS004	35	38	37	40	40	36	38	39	40	-	
8	1KS19CS005	35	38	37	38	40	34	40	39	39	-	
9	1KS19CS006	34	39	37	40	40	33	38	39	40	-	
10	1KS19CS007	34	37	37	40	39	34	37	39	37	-	
11	1KS19CS009	35	39	35	40	40	34	39	40	40	-	
12	1KS19CS010	35	40	37	40	40	35	38	38	40	-	
13	1KS19CS011	36	40	38	40	40	39	39	40	40	-	
14	1KS19CS012	36	40	36	40	40	38	40	40	40	-	
15	1KS19CS014	36	40	38	40	40	37	40	38	40	-	
16	1KS19CS015	37	40	38	40	40	38	40	40	40	-	
17	1KS19CS016	37	40	39	40	40	38	40	40	39	-	
18	1KS19CS017	37	40	40	40	40	40	40	40	40	-	
19	1KS19CS018	36	40	38	40	40	38	40	40	40	-	
20	1KS19CS019	36	40	39	40	40	37	40	40	40	-	
21	1KS19CS020	36	40	38	39	39	35	39	40	40	-	
22	1KS19CS021	36	40	40	40	40	37	40	40	40	-	
23	1KS19CS022	35	30	33	34	38	33	38	38	37	-	
24	1KS19CS023	36	39	35	38	40	35	37	40	40	-	
25	1KS19CS024	36	40	37	40	40	37	40	40	40	-	
26	1KS19CS025	36	40	37	40	40	36	40	40	40	-	
27	1KS19CS026	35	40	38	39	39	35	38	40	39	-	
28	1KS19CS028	36	39	38	40	40	33	38	39	40	-	
29	1KS19CS029	35	38	36	40	39	31	38	37	37	-	
30	1KS19CS030	35	40	37	40	39	38	37	38	40	-	
31	1KS19CS031	35	38	32	37	40	31	37	38	39	-	
32	1KS19CS032	36	38	36	40	40	34	37	38	39	-	
33	1KS19CS033	36	37	38	40	38	36	40	39	40	-	
34	1KS19CS034	36	38	37	40	40	30	40	39	39	-	
35	1KS19CS035	35	39	39	40	40	38	40	40	40	-	
36	1KS19CS036	37	40	39	40	40	37	40	40	40	-	
37	1KS19CS038	35	40	39	40	40	36	40	40	40	-	
38	1KS19CS039	38	40	37	40	40	36	40	40	40	-	
39	1KS19CS040	33	38	35	38	38	32	40	38	40	-	
40	1KS19CS041	38	39	39	40	39	36	40	39	40	-	
41	1KS19CS042	36	40	39	40	40	39	40	40	40	-	
42	1KS19CS043	35	40	38	40	40	38	40	40	40	-	
43	1KS19CS044	36	40	37	40	40	37	40	40	39	-	
44	1KS19CS045	35	37	32	39	40	33	38	37	38	-	
45	1KS19CS046	33	38	36	38	40	32	37	37	37	-	
46	1KS19CS047	32	30	31	30	37	30	37	38	35	-	

Sl NO.	USN	18CPC49	18CS42	18CS43	18CS44	18CS45	18CS46	18CSL47	18CSL48	18MAT41	18MATDIP41	STUDENT SIGNATURE
47	1KS19CS048	35	32	31	34	40	30	37	37	33	-	
48	1KS19CS049	36	38	38	40	40	39	39	40	40	-	
49	1KS19CS050	35	40	38	40	40	38	40	40	40	-	
50	1KS19CS051	35	40	39	40	40	39	40	40	40	-	
51	1KS19CS052	36	40	39	40	40	37	40	40	40	-	
52	1KS19CS053	35	40	36	38	39	31	37	37	40	-	
53	1KS19CS054	33	39	35	39	40	33	37	37	39	-	
54	1KS19CS055	36	40	39	40	40	38	40	40	40	-	
55	1KS19CS056	35	40	38	40	40	38	39	40	40	-	
56	1KS19CS057	36	40	38	40	40	38	40	40	38	-	
57	1KS19CS058	37	40	37	40	39	39	40	40	40	-	
58	1KS19CS059	36	38	38	39	39	34	40	40	39	-	
59	1KS19CS060	35	37	38	39	40	38	39	39	40	-	
60	1KS19CS061	36	38	36	40	39	34	40	39	40	-	
61	1KS19CS062	36	37	32	39	40	31	39	40	38	-	
62	1KS19CS063	36	38	38	40	40	37	40	40	40	-	
63	1KS19CS064	35	37	34	39	40	33	38	39	40	-	
64	1KS19CS065	35	37	36	40	40	39	38	39	39	-	
65	1KS19CS066	35	37	37	39	40	34	40	40	40	-	
66	1KS19CS067	35	40	39	40	40	39	40	40	40	-	
67	1KS19CS068	36	39	35	39	40	35	40	39	40	-	
68	1KS19CS069	36	34	31	38	38	30	37	40	39	-	
69	1KS19CS070	36	40	37	40	39	38	40	40	40	-	
70	1KS19CS071	37	40	40	40	40	40	40	40	40	-	
71	1KS19CS072	36	37	39	40	40	38	40	40	40	-	
72	1KS19CS073	36	37	37	39	40	38	39	40	40	-	
73	1KS19CS074	35	37	37	40	40	37	40	39	40	-	
74	1KS19CS075	35	38	38	39	40	37	40	40	40	-	
75	1KS19CS076	34	34	31	37	39	32	37	39	39	-	
76	1KS19CS077	35	37	38	40	40	37	38	39	40	-	
77	1KS19CS078	36	40	37	40	40	37	39	40	40	-	
78	1KS19CS079	35	37	38	40	40	36	39	39	40	-	
79	1KS19CS080	36	37	38	40	40	38	37	39	40	-	
80	1KS19CS081	35	37	36	38	40	33	40	40	40	-	
81	1KS19CS082	35	33	32	39	38	32	37	40	37	-	
82	1KS19CS083	35	37	37	40	40	37	40	40	40	-	
83	1KS19CS084	35	37	37	39	39	36	39	40	40	-	
84	1KS19CS085	36	40	40	40	40	38	40	40	40	-	
85	1KS19CS086	35	37	38	40	40	36	40	40	40	-	
86	1KS19CS087	35	39	39	39	40	38	40	39	40	-	
87	1KS19CS088	35	37	38	40	40	35	40	40	40	-	
88	1KS19CS089	36	37	39	39	40	37	39	40	40	-	
89	1KS19CS090	35	37	38	39	40	35	38	40	40	-	
90	1KS19CS091	36	37	39	39	40	35	38	40	40	-	
91	1KS19CS092	35	34	39	40	40	36	38	40	40	-	
92	1KS19CS093	35	39	38	39	40	35	37	38	40	-	
93	1KS19CS094	36	39	38	39	40	38	39	40	40	-	
94	1KS19CS096	36	35	36	39	40	32	38	40	40	-	
95	1KS19CS097	36	37	39	39	40	35	40	39	40	-	
96	1KS19CS098	35	39	38	40	40	37	40	40	40	-	
97	1KS19CS099	35	36	35	38	37	30	38	39	40	-	
98	1KS19CS100	37	37	34	39	40	35	37	39	40	-	
99	1KS19CS101	36	37	36	40	40	37	38	40	40	-	
100	1KS19CS102	34	37	37	39	40	38	38	39	40	-	
101	1KS19CS103	34	37	33	39	40	32	38	40	38	-	

Sl NO.	USN	18CPC49	18CS42	18CS43	18CS44	18CS45	18CS46	18CSL47	18CSL48	18MAT41	18MATDIP41	STUDENT SIGNATURE
102	1KS19CS104	36	39	38	40	40	38	40	40	40	-	
103	1KS19CS105	35	35	36	38	40	33	38	40	40	-	
104	1KS19CS106	34	37	34	40	40	32	37	39	40	-	
105	1KS19CS107	36	37	39	38	40	38	40	40	40	-	
106	1KS19CS108	33	33	31	34	37	30	37	38	34	-	
107	1KS19CS109	34	37	37	40	40	32	40	40	40	-	
108	1KS19CS110	37	39	38	40	40	38	40	40	40	-	
109	1KS19CS111	38	37	35	40	40	33	37	40	40	-	
110	1KS19CS112	35	37	35	39	40	36	39	40	40	-	
111	1KS19CS113	36	40	39	40	40	40	40	40	40	-	
112	1KS19CS115	38	40	40	39	40	38	40	40	40	-	
113	1KS19CS116	35	30	31	36	40	35	40	39	39	-	
114	1KS20CS400	36	40	37	39	40	35	39	39	39	36	
115	1KS20CS401	36	37	36	39	40	35	38	39	40	38	
116	1KS20CS402	35	39	32	39	39	30	39	39	37	37	
117	1KS20CS403	36	37	37	40	40	31	39	40	40	36	
118	1KS20CS404	36	36	32	33	35	30	38	38	35	33	
--x--	Faculty Signature											----xxxxxxx----

* - values are either optional subjects or the faculty has not yet entered the marks

Mangalapuri 2/9/2021

HOD

Seal and Signature

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

PRINCIPAL

Seal and Signature

PRINCIPAL
K.S. INSTITUTE OF TECHNOLOGY
BENGALURU - 560 109

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
DEARTMENT OF COMPUTER SCIENCS & ENGINEERING
TEACHING AND LEARNING
PEDAGOGY REPORT

Academic Year	**2020-2021**
Name of the Faculty	**Mrs. Vaneeta M**
Course Name /Code	**Operating Systems / 18CS56**
Semester/Section	**4th A**
Activity Name	**Demonstration of Programs (team Activity)**
Topic Covered	**Module 1(Process Scheduling Algorithms)**
Date	**03-06-2021**
No. of Participants	**59 /59**
Objectives/Goals	• **To improve the knowledge of scheduling techniques and programming skills through Experiential learning.** • **To improve team work through collaborative learning**
ICT Used	**Used Codeblocks for implementation and Online tool Microsoft team for presentation**

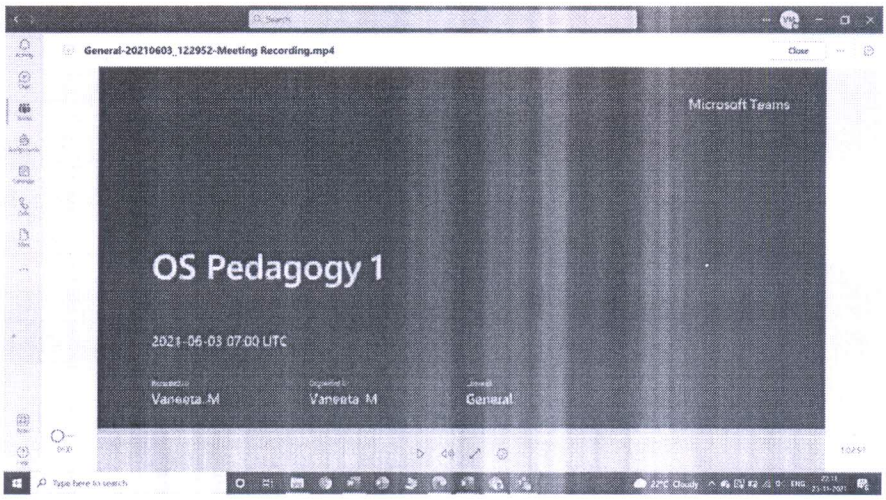
Appropriate Method/Instructional materials/Exam Questions

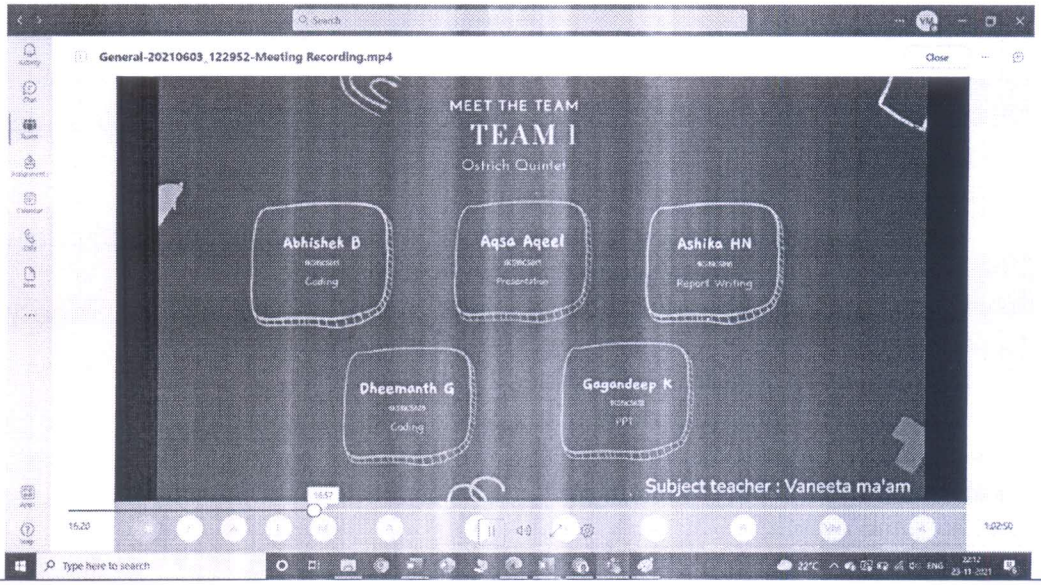
- It was a Team Activity.
- In the class of 59 students 12 teams are formed as Team A1, A2 and A3..... Each team consists of minimum 2 and maximum 4 students
- The team leaders are identified by team mates.
- The topics of demonstration on Implementation of CPU Scheduling algorithm using any programming language assigned to teams were as follows:
- First Come First Serve, Shortest Jib First, Shortest remaining Time First, Priority and Round Robin
- Team of 4 students or more were created.
- Each team implemented algorithm using codeblocks. Teams did power point presentation. They explained the algorithm, traced and executed the program.
- Teams submitted report containing detail description about scheduling method, code, snapshots and output.

Relevant PO's:	**PO: 1,2,3, 5, 9 and 12**
Significance of Results/Outcomes	Students were able to know the importance of experiential learning and understand the process scheduling algorithms.
Reflective Critique	The main goal of this team demo is to know how well students will be able to work as a team, do the presentation, explore the execution of programs and apply the knowledge effectively.

Proofs (Photographs/Videos/Reports/Charts/Models)

PFA one sample mini project report on Implementation of Transposition Cipher Algorithm





Signature of Course In charge

Signature of HOD CSE

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

Pedagogy Assessment

Team 1	Algorithm	Assignment 1, Q.7 (4 Marks)
Gagandeep K	Shortest Remaining Time First	4
Abhishek B		4
Ashika		4
Aqsa		4
Dheemanth		4
Team 2		
Devi Prasad	Priority Scheduling	4
Akash		4
Ashok		4
Dinesh		4
Suchith		4
Team 3	Round Robin	
Bhoomika AM		4
Bhoomika K		4
Vagheesh		4
Kishan		4
Team 4		
Gulshan	Round Robin	4
Shashank Rao		4
Indrajith H M		4
Ajay S Kalburgi		4
Keerthan		4
Team 5		
Deeksha	Shortest Job First	4
Anagha		4
Akif		4
Lisha		4
Team 6		
Amogh	First Come First Serve	4
Adith Karthik		4
Preritha		4
Gagan reddy		4
Amrutha		4
Team 7		
Manasa	Priority Scheduling	4
Bhavya		4
Sreeja		4
Spurthi		4
Anusha		4
Team 8		
Chaithanya	First Come First Serve	4
Deepthi		4
Deepa G		4
Bhumika M		4
Team 9		
Aakriti	Shortest Job First	4
Mahak shree		4
Mythreyi U		4

Aman Kushwaha		4
Mukesh Kumar		4
Team 10		
Bharath R	First Come First Serve	4
Kalyan Chowdhary		4
Chandan B V		4
Krishna		4
Kumar		4
Team 11		
Shreyas	Priority Scheduling	4
Amith K B		4
Likith		4
Mommed Noor Aman		4
Abhishek Yadav		4
Team 12		
Geethanjali B K	Shortest Remaining Time First	4
Kavyashree S L		4
Harshitha J		4
Karthik S M		4

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
DEARTMENT OF COMPUTER SCIENCS & ENGINEERING
TEACHING AND LEARNING
PEDAGOGY REPORT

Academic Year	**2020-2021**
Name of the Faculty	**Mrs. Vaneeta M**
Course Name /Code	**Operating Systems / 18CS56**
Semester/Section	**4th B**
Activity Name	**Demonstration of Programs (team Activity)**
Topic Covered	**Module 1 (Process Scheduling Algorithms)**
Date	**03-06-2021**
No. of Participants	**57 /59**
Objectives/Goals	**Experiential learning**
ICT Used	**Codeblocks for demo and Online tool Microsoft team for presentation**

Appropriate Method/Instructional materials/Exam Questions

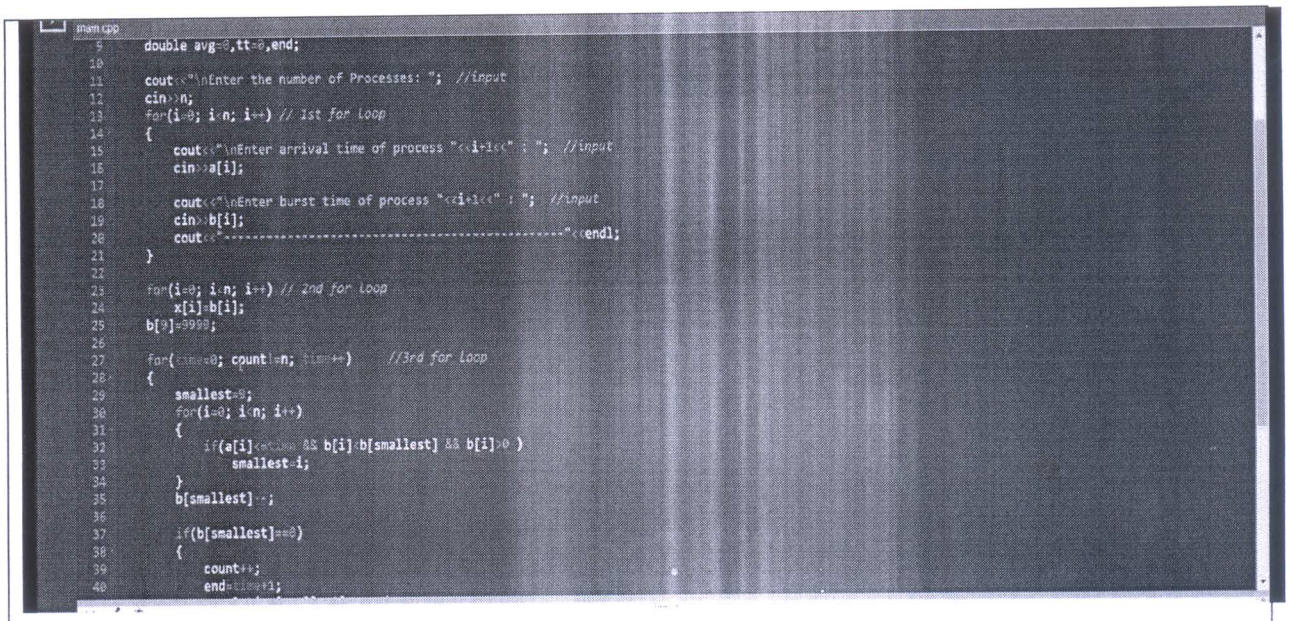
- It was a Team Activity.
- In the class of 59 students 12 teams are formed as Team A1, A2 and A3..... Each team consists of minimum 2 and maximum 4 students
- The team leaders are identified by team mates.
- The topics of demonstration on Implementation of CPU Scheduling algorithm using any programming language assigned to teams were as follows:
- First Come First Serve, Shortest Jib First, Shortest remaining Time First, Priority and Round Robin
- Team of 4 students or more were created.
- Each team implemented algorithm using codeblocks. Teams did power point presentation. They explained the algorithm, traced and executed the program.

Teams submitted Report containing detail description about scheduling method, code,

snapshots and output.

Relevant PO's:	**PO: 1,2,5 9 and 12**
Significance of Results/Outcomes	Students were able to know the importance of experiential learning and understand the process scheduling algorithms.
Reflective Critique	The main goal of this Group demo is to know how well students will be able to work as a team, explore the execution of programs and apply the knowledge effectively.

Proofs (Photographs/Videos/Reports/Charts/Models)



```
main.cpp
    double avg=0,tt=0,end;

    cout<<"\nEnter the number of Processes: ";  //input
    cin>>n;
    for(i=0; i<n; i++) // 1st for loop
    {
        cout<<"\nEnter arrival time of process "<<i+1<<" : ";  //input
        cin>>a[i];

        cout<<"\nEnter burst time of process "<<i+1<<" : ";  //input
        cin>>b[i];
        cout<<"-----------------------------------------------"<<endl;
    }

    for(i=0; i<n; i++) // 2nd for loop
        x[i]=b[i];
    b[9]=9999;

    for(time=0; count!=n; time++)     //3rd for loop
    {
        smallest=9;
        for(i=0; i<n; i++)
        {
            if(a[i]<=time && b[i]<b[smallest] && b[i]>0 )
                smallest=i;
        }
        b[smallest]--;

        if(b[smallest]==0)
        {
            count++;
            end=time+1;
```

Mary

Signature of Course In charge

Signature of HOD CSE

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

Pedagogy Assessment in Assignment 1 Question 7 (4 Marks)

Team:1: Shortest Remaining Time First

1. Maurya. = 4M
2. Sanjeev. =4M
3. Syed. =4M
4. Srujan. 4M
5. Varun.4M

Team:2: Shortest Job First

1. Sai shirisha 4M
2. Tanushree R 4M
3. Shivani 4M
4. Shreya 4M

Team:3: Round Robin

1. Varsha Bai R4M
2. Sindhu S 4M
3. Sushmita K 4M
4. Tejaswini Nayaka 4M
5. Smrithi Shekar 4M

Team:4: First Come First Serve

1. G Sarayu Chowdary 4M
2. Anusha A R 4M

Team:5: Priority Scheduling

1. Pooja J 4M
2. Sai Priyanka 4M
3. Priyanka R 4M
4. Raksha TM 4M
5. Srusti S Gowda 4M

Team:6: Shortest Job First

1. Nithya s n 4 M
2. Pavithra s p 4M
3. Priya E 4M
4. Shanvi 4M
5. Nethra 4M

Team:7: Priority Scheduling

1. Neelamma Sali 4M

2. Swetha Kulkarni 4M
3. Sahana S 4M
4. Ranjitha D V 4M
5. Thriveni U 4M

Team:8: Shortest Job First

1. Naveen K M 4 M
2. R V Yashvanth 4M
3. Rakesh 4M
4. Puvan Kumar 4M
5. Yashvanth S R 4M

Team:9:Shortest Job First

1. Nithin 4M
2. Suhas.S 4M
3. Tejas.N 4M
4. Prajwal.G 4M
5. Prajwal.N 4M

Team:10: Round Robin

1. Satish V 4M
2. Pavan Lakkana 4M
3. Pavan Indresh 4M
4. Nikhil Chowdhary 4M
5. Udhay 4M

Team:11: Shortest Remaining Time First

1. Preethi Kp 4M
2. Prajwal kulkarani 4M
3. Akshay R 4M
4. Shreesha S 4M

Team:12: First Come First Serve

1. Vaishnavi.G 4M
2. Tanushree.S 4M

Team:13: Shortest Job First

Sangharsh Kumar Rai=4M

Vishal Gupta=4M

Raj Mishra=4M

Sriharsha=4M

Sankalp=4M

Team:14: First Come First Serve

Tejas P =0M

Thirumal=0M

KS INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

SUBJECT : OPERATING SYSTEM

FACULTY : VANEETHA M

SHORTEST REMAINING TIME FIRST

BY

HARSHITHA J(1KS19CS035)

KAVYASHREE S L(1KS19CS042)

GEETHANJALI B K PRASAD(1KS19CS033)

KARTHIK S MORAJKAR(1KS19CS041)

SRTF(THEORY)

SRTF, which stands for **shortest remaining time first** is a scheduling algorithm used in operating systems, which can also be called as the preemptive version of the SJF scheduling algorithm. The process which has the least processing time remaining is executed first.

This algorithm is the **preemptive version** of **sjf scheduling**. In SRTF, the execution of the process can be stopped after certain amount of time. At the arrival of every process, the short term scheduler schedules the process with the least remaining burst time among the list of available processes and the running process.

Once all the processes are available in the **ready queue**, no preemption will be done and the algorithm will work as **sjf scheduling**. The context of the process is saved in the **process control block** when the process is removed from the execution and the next process is scheduled. This PCB is accessed on the **next execution** of this process

Advantages

- Short processes are handled very quickly.
- The system also requires very little overhead since it only makes a decision when a process completes or new process is added.
- When a new process is added the algorithm only needs to compare the currently executing process with the new process ,ignoring all other processes currently waiting to execute.

Disadvantages

- Like shortest job first ,it has the potential for process starvation
- Long processes may be held off indefinitely if short processes are continually added.

Suppose we have the following 3 processes with process ID's P1, P2, and P3 and they arrive into the CPU in the following manner:

Process ID	Arrival Time (milliseconds)	Burst Time (milliseconds)
P1	0	8
P2	1	2
P3	4	3

GANTT CHART

0 P1	1 P2	2 P2	3 P1	4 P3	5 P3	6 P3	7 P1	8 P1	9 P1	10 P1	11 P1	12 P1 13

EXPLAINATION

- At the 0th unit of the CPU, we have only process **P1**, so it gets executed for the 1-time unit.
- At the 1st unit of the CPU, the Process **P2** also arrives. Now, the **P1** needs 7 more units more to be executed, and **P2** needs only 2 units. So, **P2** is executed by preempting **P1**.
- **P2** gets completed at time unit 3, and unit now no new process has arrived. So, after the completion of **P2**, again **P1** is sent for execution.
- Now, **P1** has been executed for one unit only, and we have an arrival of new process **P3** at time unit 4. Now, the **P1** needs 6-time units more and **P3** needs only 3-time units. So, **P3** is executed by preempting **P1**.
- **P1** gets completed at time unit 7, and after that, we have the arrival of no other process. So again, **P1** is sent for execution, and it gets completed at 13th unit.

P ID	Arrival Time	Burst Time	Completion time (milliseconds)	Turn Around Time (milliseconds)	Waiting Time (milliseconds)
P1	0	8	13	13	5
P2	1	2	3	2	0
P3	4	3	7	3	0

Total Turn Around Time = 13 + 2 + 3= 18 milliseconds

Average Turn Around Time = Total Turn Around Time /Total No. of Processes
= 18 / 3 =6 milliseconds

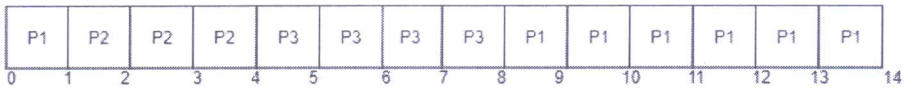
Total Waiting Time = 5 + 0 + 0 = 5 milliseconds

Average Waiting Time = Total Waiting Time / Total No. of Processes
= 5 / 3= 1.67 milliseconds

EXAMPLE-2

Process	Burst Time	Arrival Time
P1	7	0
P2	3	1
P3	4	3

The Gantt Chart for SRTF will be:



Explanation

- At the 0th unit of the CPU, there is only one process that is **P1**, so P1 gets executed for the 1 time unit.
- At the 1st unit of the CPU, Process **P2** arrives. Now, the **P1** needs 6 more units more to be executed, and the **P2** needs only 3 units. So, **P2** is executed first by preempting **P1**.
- At the 3rd unit of time, the process **P3** arrives, and the burst time of P3 is 4 units which is more than the completion time of P2 that is 1 unit, so P2 continues its execution.
- Now after the completion of **P2**, the burst time of **P3** is **4 units** that mean it needs only 4 units for completion while P1 needs 6 units for completion.
- So, this algorithm picks **P3** above **P1 due to the reason that the completion time of P3 is less than that of P1**
- P3 gets completed at time unit 8, there are no new process arrived.
- So again, **P1** is sent for execution, and it gets completed at the 14th unit

Process	Arrival time	Burst time	Completion time	Turnaround time	Waiting time
P1	0	7	14	14-0=14	14-7=7
P2	1	3	5	5-1=4	4-3=1
P3	3	4	8	8-3=5	5-4=1

Average waiting time =7+1+1=9/3=3ms

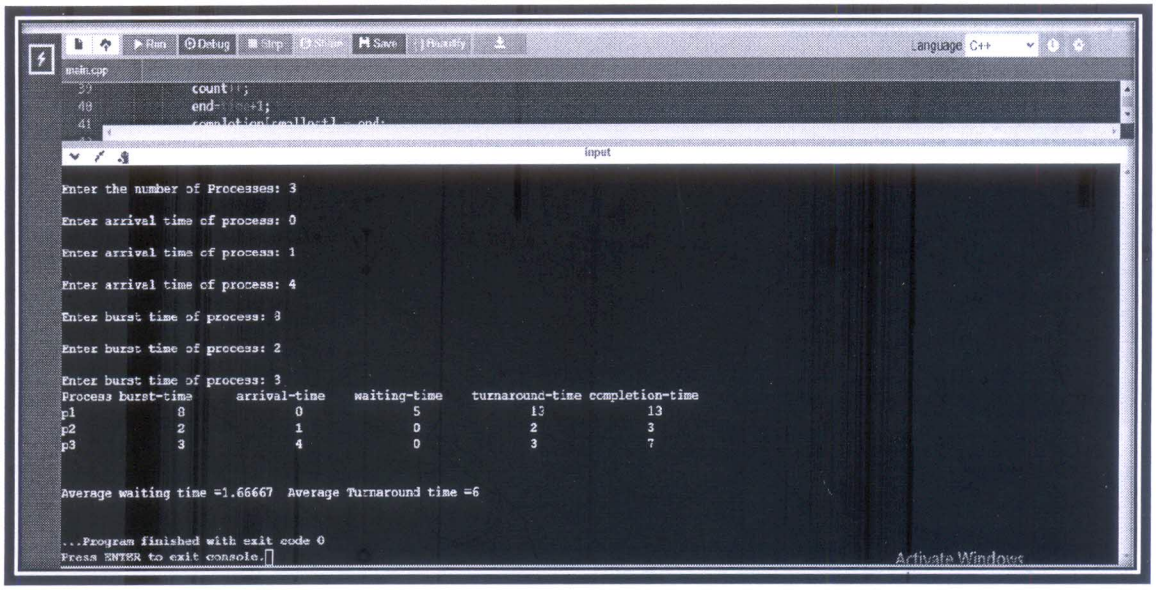
Average turnaround time =14+4+5=23/3=7.67ms

Implementation of SRTF in c++

```
#include<iostream>

using namespace std;
int main()
{
  int a[10],b[10],x[10];
  int waiting[10],turnaround[10],completion[10];
  int i,j,smallest,count=0,time,n;
  double avg=0,tt=0,end;

  cout<<"\nEnter the number of Processes: ";  //input
  cin>>n;
  for(i=0; i<n; i++)
  {
    cout<<"\nEnter arrival time of process: ";  //input
    cin>>a[i];
  }

for(i=0; i<n; i++)
  {
    cout<<"\nEnter burst time of process: ";  //input
    cin>>b[i];
  }
  for(i=0; i<n; i++)
    x[i]=b[i];

  b[9]=9999;
  for(time=0; count!=n; time++)
  {
    smallest=9;
    for(i=0; i<n; i++)
    {
      if(a[i]<=time && b[i]<b[smallest] && b[i]>0 )
        smallest=i;
    }
    b[smallest]--;

if(b[smallest]==0)
    {
      count++;
      end=time+1;
      completion[smallest] = end;
      waiting[smallest] = end - a[smallest] - x[smallest];
      turnaround[smallest] = end - a[smallest];
    }
  }
  cout<<"Process"<<"\t"<< "burst-time"<<"\t"<<"arrival-time" <<"\t"<<"waiting-time"
```

```
<<"\t"<<"turnaround-time"<<    "\t"<<"completion-time"<<endl;
  for(i=0; i<n; i++)
  {
    cout<<"p"<<i+1<<"\t\t"<<x[i]<<"\t\t"<<a[i]<<"\t\t"<<waiting[i]<<"\t\t"<<turnaround[i]<<"\t\t"<<comp
letion[i]<<endl;
    avg = avg + waiting[i];
    tt = tt + turnaround[i];
  }
  cout<<"\n\nAverage waiting time ="<<avg/n;
  cout<<"  Average Turnaround time ="<<tt/n<<endl;
}
```

output



K S INSTITUTE OF TECHNOLOGY BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

15CS64-OPERATING SYSTEMS
EXHAUSTED QUESTION BANK

Name of Faculty: Vaneeta M

MODULE-I

1. What is an Operating System? Explain considering different possible views
2. **List the essential properties of the following operating system (4 M)**
   a. **Batch processing**
   b. **Real time operating system**
   c. **Network operating system**
   d. **Distributed operating systems**
3. Describe Single processor, multiprocessing and clustered systems. What are the advantages and dis-advantages of multiprocessor systems? (8M)
4. Describe differences between symmetric and asymmetric multiprocessor system. (4M)
5. Explain the 'graceful degradation' and 'fault tolerant' in a multiprocessor system.(2M)
6. Distinguish among the following terminologies associated with the operating system and explain each of them in detail.(4M)
   a. Multiprogramming systems.
   b. Multitasking systems.
   c. Time Sharing
7. Explain dual mode of operation with a neat diagram. (4M)
8. What are the five major activities of an operating system in regard to process management, memory and Mass-storage management? (6M)
9. List services provided by an operating system.(6M)
10. What are client server systems & Peer-to-Peer systems? (4M)
11. What are system calls? What are types of system calls? (7 M)
12. Write and explain the sequence of system calls for copying a file to another (new) file.(4M)
13. What are system programs? List types of system programs (7M)
14. **Explain the layered, microkernel and modules approach to structuring the operating system along with relevant diagrams**. (6M)
15. Explain the concepts of virtual machine. What is the main advantage of virtual machine for an operating-system designer and a user? (7M)

16. Explain VMware and Java virtual machine. (5M)
17. What is a process? Describe process states with the help of process transition diagram. (6)
18. Give the information that is kept in process control block? What is the need for context-switch between processes? (6M)
19. Describe the differences among short-term, medium-term, and long term scheduling. (6M)
20. Why it is important for scheduler to distinguish I/O bound programs from CPU bound programs.(4M)
21. Distinguish between : (9M)
    a. Process and Program
    b. Multiprogramming and multiprocessing
    c. Job scheduling and CPU scheduling
22. Explain how process are created and terminated.(5M)
23. Explain direct and indirect communication with respect to message passing system. (9M)
24. What are co-operating processes? Describe the mechanism of inter process. communication using shared memory in a producer-consumer problem.(5M)

MODULE-II

Chapter Multitherading

1. What are threads? How do they differ from a process? (4M)
2. Discuss three common ways of establishing relationship between the user thread and kernel thread. (or) Explain multithreading models.(6M)
3. Differentiate between user level threads and kernel level threads.(4M)
4. What are the different ways in which threads terminate.(4M)

Chapter- Process Scheduling

5. Explain different process scheduling criteria's. (5M)
6. What is the difference between preemptive and non-preemptive scheduling algorithms? (2M)
7. Explain the different types of scheduling algorithms.(8M)
8. Explain multilevel Queue scheduling, Multilevel feedback-Queue scheduling and Multiprocessor scheduling. (10M)

9. Consider the following data with burst time given in milliseconds: (5M)

Process	Burst time	Priority
p1	10	3
p2	1	1
p3	2	3
p4	1	4
p5	5	2

All the processes have arrived in the order p1, p2, p3, p4, p5 at time 0.

a. Draw Gantt charts for the execution of these processes using FCFS, SJF, a non-preemptive priority and RR (quantum=1) scheduling.

b. What is the turnaround time and waiting time of each process for each of the scheduling algorithm and also calculate average waiting time?

10. Suppose the following jobs arrive for processing at the times indicated, each job will run the listed amount of time. (10M)

Job	arrival time	burst time
1	0.0	8
2	0.4	4
3	1.0	1

i) Give a Gantt chart illustrating the execution of these jobs using the non preemptive FCFS and SJF scheduling algorithms.

ii) What is turnaround time and wait time of each job for the above algorithms?

iii)Compute average turnaround time if the CPU is left idle for the first 1 unit and then SJF scheduling is used.(job 1 and job 2 will wait during this time)

11. Consider the following set of processes, with the length of CPU burst in milliseconds. (8M)

Process	PI P2 P3 P4 P5
Arrival time	00 02 03 06 30
Burst time	10 12 14 16 05

a. Draw a Gantt chart that illustrates the execution of these processes using the preemptive shortest job first (SJF) algorithm. Hence find the average waiting time.

b. Draw a Gantt chart that illustrates the execution of these processes using preemptive priority scheduling algorithm. Given priority of each process is PI = 4, P2=3, P3=5, P4= 1 and P5= 1. Also find the average waiting time.

12. Consider the following set of processes, with the length of CPU burst in milliseconds. (9M)

Process PI P2 P3 P4
Arrival time 00 01 02 03
Burst time 10 29 03 07

Draw a Gantt chart and calculate waiting and turnaround time for FCFS, SJF, RR for time quantum of 10. Also calculate average waiting time.

13. Consider the following set of processes with their arrival and burst times as shown (12M)

Process	**A.T**	**B.T**
P0	0	6
P1	1	3
P2	2	1
P3	3	4

a) Draw a Gantt chart that illustrates the execution of these processes using the SRTF and non preemptive SJF
b) Find turnaround time and waiting time for each process. Hence show that SRTF is faster the SJF.

14. For the processes listed below draw gantt charts using preemptive and non preemptive priority scheduling algorithm. A larger priority number has higher priority. (5M)

Process	**A.T**	**B.T**	**Priority**
P0	0	6	4
P1	3	5	2
P2	3	3	6
P3	5	5	3

15. Consider 5 processes arrive a time 0 in the order givenwith thw length of CPU burst given in milliseconds.

Process	Burst Time
P1	10
P2	29
P3	3
P4	7

23. Build the solution to Dinning philosopher problem using monitors.
24. Mention three classical problems of process synchronization. Explain any one in detail.
25. What are monitors? Compare with semaphore with their relative advantages and disadvantages.
26. Write and explain with respect to sunchronization hardware
    a. The definition of TestAndSet() instruction.
    b. Mutual exclusion implementation with TestAndSet().

MODULE-3

Chapter-Deadlocks

1. What are deadlocks? What are its characteristics?
2. What is deadlock? Explain the necessary conditions for its occurrence. Indicate how many of these should ouccur for deadlock to happen?
3. Build Safety algorithm using necessary data structure for deadlock avoidance. (or) Explain Bankers algorithm for deadlock avoidance.
4. Make use of suitable example and construct resource allocation graph (RAG) for the follows: i) With Deadlock ii) With cycle but no Deadlock.
5. What is wait for graph? Explain how it is useful for detection of deadlocks.
6. Deadlock exists if cycle exists. Yes or no. Justify your answer with a suitable example.
7. Consider a system containing m resources of same type being shared by n processes. Resources can be requested and released by processes only one at a time. Show that system is deadlock free if the following two conditions hold.
    a. The maximum need of each process is between 1 and m resources.
    b. The sum of all maximum needs is less than m+n.
8. A system has 5 processes; P0, P1, P2, P3 and P4.There are 3 types of resources R1, R2 and R3.there are 10 instances of R1, 5 instances of R2 and 7 instances of R3. At time T0, the situation is as follows:

Process	Allocation			Max Required		
	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	5	3
P1	2	0	0	3	2	2
P2	3	0	2	9	0	2
P3	2	1	1	2	2	2
P4	0	0	2	4	3	3

i) Calculate Need and Available matrix.
ii) Is the system in a safe state at time T0? If yes give the safe Sequence.

P5	12

Consider FCFS, SJR, RR(TQ=10ms) scheduling, draw gantt chart for each of the scheduling algorithm. Determine average waiting time and turn around time for all three algorithms. Which algorithm would give minimum average waiting time.

16. For process listed below draw gantt chart using preemptive and non preemptive priority scheduling algorithm. A larger priority number has higher priority. Calculate average waiting time and average turnaround time.

Jobs	Arrival time	Burst time	Priority
J1	0	6	4
J2	3	5	2
J3	3	3	6
J4	5	5	3

Is CPU scheduling necessary? Discuss the five different scheduling criteria's used in computing scheduling algorithms.

17. Assume the following workload in a system. All jobs arrive at time 0 in the order
given.

Process Burst Time Priority
P1 30 High
P2 28 High
P3 04 Low
P4 16 Medium

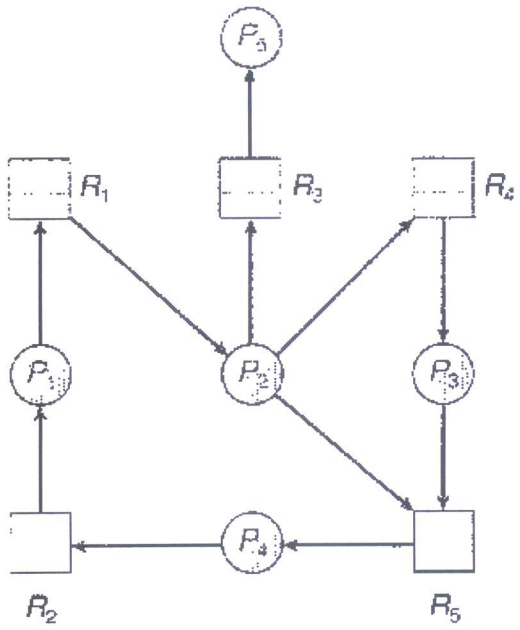
Draw a Gantt chart illustrating the execution of these jobs using Priority CPU scheduling algorithm and also Calculate the average waiting time and average turnaround time

Chapter- Process Synchronization

15. Identify and explain three requirements' that a solution to critical section problem must satisfy.
16. What is race condition? List three requirements' that a solution to critical section problem must satisfy.
17. Apply Peterson's solution to critical section problem and explain.
18. Define semaphore. Explain its usage and implementation.
19. Identify the use of binary and counting semaphore and write the C structure of wait () and signal ( ) operations.
20. Utilize semaphore and give solution to Reader- Writers problem.
21. Develop the solution to producer consumer problem using semaphore.
22. Apply TestAndSet() and Swap() instructions for implementation of mutual exclusion.

iii) Suppose at time T1, process P1 requests one additional instance of resource type R1and 2 instances of resource type R3, supposing the request if granted will the system be in a safe state?

9. For the following resource allocation graph, draw the corresponding wait-for graph.



10. Consider the following snapshot of the system and answer the following questions using Banker's algorithm?

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	0	0	1	2	0	0	1	2	1	5	2	0
P2	1	0	0	0	1	7	5	0				
P3	1	3	5	4	2	3	5	6				
P4	0	6	3	2	0	6	5	2				
P5	0	0	1	4	0	6	5	6				

a. Find the need of the allocation?

b. Is the system is in safe state?

c. If the process P1 request (0,4,2,0) resources cam the request be granted immediately?

11. The operating system contains three resources. The numbers of instances of each resource type are (7, 7,10). The current allocation state is given below.

a. Is the current allocation is safe?

b. find need?

c. Can the request made by the process P1(1,1,0) can be granted?

Process	Allocation			Max		
	R1	R2	R3	R1	R2	R3
P1	2	2	3	3	6	8
P2	2	0	3	4	3	3
P3	1	2	4	3	4	4

12. Explain different methods to recover from deadlock?
13. Consider the following snapshot of resource allocation at time t1.

Process	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	0	0	0	0	0	0
P1	2	0	0	2	0	2			
P2	3	0	3	0	0	0			
P3	2	1	1	1	0	0			
P4	0	0	2	0	0	2			

a) Show that the system is not deadlocked by generating one safe sequence.
b) At instance t1, P2 makes one additional request for instance of type C. Show that the system is deadlocked if the request is granted. Write down the deadlocked processes.

Chapter- Memory Management

14. Explain the multistep processing of a user program with a neat block diagram.
15. What is address binding? Explain the concept of dynamic relocation of addresses.
16. Make use of an example and explain the drawbacks of contiguous memory allocation, first fit, best fit and worst fit memory allocation methods.
17. Given the memory partitions of 100K, 500K, 200K, 300K and 600K apply first fit and best fit algorithm to place 212K, 417K, 112K, and 426K.
18. What is swapping? Does this increase the operating system overhead? Justify your answer.
19. What is the principle behind paging? Explain its operation, clearly indicating how the logical addresses are converted to physical addresses
20. Develop comparison study between the following:
    a) Logical and physical address space.
    b) Internal and External fragmentation
21. Construct hardware support for TLB paging scheme with a neat diagram and explain in detail.

22. In paging scheme with TLB, it takes 20ns to search the TLB and 100ns to access memory. Find the effective access time and percentage slowdown in memory access time if
    i. Hit ratio is 80%
    ii. Hit ratio is 98%
23. Build an example segment table using Segmentation memory management hardware and explain.
24. Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2		
90	100	
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?
a) 0,430
b) 1,10
c) 2,500
d) 3,400

MODULE-IV

Chapter- Virtual Memory

1. Explain steps involved page fault/ Explain performance of Demand paging
2. Explain different page replacement algorithms
3. What is Balady's anomaly? Explain with an example
4. Apply FCFS, LRU, Optimal for following reference strings 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1 and find the number of page faults
5. Apply FCFS, LRU, Optimal for following reference strings 1,2,3,4,1,2,5,1,2,3,4,5 and find the number of page faults
6. Apply FCFS, LRU, Optimal for following reference strings 1,3,4,0,5,3,2,1,0,4,5,2 and find the number of page faults
7. For the following page references calculate the page faults that occur using FIFO and LRU for 3 and 4 page frames respectively 5,4,3,2,1,4,3,5,4,3,2,1,5.
8. Consider the following page references string 1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,1,2,3,1. How many page faults would occur for the following page replacement algorithms assuming 3 and 5.
    a. LRU

b. Optimal

9. Explain thrashing phenomenon.
10. Explain working set model.

Chapter- File System

11. Explain Types of Directory Structures
12. List and explain different file attributes
13. List and explain different file operations
14. List and explain different file types with their functions and extensions
15. Explain Sequential and Direct file access methods
16. Explain file system mounting
17. Explain different ways of File Sharing
18. Explain Linear and Hash table methods of directory implementation
19. Explain File Control Block(FCB)
20. Explain the methods of free space is management
21. What are files? Explain contiguous, Linked and Index methods of file allocation
22. Explain Layered approach of file structure.
23. Explain Directory operations

MODULE-V

Chapter- Disk Management

1. Explain Boot block
2. Explain Bad Blocks
3. Explain various disk scheduling algorithms
4. Disk Scheduling Problems
5. Consider a disk queue with requests for I/O to blocks on cylinders 98,183,37,122,14,124,65,67 in that order. Assume the disk head is initially at cylinder 53
   a. Show diagrammatically as well as explain the FCFS disk scheduling and SSTF disk scheduling
   b. Find out total head movements for FCFS SSTF algorithm
   c. Describe the performance of both the algorithms.
6. Consider a disk queue with requests for I/O to blocks on cylinders 95,180,34,119,11,121,62,64 in that order. Assume the disk head is initially at cylinder 53
   a. Show diagrammatically as well as explain the FCFS disk scheduling and SSTF disk scheduling
   b. Find out total head movements for FCFS SSTF algorithm

c. Describe the performance of both the algorithms.

7. Suppose that the disk has 50 cylinders named 0 to 49. The R/W head is currently serving at cylinder 15. The queue of pending requests in order : 4,40,11,35,7,14 starting from the current head position. What is the total distance travelled (in cylenders) by the disk arm to satisfy the requests using algorithms FCFS, SSTF and LOOK. Illustrate with figure in each case.
8. Suppose that the disk has 5000 cylinders named 0 to 4999. The R/W head is currently serving at cylinder 125. The queue of pending requests in order : 1470, 913, 1774, 948, 1509, 1022, 1750, 130 starting from the current head position. What is the total distance travelled (in cylenders) by the disk arm to satisfy the requests using algorithms FCFS, SCAN and LOOK. Illustrate with figure in each case.
9. Consider a disk queue with following requests for I/O to blocks on cylinders 30,70,115,130.110,80,20,25 (Assume disk head is at 90) Draw FCFS and SSTF scheduling and also determine how many times the disk head changes its direction for each of the above mentioned scheduling techniques.

Chapter- Protection

10. Explain Goals and principles of protection
11. What is protection?Distinguish between policies and mechanism. Explain access matrix with domains as objects
12. Differentiate between protection and security.
13. Write a note on
    a. Domain Protection
    b. Access Matrix
14. Explain Access matrix of domain and objects as implementation mechanism used for protection by operating system
15. Explain Access matrix of domain and objects with copy, transfer and owner, switch and control rights.
16. Describe the scheme of capability list to implement protection.
17. Write short note on
    a. Swap space management
    b. Revocation of access rights.

Chapter- Linux Case Study

18. Explain the different components of a Linux system
19. Discuss how memory management is dealt with in Linux system.
20. Explain different IPC mechanisms available in Linux
21. Explain how process is managed on Linux platform
22. Write a brief note on design principle of Linux
23. Explain in detail the components that the kernel supports under Linux

24. Explain process scheduling in a Linux system
25. Explain file system implementation in Linux
26. What do you mean by cloning? How is it achieved in Linux system?
27. Explain the following with respect to process management in Linux
    a. Process Identity
    b. Process Context
28. With a neat diagram explain the memory layout for ELF programs in linux
29. How does Linux manage authentication and access control mechanism.

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

Sixth Semester B.E. Degree Examination, Aug./Sept. 2020
Operating Systems

Time: 3 hrs. Max. Marks: 80

Note: ***Answer any FIVE full questions, choosing ONE full question from each module.***

Module-1

1 a. Define operating systems. What are multiprocessor systems? Explain their three main advantages. **(05 Marks)**

b. Compare multi-programming and time sharing systems. **(05 Marks)**

c. Point out and explain the various operating system services. **(06 Marks)**

OR

2 a. What are microkernals? Point out their advantages. **(05 Marks)**

b. What are the two models of inter process communications? What are the strengths and weakness of the two approaches? **(05 Marks)**

c. Compare and contrast, short term, medium term and long – term scheduling. **(06 Marks)**

Module-2

3 a. Point out and explain the various benefits of multi threaded programming. **(04 Marks)**

b. Consider the five processes arrive at time 0, in the order given, with the length of the CPU burst given in milliseconds.

Process	Burst time
P_1	10
P_2	29
P_3	3
P_4	7
P_5	12

Consider the FCFS, SJF and RR (quantum = 10ms) scheduling, draw the Gantt chart for each of the scheduling. Determine average waiting time and turnaround time for all the 3 scheduling algorithm. Which algorithm would give the minimum average waiting time? **(12 Marks)**

OR

4 a. What is the critical section problem point out and explain its three requirements. **(05 Marks)**

b. What are semaphores, explain how mutual exclusion is implemented with semaphores. **(05 Marks)**

c. What is Dimming philosopher problem explain its monitor solution. **(06 Marks)**

Module-3

5 a. What are deadlocks? Point out and explain its necessary conditions. **(04 Marks)**

b. Explain the various methods of recovery from deadlock. **(05 Marks)**

c. Consider a system with five processes P_0 through P_4 and three resources types A, B and C. Resource type A has 10 instances, resource type B has 5 instances and resource type C has 7 instances suppose that, at time T_0, the following snapshot of the system.

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

Draw the need matrix.

The sequence $< P_1, P_3, P_4, P_2, P_0 >$ is safe state or not. **(07 Marks)**

OR

6 a. Define paging. Explain paging hardware with a neat block diagram. **(08 Marks)**

b. What is segmentation? Explain basic method of segmentation with an example. **(08 Marks)**

Module-4

7 a. What is demand paging? Explain the steps in handling a page fault with a neat diagram. **(08 Marks)**

b. Consider the following sequence

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

How many page faults occurs with three page frames :

i) FIFO

ii) Optimal page replacement

iii) LRU page replacement algorithm. **(08 Marks)**

OR

8 a. What is a file? What are its attributes, explain file operations. **(06 Marks)**

b. Explain what are the different types of files. **(05 Marks)**

c. Explain file system mounting. **(05 Marks)**

Module-5

9 a. Explain various disk scheduling algorithm with an example. **(10 Marks)**

b. Explain access matrix protection system of O.S. **(06 Marks)**

OR

10 a. Explain the various. Components of the Linux system. **(08 Marks)**

b. Explain the process management in Linux. **(08 Marks)**

Sixth Semester B.E. Degree Examination, Dec.2019/Jan.2020
Operating Systems

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. What is operating system? Explain multiprogramming and time sharing systems. **(06 Marks)**

b. Explain dual mode operating in operating system with a neat block diagram. **(05 Marks)**

c. What are system calls? Briefly point out its types. **(05 Marks)**

OR

2 a. Explain process states with state transition diagram. Also explain PCB with a neat diagram. **(06 Marks)**

b. What is interprocess communication? Explain its types. **(05 Marks)**

c. With a neat diagram, explain the concept of virtual machines. **(05 Marks)**

Module-2

3 a. For the process listed below, draw Gantt charts using pre-emptive and non-preemptive priority scheduling algorithm. A larger priority number has a higher priority. Calculate Average Weighing Time and Average turnaround time.

Jobs	Arrival Time	Burst Time	Priority
J_1	0	6	4
J_2	3	5	2
J_3	3	3	6
J_4	5	5	3

(06 Marks)

b. Is CPU scheduling necessary? Discuss the five different scheduling criterias used in the computing scheduling mechanism. **(05 Marks)**

c. Explain multithreading models. **(05 Marks)**

OR

4 a. Define semaphores. Explain its usage and implementation. **(06 Marks)**

b. Explain Reader-Write problem with semaphore in detail. **(05 Marks)**

c. What are monitors? Explain dining Philospher's solution using monitor. **(05 Marks)**

Module-3

5 a. System consists of five jobs (J_1, J_2, J_3, J_4, J_5) and three resources (R_1, R_2, R_3). Resource type R_1 has 10 instances, resource type R_2 has 5 instances and R_3 has 7 instances. The following snapshot of the system has been taken.

Jobs	Allocation			Maximum			Available		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
J_1	0	1	0	7	5	3	3	3	2
J_2	2	0	0	3	2	2			
J_3	3	0	2	9	0	2			
J_4	2	1	1	2	2	2			
J_5	0	0	2	4	3	3			

Find need matrix and calculate the safe sequence by using Banker's algorithm. Mention the above system is safe or not safe. **(06 Marks)**

b. What is dead lock? What are necessary conditions an operating system must satisfy for a dead lock to occur? **(05 Marks)**

c. What is a Resource Allocation Graph (RAG)? Explain how RAG is very useful is describing deadly embrace by considering own example. **(05 Marks)**

OR

6 a. What are Translation Load aside Buffer (TLB)? Explain TLB in detail with a simple paging system with a neat diagram. **(06 Marks)**

b. Given the memory partitions of 100 K, 500 K, 200 K, 300 K and 600 K apply first fit, best fit and worst fit algorithms to place 212K, 417K, 112K and 426K. **(05 Marks)**

c. Describe both internal and external fragmentation problems encountered in a contiguous memory allocation scheme. **(05 Marks)**

Module-4

7 a. Consider the following page reference stream: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1. How many page faults would occur for LRU and FIFO replacement algorithms assuming 3 frames? Which one of the above is most efficient? **(06 Marks)**

b. Explain demand paging system. **(05 Marks)**

c. What is thrashing? How can it be controlled? **(05 Marks)**

OR

8 a. Explain briefly the various operations performed on files. **(06 Marks)**

b. Explain the various access methods of files. **(05 Marks)**

c. Explain various allocation methods in implementing file systems. **(05 Marks)**

Module-5

9 a. Explain the various Disk Scheduling algorithms with example. **(08 Marks)**

b. Explain access matrix method of system protection. **(08 Marks)**

OR

10 a. With a neat diagram explain in detail components of a Linux system. **(06 Marks)**

b. Explain the different IPC mechanisms available in Linux. **(05 Marks)**

c. Explain process scheduling in a Linux system. **(05 Marks)**

CBCS SCHEME

USN | 1 | KS | 16 | CS | 01 | 7 |

15CS64

Sixth Semester B.E. Degree Examination, June/July 2019
Operating Systems

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Explain the role of operating system from different viewpoints. Explain the dual mode of operation of an operating system. **(07 Marks)**

b. Demonstrate the concept of virtual machine with an example. **(05 Marks)**

c. Explain the types of multiprocessing system and the types of clustering. **(04 Marks)**

OR

2 a. Describe the implementation of interprocess communication using shared memory and message passing. **(06 Marks)**

b. Demonstrate the operations of process creation and process termination in UNIX. **(06 Marks)**

c. Explain the different states of a process, with a neat diagram. **(04 Marks)**

Module-2

3 a. Discuss the threading issues that come with multithreaded program. **(08 Marks)**

b. Illustrate how Reader's-Writer's problem can be solved by using semaphores. **(08 Marks)**

OR

4 a. Calculate the average waiting time by drawing Gantt chart using FCFS (First Come First Serve), SRTF (Shortest Remaining Time First), RR (Round Robin) [q = 2 ms] algorithms.

Process	Arrival time	Burst time
P_1	0	9
P_2	1	4
P_3	2	9
P_4	3	5

(08 Marks)

b. Explain the Dining-Philosopher's problem using monitors. **(08 Marks)**

Module-3

5 a. Determine whether the following system is in safe state by using Banker's algorithm.

Process	Allocation			Maximum			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	0	4	3	3			

If a request for P_1 arrives for (1 0 2), can the request be granted immediately? **(09 Marks)**

b. Discuss the various approaches used for deadlock recovery. **(07 Marks)**

OR

6 a. Illustrate with example, the internal and external fragmentation problem encountered in continuous memory allocation. **(07 Marks)**

b. Explain the structure of page table. **(09 Marks)**

Module-4

7 a. Illustrate how demand paging affects systems performance. **(08 Marks)**

b. Describe the steps in handling a page fault. **(08 Marks)**

OR

8 a. Explain the various types of directory structures. **(08 Marks)**

b. Describe various file allocation methods. **(08 Marks)**

Module-5

9 a. Explain the access matrix model of implementing protection in operating system. **(07 Marks)**

b. Explain the following disk scheduling algorithm in brief with examples:

i) FCFS scheduling
ii) SSTF scheduling
iii) SCAN scheduling
iv) LOOK scheduling **(09 Marks)**

OR

10 a. Explain the components of LINUX system with a neat diagram. **(08 Marks)**

b. Explain the way process is managed in LINUX platform. **(08 Marks)**

* * * * *

CBCS SCHEME

USN | 1 | K | S | 1 | 6 | C | S | 4 | 0 | 6 | 15CS64

Sixth Semester B.E. Degree Examination, Dec.2018/Jan.2019
Operating Systems

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Distinguish between the following terms :
i) Multiprogramming and multitasking
ii) Multiprocessor systems and clustered systems. **(04 Marks)**

b. Analyze modular kernel approach with layered approach with a neat sketch. **(06 Marks)**

c. List and explain the services provided by OS for the user and efficient operation of system. **(06 Marks)**

OR

2 a. Illustrate with a neat sketch, the process states and process control block. **(08 Marks)**

b. Discuss the methods to implement message passing IPC in detail. **(08 Marks)**

Module-2

3 a. Discuss the benefits of multithreaded programming. **(04 Marks)**

b. Consider the following set of processes with CPU burst time (in ms).

Process	Arrival time	Bunt time
P1	0	6
P2	1	3
P3	2	1
P4	3	4

Compute the waiting time and average turnaround time for the above process using FCFS, SRT and RR (time quantum = 2ms) scheduling algorithm. **(12 Marks)**

OR

4 a. Illustrate with examples the Peterson's solution for critical section problem and prove that the mutual exclusion property is preserved. **(08 Marks)**

b. Show how semaphore provides solution to reader writers problem. **(08 Marks)**

Module-3

5 a. Define deadlock. Write short notes on 4 necessary conditions that arise deadlocks. **(06 Marks)**

b. Assume that there are 5 processes PO through P4 and 4 types of resources. At time T_0 we have the following state :

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

Apply Banker's algorithm to answer the following :

i) What is the content of need matrix?

ii) Is the system in a safe state?

iii) If a request from a process P1(0, 4, 2, 0) arrives, can it be granted? **(10 Marks)**

OR

6 a. Write short notes on :

i) External and internal fragmentation

ii) Dynamic loading and linking. **(04 Marks)**

b. Analyze the problem in simple paging technique and show how TLB is used to solve the problem. **(08 Marks)**

c. Given the memory partitions of 200k, 700k 500k, 300k, 100k, 400k. Apply first fit and best fit to place 315k, 427k, 250k, 550k. **(04 Marks)**

Module-4

7 a. For the following page reference string 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5. Calculate the page faults using FIFO and LRU for memory with 3 and 4 frames. **(08 Marks)**

b. Explain demand paging in detail. **(08 Marks)**

OR

8 a. What do you mean by free space list? With suitable example, explain any 3 methods of free space list implementation. **(08 Marks)**

b. Write short notes on linked and indexed allocation method with a neat diagram. **(08 Marks)**

Module-5

9 a. Given the following sequences 95, 180, 34, 119, 11, 123, 62, 64 with the head initially at track 50 and ending at track 199. What is the total disk traveled by the disk arm to satisfy the request using FCFS, SSTF, LOOK and CLOOK algorithm. **(12 Marks)**

b. Write short notes on access matrix and its implementations. **(04 Marks)**

OR

10 a. Explain the components of Linux system with a neat diagram. **(08 Marks)**

b. Describe briefly on Linux Kernel modules. **(08 Marks)**

* * * * *

Sixth Semester B.E. Degree Examination, June/July 2018
Operating Systems

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing one full question from each module.

Module-1

1 a. Define Operating System. with a neat diagram, explain the dual-mode operation of operating system. **(06 Marks)**

b. Explain the services of operating system that are helpful for user and the system. **(06 Marks)**

c. Define the following terms :
   i) Virtual Machine
   ii) CPU scheduler
   iii) System call
   iv) Context switch. **(04 Marks)**

OR

2 a. With a neat diagram, explain the different states of a process. **(05 Marks)**

b. Explain the layered approach of operating system structure, with supporting diagram. **(05 Marks)**

c. What is interprocess communication? Explain direct and indirect communication with respect to message passing system. **(06 Marks)**

Module-2

3 a. Explain multithreading models, Also list the benefits of multithreaded programming. **(06 Marks)**

b. Consider the following set of processes given in table

Processes	Arrival Time (m sec)	Burst Time (m sec)	Priority
P_1	0	10	4
P_2	3	5	2
P_3	3	6	6
P_4	5	4	3

Consider larger number as highest priority. Calculate average waiting time and turn around time and draw Gantt chart for preemptive priority scheduling and preemptive SJF scheduling. **(06 Marks)**

c. Explain multiprocessor scheduling. **(04 Marks)**

OR

4 a. What are the requirements to critical section problem? Explain Peterson's solution to critical section problem. **(06 Marks)**

b. Explain Dining-philosophers problem with semaphores. **(05 Marks)**

c. Explain the syntax and schematic view of monitors. **(05 Marks)**

Module-3

5 a. Consider the following snapshot of a system

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	0	2	0	0	4	1	0	2
P_1	1	0	0	2	0	1			
P_2	1	3	5	1	3	7			
P_3	6	3	2	8	4	2			
P_4	1	4	3	1	5	7			

Find the need matrix and calculate safe sequence using Banker's algorithm. Mention the above system is safe or not safe. **(08 Marks)**

b. What are the necessary conditions for deadlock? Explain different methods to recover from deadlock. **(08 Marks)**

OR

6 a. What is paging? Explain paging hardware with translation look-aside buffer. **(06 Marks)**

b. Explain the structure of page table with respect to hierarchical paging. **(06 Marks)**

c. Given the 5 memory partitions 100 KB, 500 KB, 200 KB, 300 KB and 600 KB, how each of the first fit, best fit and worst fit algorithms place processes of 212 KB, 417 KB, 112KB and 426KB size. Which algorithm makes efficient use of memory? **(04 Marks)**

Module-4

7 a. What is a page fault? With a supporting diagram explain the steps involved in handling page fault. **(06 Marks)**

b. Consider the page reference string for a memory with three frames, how many page faults will occur for FIFO, LRU and optimal page replacement algorithms. Which is most efficient?

Reference string : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

(06 Marks)

c. Explain copy-on-write process in virtual memory. **(04 Marks)**

OR

8 a. What are the different allocation methods in disk? Explain in detail any two methods. **(06 Marks)**

b. List the different directory structure. Explain acyclic – graph directory and tree structured directory. **(07 Marks)**

c. What is a file? Also list different file operations. **(03 Marks)**

Module-5

9 a. List the different disk scheduling techniques, explain any two scheduling, considering the following disk queue requests.
98, 183, 37, 122, 14, 124, 65, 67. **(06 Marks)**

b. What is an access matrix? Explain the different methods of implementing access matrix. **(06 Marks)**

c. Explain bad – block recovery in disk. **(04 Marks)**

OR

10 a. Explain the design principle of Linux. **(06 Marks)**

b. Explain the process management in Linux platform. **(06 Marks)**

c. Explain the interprocess communication mechanisms in Linux. **(04 Marks)**

USN | | | | | | | | | | |

10CS53

Fifth Semester B.E. Degree Examination, June/July 2015
Operating Systems

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting atleast TWO questions from each part.***

PART – A

1 a. What are the activities for which the operating system is responsible for, in connection with : i) Process management ii) File management. **(10 Marks)**

b. Explain any two types of system calls. **(05 Marks)**

c. What are virtual machines? Explain the benefit of creating virtual machines. **(05 Marks)**

2 a. With a diagram, explain different states of a process. **(04 Marks)**

b. Differentiate between direct and indirect interprocess communication. **(04 Marks)**

c. Explain any three multithreading models in brief. **(03 Marks)**

d. Consider the following set of processes :

Process	Arrival time	Burst time
P_1	0	5
P_2	1	1
P_3	2	4

Compute average turn around d time and average waiting time using FCFS, preemptive SJF and RR (quantum – 4). **(09 Marks)**

3 a. Explain Peterson's solution to critical section problem. **(06 Marks)**

b. Describe the mutual – exclusion implementation with TestAndSet(). **(06 Marks)**

c. Mention three classical problems of synchronization. Explain any one in detail. **(08 Marks)**

4 a. Consider the following snapshot of a system :

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

Answer the following questions using banker's algorithm

i) What is the content of the matrix need?

ii) Is the system in a safe state?

iii) If a request from process P_1 arrives for (1, 0, 2), can the request be granted immediately? **(12 Marks)**

b. For the following resource-allocation graph, write the corresponding wait – for graph. **(04 Marks)**

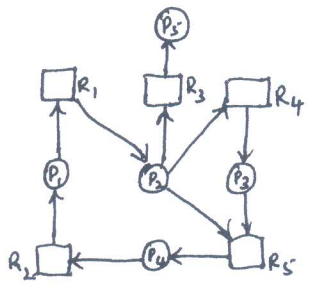


Fig. Q4(b)

c. Explain the different methods used to recover from deadlock. **(04 Marks)**

PART – B

5 a. With a supporting paging hardware, explain in detail concept of paging with an example for a 32 –byte memory with 4 – type pages with a process being 16-bytes. How many bits are reserved for page number and page offset in the logical address. Suppose the logical address is 5, calculate the corresponding physical address, after populating memory and page table. **(10 Marks)**

b. Discuss on the performance of demand paging. **(05 Marks)**

c. What is Belady's anomaly? Explain with an example. **(05 Marks)**

6 a. Mention any five :
i) File attributes
ii) File operations. **(05 Marks)**

b. With supporting diagrams distinguish between single-level and two-level directory structure. **(05 Marks)**

c. Compare contiguous and linked allocation methods for disk space. **(05 Marks)**

d. Explain bit vector free-space management technique. **(05 Marks)**

7 a. With an illustrative example, distinguish between SSTF, FCFS, SCAN and LOOK DISK schedulings. **(08 Marks)**

b. What are boot block and bad blocks? Explain. **(06 Marks)**

c. Explain the goals and principles of protection. **(06 Marks)**

8 Write short notes on :
a. Design principles of Linux system
b. Linux virtual memory system
c. Segmentation
d. LRU page replacement algorithm. **(20 Marks)**

* * * * *

Code No: R1631055 **R16** SET - 1

III B. Tech I Semester Regular Examinations, October/November - 2018

OPERATING SYSTEMS

(Common to Computer Science Engineering, Information Technology)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A

1. a) List out the services provided by an operating system. [2M]
   b) List Fields of Process Control Block. [2M]
   c) What is Virtual Address Space? [2M]
   d) What is Resource-Allocation-Graph. [3M]
   e) What are the two ways of accessing disk storage? [3M]
   f) What is an Activity Stack in Android? [2M]

PART -B

2. With a neat sketch, Explain in detail about the interrelation between various services provided by the operating system. [14M]

3. a) Explain in detail, the sequence of actions taken by the operating system to context switch between processes. [6M]
   b) Assume the following workload in a system. All jobs arrive at time 0 in the order given. [8M]

Process	Burst Time	Priority
P1	30	High
P2	28	High
P3	04	Low
P4	16	Medium

Draw a Gantt chart illustrating the execution of these jobs using Priority CPU scheduling algorithm and also Calculate the average waiting time and average turnaround time.

4. a) What is a page fault? Explain the steps involved in handling a page fault with a neat sketch. [7M]
   b) Explain the process of converting virtual addresses to physical addresses with a neat diagram. [7M]

5. a) What is a Critical Section problem? Give the conditions that a solution to the critical section problem must satisfy. [7M]
   b) What is a deadlock? How deadlocks are detected? [7M]

6. a) Briefly explain about single-level, two-level and Tree-Structured directories. [6M]
   b) Consider a disk queue with following requests for I/O to blocks on cylinders 30,70,115,130.110,80,20,25 (Assume disk head is at 90) [8M]
   Draw FCFS and SSTF scheduling and also determine how many times the disk head changes its direction for each of the above mentioned scheduling techniques.

7. a) Discuss the advantages and several disadvantages of open-source Operating systems. [6M]
   b) Discuss in detail about Android Runtime Application Development. [8M]
~~~~~~

Code No: R1631055 **R16** SET - 2

III B. Tech I Semester Regular Examinations, October/November - 2018

OPERATING SYSTEMS

(**Common to** Computer Science Engineering, Information Technology)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A

1.	a)	List out the types of System calls.	[3M]
	b)	What is Multi-Threading?	[2M]
	c)	What is the Cause of Thrashing?	[2M]
	d)	What is Process Synchronization?	[3M]
	e)	What is a device driver	[2M]
	f)	What is the Dalvik Virtual machine in Android?	[2M]

PART -B

2.	a)	Discuss various operating system services that are very much helpful to the user.	[7M]
	b)	What are the advantages and disadvantages of using the same system call interface for manipulating both files and devices?	[7M]
3.	a)	With a neat diagram, explain the process state diagram.	[6M]
	b)	What are the advantages of inter-process communication? How communication takes place in a shared-memory environment? Explain.	[8M]
4.	a)	How demand paging affects the performance of a computer system? Give explanation.	[7M]
	b)	How does the system detect Thrashing? What can the system do to eliminate this problem? Explain.	[7M]
5.	a)	What happens if the wait() and signal() semaphore operations are not executed atomically? Give explanation.	[7M]
	b)	What is Dining Philosophers problem? Discuss the solution to Dining philosopher's problem using monitors.	[7M]
6.	a)	Discuss various file access methods in detail.	[7M]
	b)	Explain about various issues involved in selecting appropriate disk scheduling algorithm.	[7M]
7.		Explain each and every component of the Android architecture with a neat sketch.	[14M]

*****
~~~~~~

Code No: R1631055 **R16** SET - 3

III B. Tech I Semester Regular Examinations, October/November - 2018

OPERATING SYSTEMS

(**Common to** Computer Science Engineering, Information Technology)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A

1. a) Define Operating System. [2M]
   b) What is Process control block? [3M]
   c) Differentiate between Logical and Physical address space. [3M]
   d) State the Critical Section problem. [2M]
   e) What are the most common attributes that are associated with an opened file? [2M]
   f) What is aninode in LINUX? [2M]

PART -B

2. a) Explain the objectives and functions of Operating system. [7M]
   b) Explain in detail about the functions of System Call Interface. [7M]

3. a) Explain the process scheduling with a neat queuing diagram. [7M]
   b) Assume the following workload in a system. All jobs arrive at time 0 in the order given. [7M]

Process	Burst Time	Order
P1	4	1
P2	5	2
P3	3	3

Draw a Gantt chart illustrating the execution of these jobs using Round Robin CPU scheduling algorithm (Assume time quantum= 1 unit) and also Calculate the average waiting time and average turnaround time.

4. a) Explain the difference between External fragmentation and Internal fragmentation. How to solve the fragmentation problem using paging. [7M]
   b) Consider the following page reference string: [7M]
   1,2,4,7,3,5,6,3,6,1,4,2,3,6,5,2
   How many page faults would occur for the optimal page replacement algorithm, assuming four frames and all frames are initially empty.

5. a) Discuss Mutual-exclusion implementation with Test And Set () instruction. [7M]
   b) Discuss various techniques to recover from the deadlock. [7M]

6. a) Explain in detail about various ways of accessing disk storage. [4M]
   b) Explain the various methods for free-space management. [10M]

7. a) How Interrupts and Exceptions are handled in LINUX? Explain. [7M]
   b) Explain about the process lifecycle in Android. [7M]

*****
~~~~~~

Code No: R1631055 **R16** SET - 4

III B. Tech I Semester Regular Examinations, October/November - 2018

OPERATING SYSTEMS

(**Common to** Computer Science Engineering, Information Technology)

Time: 3 hours Max. Marks: 70

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)
2. Answer **ALL** the question in **Part-A**
3. Answer any **FOUR** Questions from **Part-B**

~~~~~~

PART –A

1.	a)	Draw the Layered structure of Operating system.	[2M]
	b)	When a process creates a new process, what is shared between parent process and child process?	[2M]
	c)	List the disadvantages of single contiguous memory allocation.	[2M]
	d)	What is Counting semaphore.	[3M]
	e)	Write about Master File Directory in two-level directory structure.	[2M]
	f)	What are Synchronous and Asynchronous interrupts in LINUX.	[3M]

PART -B

2.	a)	Discuss various challenges and issues that are to be considered while designing an operating system.	[5M]
	b)	Why Real time operating systems are needed? Give some examples.	[4M]
	c)	Write the difference between the function and system call. Briefly explain the six major categories of system calls.	[5M]
3.	a)	Explain in detail the two popular inter-process communication mechanisms.	[7M]
	b)	What are the criteria for evaluating the CPU scheduling algorithms? Why do we need it?	[7M]
4.	a)	Discuss various issued related to the allocation of frames to processes.	[7M]
	b)	Consider the following page reference string: 1,2,4,7,3,5,6,3,6,1,4,2,3,6,5,2 How many page faults would occur for the LRU page replacement algorithm, assuming four frames and all frames are initially empty.	[7M]
5.	a)	How does the signal() operation associated with monitors differ from the corresponding operation defined for semaphores.	[7M]
	b)	Explain the various ways of aborting a process in order to eliminate deadlocks.	[7M]
6.	a)	Explain the Indexed allocation of disk space.	[7M]
	b)	Compare the SCAN and C-SCAN disk scheduling algorithms with an example.	[7M]
7.	a)	Explain about the key Components of a LINUX system.	[7M]
	b)	What is Android? Is Android Open-Source? Discuss the features of Android.	[7M]

*****
~~~~~~

K. S. Institute of Technology, Bangalore – 109

Department of Computer Science and Engineering

COURSE END SURVEY

ACADEMIC YEAR 2020-2021 (EVEN SEMESTER)

Subject:Operating Systems **Faculty Name: Mrs. Vaneeta M**

Sem and Sec:IV

1. Rate your understanding about various types of Operating Systems, its need and services.
2. Grade your knowledge about applying suitable techniques for process scheduling, synchronization and thread management.
3. To what extent you are able to make use of different methods for preventing or avoiding deadlock and managing memory efficiently.
4. Rate you understanding about the benefits of virtual memory; explore file system and directory structures.
5. Grade your knowledge with different disk management schemes and realize the concepts of Operating System with case studies

SL. No	Name of student	USN	Q1	Q2	Q3	Q4	Q5
1	Bharath R	1KS18CS011	Excellent	Excellent	Excellent	Excellent	Excellent
2	Aakriti	1KS19CS001	Excellent	Excellent	Good	Excellent	Good
3	Abhishek B	1KS19CS002	Good	Good	Good	Good	Good
4	Abhishek Yadav	1KS19CS003	Excellent	Excellent	Excellent	Excellent	Excellent
5	Adith Karthik Raju	1KS19CS004	Good	Good	Good	Good	Good
6	AKASH.A.S	1KS19CS006	Excellent	Excellent	Good	Excellent	Good
7	Aman kushwaha	1KS19CS007	Excellent	Excellent	Excellent	Excellent	Excellent
8	Amit K B	1KS19CS009	Excellent	Good	Excellent	Excellent	Excellent
9	AMOGHA.HS	1KS19CS010	Excellent	Excellent	Excellent	Excellent	Excellent
10	Amrutha K H	1KS19CS011	Good	Good	Good	Good	Good
11	Anagha A Hebbar	1KS19CS012	Excellent	Good	Excellent	Excellent	Excellent
12	Anusha.B	1KS19CS014	Good	Good	Good	Excellent	Excellent
13	Aqsa Aqeel	1KS19CS015	Good	Good	Excellent	Excellent	Good
14	Ashika HN	1ks19cs016	Good	Excellent	Good	Excellent	Good
15	Bhoomika AM	1KS19CS017	Good	Good	Excellent	Good	Good
16	BHOOMIKA K	1KS19CS018	Good	Good	Excellent	Good	Good
17	Bhumika M	1KS19CS019	Good	Good	Satisfactory	Good	Good
18	C N Shreyas	1KS19CS020	Good	Good	Good	Good	Good
19	Chaitanya Shivaraju	1KS19CS021	Good	Good	Excellent	Excellent	Good
20	Deepa.G	1KS19CS024	Excellent	Excellent	Excellent	Excellent	Excellent
21	Deepthi. NK	1KS19CS025	Excellent	Good	Excellent	Excellent	Good
22	Devi prasad	1KS19CS026	Good	Excellent	Good	Excellent	Good
23	Dheemanth G	1KS19CS028	Excellent	Excellent	Good	Excellent	Excellent
24	G Preritha	1KS19CS030	Good	Good	Good	Good	Good
25	Gagan Reddy S	1KS19CS031	Satisfactory	Satisfactory	Satisfactory	Satisfactory	Satisfactory
26	Gagandeep K	1KS19CS032	Good	Good	Good	Good	Good
27	Geethanjali BK prasad	1KS19CS033	Excellent	Good	Good	Good	Good
28	Harshitha	1KS19CS035	Excellent	Excellent	Good	Good	Good
29	INDRAJITH H M	1KS19CS036	Excellent	Excellent	Excellent	Excellent	Excellent
30	K KISHAN	1KS19CS038	Excellent	Excellent	Excellent	Good	Excellent
31	K.R.VAGEESH	1KS19CS039	Excellent	Excellent	Excellent	Good	Good

SL. No	Name of student	USN	Q1	Q2	Q3	Q4	Q5
32	Karthik S Morajkar	1KS19CS041	Good	Excellent	Good	Excellent	Excellent
33	KavyaShree S L	1KS19CS042	Excellent	Excellent	Excellent	Excellent	Good
34	Keerthan Gowda S	1KS19CS043	Excellent	Excellent	Excellent	Excellent	Excellent
35	Kothapalli Sreeja	1KS19CS044	Good	Good	Good	Good	Good
36	KOTTALA.SAIVENKATASUCHITH	1KS19CS045	Good	Good	Good	Good	Good
37	Krishna k r	1KS19CS046	Excellent	Excellent	Excellent	Excellent	Excellent
38	KUMAR S	1KS19CS047	Good	Good	Good	Good	Good
39	Lisha C	1KS19CS049	Good	Good	Good	Good	Good
40	Mahak shree	1KS19CS050	Good	Good	Excellent	Good	Good
41	Mallipalli Spurthi Reddy	1KS19CS051	Good	Good	Good	Good	Good
42	Manasa G L	1KS19CS052	Excellent	Excellent	Excellent	Excellent	Excellent
43	Mohammed Noor Aman	1KS19CS053	Excellent	Good	Good	Excellent	Excellent
44	Mukesh Kumar	1KS19CS054	Satisfactory	Satisfactory	Satisfactory	Satisfactory	Satisfactory
45	Mythreyi U	1KS19CS055	Excellent	Excellent	Excellent	Excellent	Good
46	N ASHOK	1KS19CS056	Excellent	Good	Good	Good	Excellent
47	N Bhavya	1KS19CS057	Good	Good	Good	Good	Good
48	NP Shashanka Rao	1KS19CS058	Good	Good	Good	Good	Good
49	Vrathika Billava	1KS19CS116	Good	Good	Excellent	Excellent	Good
50	Akif	1ks20cs400	Excellent	Excellent	Excellent	Excellent	Excellent
51	KEERTHI KUMAR V	1KS20CS402	Excellent	Excellent	Excellent	Excellent	Excellent
52	Pranav Chandran P	1KS20CS404	Excellent	Excellent	Excellent	Excellent	Excellent
53	AJAY S KALBURGI	1KS19CS005	Excellent	Excellent	Good	Good	Good
54	CHANDAN B V	1KS19CS022	Excellent	Excellent	Good	Good	Good
55	DEEKSHA NAIDU R	1KS19CS023	Excellent	Excellent	Excellent	Excellent	Excellent
56	DINESH M	1KS19CS029	Excellent	Excellent	Good	Good	Good
57	GULSHAN KUMAR S	1KS19CS034	Excellent	Excellent	Good	Good	Good
58	KALYAN CHOWDHARY B	1KS19CS040	Good	Excellent	Good	Excellent	Excellent
59	LIKITH G	1KS19CS048	Excellent	Excellent	Good	Good	Good
60	G.Sarayu Chowdary	1KS18CS021	Good	Good	Good	Good	Good
61	Prajwal N	1KS18CS068	Excellent	Excellent	Excellent	Excellent	Excellent
62	Naveen K M	1KS19CS059	Excellent	Excellent	Excellent	Excellent	Excellent
63	Nethra R	1ks19cs061	Excellent	Excellent	Excellent	Excellent	Excellent
64	Nikhil Chowdary V R	1KS19CS062	Excellent	Excellent	Excellent	Excellent	Excellent
65	Nithya s n	1ks19cs063	Good	Good	Good	Good	Good
66	Pavan B Indresh	1KS19CS064	Good	Good	Good	Excellent	Good
67	Pavan L	1KS19CS065	Excellent	Excellent	Excellent	Excellent	Excellent
68	Pavithra sp	1ks19cs066	Excellent	Excellent	Excellent	Excellent	Excellent
69	POOJA J	1KS19CS067	Excellent	Excellent	Excellent	Excellent	Excellent
70	Sai Priyanka Pothuru	1KS19CS068	Excellent	Excellent	Excellent	Excellent	Excellent
71	Prajwal G	1KS19CS069	Excellent	Good	Good	Good	Good
72	PREETHI K P	1KS19CS071	Good	Good	Good	Good	Good
73	Priya E	1KS19CS072	Good	Satisfactory	Good	Good	Good
74	Priyanka R	1KS19CS073	Excellent	Excellent	Excellent	Excellent	Good
75	Puvan Kumar V	1KS19CS074	Excellent	Excellent	Excellent	Excellent	Excellent
76	Rv Yashvanth	1KS19CS075	Good	Excellent	Excellent	Excellent	Excellent
77	Raj kumar	1ks19cs076	Excellent	Excellent	Excellent	Excellent	Excellent
78	Raksha T M	1KS19CS078	Good	Good	Good	Good	Good
79	Ranjitha DV	1KS19CS079	Excellent	Excellent	Excellent	Good	Good
80	Sahana.S	1KS19CS080	Excellent	Excellent	Excellent	Excellent	Excellent

Sl. No	Name of student	USN	Q1	Q2	Q3	Q4	Q5
81	Sai shirisha S B	1KS19CS081	Good	Good	Good	Good	Good
82	Sanjeev Mysore	1KS19CS083	Good	Good	Good	Good	Good
83	Sankalp Kesti	1KS19CS084	Excellent	Excellent	Excellent	Excellent	Excellent
84	Shanvi	1KS19CS086	Good	Good	Good	Good	Good
85	Shreesha	1ks19cs088	Excellent	Excellent	Excellent	Satisfactory	Satisfactory
86	Shriharsha	1ks19cs090	Good	Excellent	Excellent	Good	Good
87	Sindhu S	1KS19CS091	Excellent	Excellent	Excellent	Excellent	Excellent
88	Smrithi Shekar	1KS19CS092	Excellent	Excellent	Excellent	Excellent	Excellent
89	Srujan K	1KS19CS093	Excellent	Excellent	Excellent	Excellent	Excellent
90	Srusti S Gowda	1KS19CS094	Excellent	Excellent	Excellent	Good	Excellent
91	Suhas S	1ks19cs096	Excellent	Excellent	Excellent	Excellent	Excellent
92	Sushmitha K	1KS19CS097	Excellent	Good	Good	Excellent	Good
93	Syed Zain	1KS19CS099	Good	Good	Good	Good	Good
94	Tanushree.R	1ks19cs101	Good	Good	Good	Good	Good
95	Tejas.n	1KS19CS102	Excellent	Excellent	Excellent	Excellent	Excellent
96	Tejas P	1KS19CS103	Good	Good	Good	Good	Good
97	Tejaswini Nayaka S	1KS19CS104	Good	Good	Good	Good	Good
98	Thanushree S	1KS19CS105	Excellent	Excellent	Excellent	Excellent	Excellent
99	THIRUMAL.R	1KS19CS106	Excellent	Good	Excellent	Excellent	Good
100	THRIVENI U	1KS19CS107	Good	Good	Good	Good	Good
101	Udhay kumar	1KS19CS108	Excellent	Excellent	Excellent	Excellent	Excellent
102	VAISHNAVI.G	1KS19CS109	Good	Good	Good	Good	Good
103	Varsha Bai R	1KS19CS110	Excellent	Excellent	Excellent	Excellent	Excellent
104	Varun Kambali	1KS19CS111	Excellent	Excellent	Good	Good	Good
105	Vishal Gupta	1KS19CS112	Good	Good	Good	Good	Good
106	Akshay R	1KS19CS115	Excellent	Excellent	Excellent	Excellent	Good
107	Anusha. A.R	1ks20cs401	Good	Good	Excellent	Good	Good
108	Nithin S	1KS20CS403	Good	Excellent	Excellent	Good	Good
109	TALLURU MAURYA	1KS19CS100	Excellent	Excellent	Excellent	Excellent	Excellent
110	SATISH V	1KS19CS085	Excellent	Excellent	Excellent	Excellent	Excellent
111	YASHWANTH S R	1KS19CS113	Good	Good	Good	Good	Good
112	PRAJWAL KULKARNI	1KS19CS070	Excellent	Excellent	Excellent	Excellent	Excellent
113	SHIVANI G K	1KS19CS087	Excellent	Excellent	Excellent	Excellent	Excellent
114	SHREYA R KIRAN	1KS19CS089	Excellent	Excellent	Excellent	Excellent	Excellent
115	SANGHARSH KUMAR RAI	1KS19CS082	Excellent	Excellent	Excellent	Excellent	Excellent
116	RAKESH M J	1KS19CS077	Excellent	Excellent	Excellent	Excellent	Excellent
117	SWETHA M KULKARNI	1KS19CS098	Good	Good	Good	Good	Good
118	NEELAMMA SALI	1KS19CS060	Excellent	Good	Good	Excellent	Excellent

Signature of Faculty Incharge

K.S. INSTITUTE OF TECHNOLOGY,
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIRECT ATTAINMENT COURSE END SURVEY

YEAR / SEMESTER	II/IV
COURSE TITLE	Operating Systems
COURSE CODE	18CS43
ACADEMIC YEAR	2020-21

	Q1	Q2	Q3	Q4	Q5	
EXCELLENT	69	66	63	64	52	
GOOD	47	49	52	51	63	
SATISFACTORY	2	3	3	3	3	98%
STUDENTS RESPONSE(GOOD & ABOVE)	98%	97%	97%	97%	97%	

STAFF SIGNATURE

HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K S Institute of Technology, Bangalore-109.

VTU RESULT JULY/ AUGUST EXAM - 2020 -21 (EVEN SEMESTER)

College Code: KS

IV Sem -Computer Science and Engineering

EX- External (Semester End Exam)

IA - Internal Assessment

Sl.No	USN	Name of the Student	18CS43 - OPERATING SYSTEMS		
			IA	EX	TOT
		CREDIT POINT		3	
1	1KS19CS001	AAKRITI	45	35	**80**
2	1KS19CS002	ABHISHEK B	47	24	**71**
3	1KS19CS003	ABHISHEK YADAV	43	23	**66**
4	1KS19CS004	ADITH KARTHIK RAJU	47	23	**70**
5	1KS19CS005	AJAY S KALBURGI	47	29	**76**
6	1KS19CS006	AKASH A S	47	27	**74**
7	1KS19CS007	AMAN KUSHWAHA	47	29	**76**
8	1KS19CS009	AMIT K B	44	29	**73**
9	1KS19CS010	AMOGHA H S	47	24	**71**
10	1KS19CS011	AMRUTHA K H	48	33	**81**
11	1KS19CS012	ANAGHA A HEBBAR	45	31	**76**
12	1KS19CS014	ANUSHA B	48	24	**72**
13	1KS19CS015	AQSA AQEEL	48	18	**66**
14	1KS19CS016	ASHIKA H N	49	30	**79**
15	1KS19CS017	BHOOMIKA A M	50	35	**85**
16	1KS19CS018	BHOOMIKA K	48	37	**85**
17	1KS19CS019	BHUMIKA M	49	36	**85**
18	1KS19CS020	C N SHREYAS	48	31	**79**
19	1KS19CS021	CHAITANYA SHIVARAJU	50	35	**85**
20	1KS19CS022	CHANDAN B V	42	20	**62**
21	1KS19CS023	DEEKSHA NAIDU R	44	23	**67**
22	1KS19CS024	DEEPA G	47	33	**80**
23	1KS19CS025	DEEPTHI.N.K	47	35	**82**
24	1KS19CS026	DEVI PRASAD N	48	28	**76**
25	1KS19CS028	DHEEMANTH G	48	23	**71**
26	1KS19CS029	DINESH M	45	21	**66**
27	1KS19CS030	G PRERITHA	47	24	**71**
28	1KS19CS031	GAGAN REDDY S	40	18	**58**
29	1KS19CS032	GAGANDEEP K	45	18	**63**
30	1KS19CS033	GEETHANJALI B K PRASAD	48	33	**81**
31	1KS19CS034	GULSHAN KUMAR S	47	29	**76**
32	1KS19CS035	HARSHITHA J	49	39	**88**
33	1KS19CS036	INDRAJITH H M	49	33	**82**
34	1KS19CS038	K KISHAN	49	26	**75**
35	1KS19CS039	K R VAGEESH	47	38	**85**
36	1KS19CS040	KALYAN CHOWDHARY B	44	22	**66**
37	1KS19CS041	KARTHIK S MORAJKAR	49	25	**74**
38	1KS19CS042	KAVYASHREE.S.L	49	35	**84**
39	1KS19CS043	KEERTHAN GOWDA S	48	36	**84**
40	1KS19CS044	KOTHAPALLI SREEJA	47	31	**78**
41	1KS19CS045	KOTTALA SAIVENKATASUCHITH	40	23	**63**
42	1KS19CS046	KRISHNA K R	45	21	**66**
43	1KS19CS047	KUMAR S	39	11	**50**

44	1KS19CS048	LIKITH G	39	24	63
45	1KS19CS049	LISHA C	48	29	77
46	1KS19CS050	MAHAK SHREE	48	40	88
47	1KS19CS051	MALLIPALLI SPURTHI REDDY	49	35	84
48	1KS19CS052	MANASA G L	49	26	75
49	1KS19CS053	MOHAMMED NOOR AMAN	45	20	65
50	1KS19CS054	MUKESH KUMAR	44	25	69
51	1KS19CS055	MYTHREYI U	49	35	84
52	1KS19CS056	N ASHOK	48	27	75
53	1KS19CS057	N BHAVYA	48	29	77
54	1KS19CS058	N P SHASHANKA RAO	47	25	72
55	1KS19CS059	NAVEEN K M	48	32	80
56	1KS19CS060	NEELAMMA SALI	48	32	80
57	1KS19CS061	NETHRA R	45	31	76
58	1KS19CS062	NIKHIL CHOWDARY V R	40	29	69
59	1KS19CS063	NITHYA S N	48	32	80
60	1KS19CS064	PAVAN B INDRESH	43	24	67
61	1KS19CS065	PAVAN L	45	20	65
62	1KS19CS066	PAVITHRA S P	47	27	74
63	1KS19CS067	POOJA J	49	35	84
64	1KS19CS068	POTHURU SAI PRIYANKA	44	25	69
65	1KS19CS069	PRAJWAL G	39	24	63
66	1KS19CS070	PRAJWAL KULKARNI	47	29	76
67	1KS19CS071	PREETHI K P	50	40	90
68	1KS19CS072	PRIYA E	49	37	86
69	1KS19CS073	PRIYANKA R	47	26	73
70	1KS19CS074	PUVAN KUMAR V	47	31	78
71	1KS19CS075	R V YASHVANTH	48	27	75
72	1KS19CS076	RAJ KUMAR	39	18	57
73	1KS19CS077	RAKESH M J	48	27	75
74	1KS19CS078	RAKSHA T M	47	26	73
75	1KS19CS079	RANJITHA D V	48	23	71
76	1KS19CS080	SAHANA S	48	21	69
77	1KS19CS081	SAI SHIRISHA S B	45	26	71
78	1KS19CS082	SANGHARSH KUMAR RAI	40	21	61
79	1KS19CS083	SANJEEV MYSORE	47	27	74
80	1KS19CS084	SANKALP KESTI	47	30	77
81	1KS19CS085	SATISH V	50	35	85
82	1KS19CS086	SHANVI B P	48	33	81
83	1KS19CS087	SHIVANI G K	49	31	80
84	1KS19CS088	SHREESHA S	48	42	90
85	1KS19CS089	SHREYA R KIRAN	49	28	77
86	1KS19CS090	SHRI HARSHA S	48	23	71
87	1KS19CS091	SINDHU S	49	24	73
88	1KS19CS092	SMRITHI SHEKAR	49	22	71
89	1KS19CS093	SRUJAN K	48	25	73
90	1KS19CS094	SRUSTI S GOWDA	48	37	85
91	1KS19CS096	SUHAS S	45	21	66
92	1KS19CS097	SUSHMITHA K	49	28	77
93	1KS19CS098	SWETHA M KULKARNI	48	32	80
94	1KS19CS099	SYED ZAINUL ABIDIN	44	13	57
95	1KS19CS100	TALLURU MAURYA	43	18	61
96	1KS19CS101	TANUSHREE R	45	25	70
97	1KS19CS102	TEJAS N	47	23	70
98	1KS19CS103	TEJAS P	42	22	64

99	1KS19CS104	TEJASWINI NAYAKA S	48	34	**82**
100	1KS19CS105	THANUSHREE S	45	19	**64**
101	1KS19CS106	THIRUMAL R	43	18	**61**
102	1KS19CS107	THRIVENI U	49	33	**82**
103	1KS19CS108	UDHAY KUMAR G	39	10	**49**
104	1KS19CS109	VAISHNAVI G	47	22	**69**
105	1KS19CS110	VARSHA BAI R	48	39	**87**
106	1KS19CS111	VARUN KAMBALI	44	15	**59**
107	1KS19CS112	VISHAL GUPTA	44	28	**72**
108	1KS19CS113	YASHWANTH S R	49	40	**89**
109	1KS19CS115	AKSHAY R	50	36	**86**
110	1KS19CS116	VRATHIKA BILLAVA	39	35	**74**
111	1KS18CS011	BHARATH R	39	21	**60**
112	1KS18CS021	G. SARAYU CHOWDARY	44	16	**60**
113	1KS18CS068	PRAJWAL N	44	22	**66**
114	1KS20CS400	AKIF DELVI	47	22	**69**
115	1KS20CS401	ANUSHA A R	45	25	**70**
116	1KS20CS402	KEERTHI KUMAR V	40	25	**65**
117	1KS20CS403	NITHIN S	47	23	**70**
118	1KS20CS404	PRANAV CHANDRAN P	40	19	**59**
		NUMBER OF PASS	00	**118**	**118**
		AVERAGE MARKS	**00**	**27**	**73**
		FAILURES IN SUBJECTS	**00**		
		ABSENTS IN SUBJECTS		**00**	

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K S INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	II / IV
COURSE TITLE	OPERATING SYSTEMS
COURSE CODE	18CS43
ACADEMIC YEAR	2020-2021

Sl. No	USN	Name of the Student	18CS43															
			IA1		A1		IA2			A2			IA3		A3		SEE	SEE
			CO1	CO2	CO1	CO2	CO2	CO3	CO4	CO2	CO3	CO4	CO4	CO5	CO4	CO5		
			18	**12**	**6**	**4**	**6**	**18**	**6**	**2**	**6**	**2**	**12**	**18**	**4**	**6**	**50**	**60**
1	1KS19CS001	AAKRITI	14	6	6	4	6	18	5	2	6	2	12	15	4	6	35	42
2	1KS19CS002	ABHISHEK B	16	11	6	4	6	15	5	2	6	2	12	15	4	6	24	29
3	1KS19CS003	ABHISHEK YADAV	12	12	6	4	6	15	5	2	6	2	6	15	4	6	23	28
4	1KS19CS004	ADITH KARTHIK RAJU	16	10	6	4	6	15	5	2	6	2	12	15	4	6	23	28
5	1KS19CS005	AJAY S KALBURGI	17	11	6	4	6	15	5	2	6	2	12	15	4	6	29	35
6	1KS19CS006	AKASH A S	15	12	6	4	6	15	5	2	6	2	11	15	4	6	27	33
7	1KS19CS007	AMAN KUSHWAHA	16	10	6	4	6	15	5	2	6	2	12	15	4	6	29	35
8	1KS19CS009	AMIT K B	12	7	6	4	6	15	5	2	6	2	12	18	4	6	29	35
9	1KS19CS010	AMOGHA H S	16	11	6	4	6	15	5	2	6	2	11	15	4	6	24	29
10	1KS19CS011	AMRUTHA K H	17	12	6	4	6	15	5	2	6	2	12	17	4	6	33	40
11	1KS19CS012	ANAGHA A HEBBAR	16	9	6	4	6	16	5	2	6	2	11	15	4	6	31	38
12	1KS19CS014	ANUSHA B	14	11	6	4	6	18	5	2	6	2	12	18	4	6	24	29
13	1KS19CS015	AQSA AQEEL	17	11	6	4	5	15	5	2	6	2	12	18	4	6	18	22
14	1KS19CS016	ASHIKA H N	18	12	6	4	6	15	6	2	6	2	11	18	4	6	30	36
15	1KS19CS017	BHOOMIKA A M	18	12	6	4	6	18	5	2	6	2	12	18	4	6	35	42
16	1KS19CS018	BHOOMIKA K	17	11	6	4	5	17	3	2	6	2	12	18	4	6	37	45
17	1KS19CS019	BHUMIKA M	15	12	6	4	6	18	5	2	6	2	12	18	4	6	36	44
18	1KS19CS020	C N SHREYAS	15	12	6	4	6	18	5	2	6	2	12	15	4	6	31	38
19	1KS19CS021	CHAITANYA SHIVARAJU	17	12	6	4	6	18	5	2	6	2	12	18	4	6	35	42
20	1KS19CS022	CHANDAN B V	15	4	6	4	5	15	5	2	6	2	11	12	4	6	20	24
21	1KS19CS023	DEEKSHA NAIDU R	16	12	6	4	6	18	5	2	6	2	12	15	NA	6	23	28
22	1KS19CS024	DEEPA G	15	12	6	4	6	15	5	2	6	2	12	15	4	6	33	40
23	1KS19CS025	DEEPTHI.N.K	13	12	6	4	6	16	5	2	6	2	12	15	4	6	35	42
24	1KS19CS026	DEVI PRASAD N	16	11	6	4	6	17	5	2	6	2	11	18	4	6	28	34
25	1KS19CS028	DHEEMANTH G	16	12	6	4	6	16	5	2	6	2	12	15	4	6	23	28
26	1KS19CS029	DINESH M	13	10	6	4	6	15	5	2	6	2	12	15	4	6	21	26
27	1KS19CS030	G PRERITHA	14	12	6	4	6	15	5	2	6	2	12	15	4	6	24	29
28	1KS19CS031	GAGAN REDDY S	15	4	6	4	6	13	5	2	6	2	11	12	4	6	18	22

29	1KS19CS032	GAGANDEEP K	16	10	6	4	6	14	5	2	6	2	11	15	4	6	18	22
30	1KS19CS033	GEETHANJALI B K PRASAD	15	11	6	4	6	16	5	2	6	2	12	18	4	6	33	40
31	1KS19CS034	GULSHAN KUMAR S	16	11	6	4	6	15	5	2	6	2	12	15	4	6	29	35
32	1KS19CS035	HARSHITHA J	17	12	6	4	6	15	5	2	6	2	12	18	4	6	39	47
33	1KS19CS036	INDRAJITH H M	15	12	6	4	6	17	5	2	6	2	12	18	4	6	33	40
34	1KS19CS038	K KISHAN	14	12	6	4	6	18	5	2	5	1	12	18	4	6	26	32
35	1KS19CS039	K R VAGEESH	15	12	6	4	6	18	5	2	6	2	12	12	4	6	38	46
36	1KS19CS040	KALYAN CHOWDHARY B	12	8	6	4	6	15	5	2	6	2	12	15	4	6	22	27
37	1KS19CS041	KARTHIK S MORAJKAR	18	12	6	4	6	15	5	2	6	2	12	18	4	6	25	30
38	1KS19CS042	KAVYASHREE.S.L	17	11	6	4	6	18	5	2	6	2	12	16	4	6	35	42
39	1KS19CS043	KEERTHAN GOWDA S	14	11	6	4	6	16	5	2	6	2	12	18	4	6	36	44
40	1KS19CS044	KOTHAPALLI SREEJA	10	11	6	4	6	16	6	2	6	2	12	18	4	6	31	38
41	1KS19CS045	KOTTALA SAIVENKATASUCHITH	14	12	6	4	6	15	5	NA	NA	NA	10	12	4	6	23	28
42	1KS19CS046	KRISHNA K R	14	12	6	4	6	15	5	2	6	2	12	12	4	6	21	26
43	1KS19CS047	KUMAR S	13	3	6	4	6	11	5	2	6	2	12	12	4	6	11	14
44	1KS19CS048	LIKITH G	11	3	6	4	6	11	4	2	6	2	12	12	4	6	24	29
45	1KS19CS049	LISHA C	16	12	6	4	6	16	5	2	6	2	12	15	4	6	29	35
46	1KS19CS050	MAHAK SHREE	17	12	6	4	6	17	5	2	6	2	11	15	4	6	40	48
47	1KS19CS051	MALLIPALLI SPURTHI REDDY	16	12	6	4	6	18	5	2	6	2	12	18	4	6	35	42
48	1KS19CS052	MANASA G L	16	12	6	4	6	18	5	2	6	2	12	18	4	6	26	32
49	1KS19CS053	MOHAMMED NOOR AMAN	16	8	6	4	6	16	5	2	6	2	11	16	4	6	20	24
50	1KS19CS054	MUKESH KUMAR	14	7	6	4	6	15	5	2	6	2	12	15	4	6	25	30
51	1KS19CS055	MYTHREYI U	17	11	6	4	6	18	5	2	6	2	12	18	4	6	35	42
52	1KS19CS056	N ASHOK	16	10	6	4	6	17	5	2	6	2	12	18	4	6	27	33
53	1KS19CS057	N BHAVYA	16	11	6	4	6	15	6	2	6	2	12	18	4	6	29	35
54	1KS19CS058	N P SHASHANKA RAO	17	9	6	4	6	16	5	2	6	2	12	14	4	6	25	30
55	1KS19CS059	NAVEEN K M	17	12	6	4	3	17	5	2	6	2	12	16	4	6	32	39
56	1KS19CS060	NEELAMMA SALI	17	12	6	4	6	17	5	2	6	2	12	15	4	6	32	39
57	1KS19CS061	NETHRA R	15	11	6	4	6	16	5	2	6	2	11	18	4	6	31	38
58	1KS19CS062	NIKHIL CHOWDARY V R	16	3	6	4	6	15	5	2	6	2	6	15	4	6	29	35
59	1KS19CS063	NITHYA S N	17	11	6	4	6	15	5	2	6	2	12	18	4	6	32	39
60	1KS19CS064	PAVAN B INDRESH	16	10	6	4	6	15	5	2	6	2	5	15	4	6	24	29
61	1KS19CS065	PAVAN L	15	10	6	4	5	17	5	2	6	2	12	14	4	6	20	24
62	1KS19CS066	PAVITHRA S P	16	10	6	4	6	15	5	2	6	2	12	15	4	6	27	33
63	1KS19CS067	POOJA J	16	5	6	4	6	16	5	2	6	2	12	15	4	6	35	42
64	1KS19CS068	POTHURU SAI PRIYANKA	16	6	6	4	6	17	5	2	6	2	12	12	4	6	25	30
65	1KS19CS069	PRAJWAL G	10	10	6	4	6	11	5	2	6	2	7	12	4	6	24	29
66	1KS19CS070	PRAJWAL KULKARNI	18	6	6	4	6	15	5	2	6	2	12	18	4	6	29	35
67	1KS19CS071	PREETHI K P	18	12	6	4	6	18	5	2	6	2	12	18	4	6	40	48
68	1KS19CS072	PRIYA E	16	11	6	4	6	17	5	2	6	2	12	18	4	6	37	45
69	1KS19CS073	PRIYANKA R	15	12	6	4	6	15	5	2	6	2	12	15	4	6	26	32
70	1KS19CS074	PUVAN KUMAR V	16	10	6	4	6	16	5	2	6	2	12	15	4	6	31	38
71	1KS19CS075	R V YASHVANTH	17	12	6	4	6	17	5	2	6	2	12	15	4	6	27	33
72	1KS19CS076	RAJ KUMAR	10	6	6	4	6	14	5	2	6	2	12	15	4	6	18	22

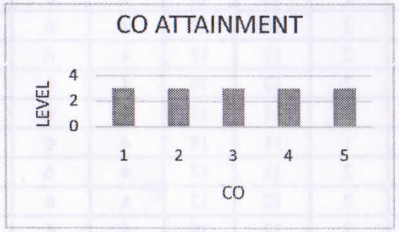
73	1KS19CS077	RAKESH M J	13	12	6	4	6	17	5	2	6	2	12	17	4	6	27	33
74	1KS19CS078	RAKSHA T M	16	6	6	4	6	17	5	2	6	2	12	18	4	6	26	32
75	1KS19CS079	RANJITHA D V	16	12	6	4	6	15	5	2	6	2	11	18	4	6	23	28
76	1KS19CS080	SAHANA S	18	12	6	4	6	16	5	2	6	2	12	15	4	6	21	26
77	1KS19CS081	SAI SHIRISHA S B	16	12	6	4	6	15	5	2	6	2	12	12	4	6	26	32
78	1KS19CS082	SANGHARSH KUMAR RAI	16	6	6	4	6	15	5	NA	NA	NA	12	13	4	6	21	26
79	1KS19CS083	SANJEEV MYSORE	18	8	6	4	6	16	5	2	6	2	12	15	4	6	27	33
80	1KS19CS084	SANKALP KESTI	16	11	6	4	6	15	5	2	6	2	12	15	4	6	30	36
81	1KS19CS085	SATISH V	18	12	6	4	6	17	5	2	6	2	12	18	4	6	35	42
82	1KS19CS086	SHANVI B P	14	12	6	4	6	16	5	2	6	2	11	18	4	6	33	40
83	1KS19CS087	SHIVANI G K	18	12	6	4	6	16	5	2	6	2	11	18	4	6	31	38
84	1KS19CS088	SHREESHA S	15	10	6	4	6	16	5	2	6	2	12	18	4	6	42	51
85	1KS19CS089	SHREYA R KIRAN	17	12	6	4	6	18	5	2	6	2	11	18	4	6	28	34
86	1KS19CS090	SHRI HARSHA S	15	12	6	4	6	17	5	2	6	2	12	15	4	6	23	28
87	1KS19CS091	SINDHU S	16	12	6	4	6	17	5	2	6	2	11	18	4	6	24	29
88	1KS19CS092	SMRITHI SHEKAR	16	12	6	4	6	17	5	2	6	2	11	18	4	6	22	27
89	1KS19CS093	SRUJAN K	16	11	6	4	6	16	5	2	6	2	11	18	4	6	25	30
90	1KS19CS094	SRUSTI S GOWDA	16	11	6	4	6	15	6	2	6	2	12	18	4	6	37	45
91	1KS19CS096	SUHAS S	18	6	6	4	6	15	5	2	6	2	12	14	4	6	21	26
92	1KS19CS097	SUSHMITHA K	16	12	6	4	6	17	5	2	6	2	11	18	4	6	28	34
93	1KS19CS098	SWETHA M KULKARNI	17	12	6	4	6	18	5	2	6	2	11	15	4	6	32	39
94	1KS19CS099	SYED ZAINUL ABIDIN	16	10	6	4	6	15	5	NA	6	2	10	12	4	6	13	16
95	1KS19CS100	TALLURU MAURYA	16	11	6	4	6	15	5	2	6	2	11	15	4	6	18	22
96	1KS19CS101	TANUSHREE R	16	12	6	4	5	16	5	2	6	2	12	12	4	6	25	30
97	1KS19CS102	TEJAS N	16	12	6	4	6	16	5	2	6	2	11	15	4	6	23	28
98	1KS19CS103	TEJAS P	18	6	6	NA	5	12	5	1	5	1	11	18	4	6	22	27
99	1KS19CS104	TEJASWINI NAYAKA S	16	12	6	4	6	17	5	2	6	2	11	17	4	6	34	41
100	1KS19CS105	THANUSHREE S	17	10	6	4	6	15	5	2	6	2	12	12	4	6	19	23
101	1KS19CS106	THIRUMAL R	14	3	6	NA	6	16	5	2	6	2	12	18	4	6	18	22
102	1KS19CS107	THRIVENI U	17	12	6	4	6	17	5	2	6	2	11	18	4	6	33	40
103	1KS19CS108	UDHAY KUMAR G	15	5	6	4	6	11	5	2	6	2	12	6	4	6	10	12
104	1KS19CS109	VAISHNAVI G	17	11	6	4	6	15	5	2	6	2	12	15	4	6	22	27
105	1KS19CS110	VARSHA BAI R	18	8	6	4	6	17	5	2	6	2	12	18	4	6	39	47
106	1KS19CS111	VARUN KAMBALI	17	10	6	4	4	15	5	2	6	2	12	12	4	6	15	18
107	1KS19CS112	VISHAL GUPTA	12	10	6	4	6	15	5	2	6	2	12	15	4	6	28	34
108	1KS19CS113	YASHWANTH S R	17	12	6	4	6	17	5	2	6	2	12	18	4	6	40	48
109	1KS19CS115	AKSHAY R	18	12	6	4	6	17	5	2	6	2	12	18	4	6	36	44
110	1KS19CS116	VRATHIKA BILLAVA	8	11	6	4	6	13	5	2	6	2	12	12	2	NA	35	42
111	1KS20CS400	AKIF DELVI	15	11	6	4	2	17	5	2	6	2	11	18	4	6	22	27
112	1KS20CS401	ANUSHA A R	13	11	6	4	6	15	5	2	6	2	12	14	4	6	25	30
113	1KS20CS402	KEERTHI KUMAR V	11	5	6	4	6	13	5	2	6	2	12	12	4	6	25	30
114	1KS20CS403	NITHIN S	15	11	6	4	6	15	5	2	6	2	12	17	4	6	23	28
115	1KS20CS404	PRANAV CHANDRAN P	11	5	6	4	6	10	5	2	6	2	10	18	4	6	19	23
60% of Maximum marks (X)			11	07	04	02	04	11	04	01	04	01	07	11	02	04	30	36

No. of students above X	111	95	115	113	113	114	114	111	113	111	111	114	113	114	42	42
Total number of students (Y)	115	115	115	113	115	115	115	112	113	113	115	115	114	114	115	115
CO Percentage	96.52	82.61	100.00	100.00	98.26	99.13	99.13	99.11	100.00	98.23	96.52	99.13	99.12	100.00	36.52	36.52
	CO1	CO2	CO1	CO2	CO2	CO3	CO4	CO2	CO3	CO4	CO4	CO5	CO4	CO5	SEE	SEE

CO	CIE	SEE	DIRECT ATTAINMENT	Level	COURSE EXIT SURVEY	LEVEL	ATTAINMENT
CO1	98.26	36.52	67.39	3.00	98.00	3.00	3
CO2	94.99	36.52	65.76	3.00	98.00	3.00	3
CO3	99.57	36.52	68.04	3.00	98.00	3.00	3
CO4	98.25	36.52	67.39	3.00	98.00	3.00	3
CO5	99.57	36.52	68.04	3.00	98.00	3.00	3
AVERAGE							3.00

	IA1	A1	IA2	A2	IA3	A3	AVG
CO1	96.52	100					98.26
CO2	82.61	100	98.26	99.11			94.99
CO3			99.13	100			99.57
CO4			99.13	98.23	96.52	99.12	98.25
CO5					99.13	100	99.57

CO Attainment Level	Significance	For Direct attainment , 50% of CIE and 50% of SEE marks are considered.
Level 3	60% and above students should have scored >= 60% of Total marks	For indirect attainment, Course end survey is considered.
Level 2	55% to 59% of students should have scored >= 60% of Total marks	CO attainment is 90%of direct attainment + 10% of Indirect atttainment.
Level 1	50% to 54% of students should have scored >= 60% of Total marks	PO attainment = CO-PO mapping strength/3 * CO attainment .



						Co-Po Mapping Table								
CO'S	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PSO2
CO1	2	2	1	-	-	-	-	-		-	-	-	1	1
CO2	3	3	2	-	2	-	-	-	2	-	-	-	2	2
CO3	3	3	2	-	-	-	-	-		-	-	-	2	2
CO4	3	3	2	-	-	-	-	-		-	-	-	2	2
CO5	3	3	2	-	-	-	-	-		-	-	-	2	2
AVG	2.80	2.8	1.80	-	2.00	-	-	-	2.00	-	-	-	1.8	1.8

PO ATTAINMENT TABLE															
CO'S	CO Attainment in %	CO RESULT	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3.00	Y	2.00	2.00	1.00	-	-	-	-	-	-	-	-	-	1.00	1.00
CO2	3.00	Y	3.00	3.00	2.00	-	2	-	-	-	2.00	-	-	-	2.00	2.00
CO3	3.00	Y	3.00	3.00	2.00	-	-	-	-	-	-	-	-	-	2.00	2.00
CO4	3.00	Y	3.00	3.00	2.00	-	-	-	-	-	-	-	-	-	2.00	2.00
CO5	3.00	Y	3.00	3.00	2.00	-	-	-	-	-	-	-	-	-	2.00	2.00
Average			2.80	2.80	1.80	-	2.00	-	-	-	2.00	-	-	-	1.80	1.80

Course Incharge

HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K S Institute of Technology

Department of Computer Science and Engineering

Welcome to

2020-21 Even Semester

Online Class

On

Operating Systems

18CS43

By:

Vaneeta M

Associate Professor

Dept. of CSE

KSIT

Program Specific Outcomes

- PSO1: Ability to understand, analyze problems and implement solutions in programming languages, as well to apply concepts in core areas of Computer Science in association with professional bodies and clubs.
- PSO2: Ability to use computational skills and apply software knowledge to develop effective solutions and data to address real world challenges.

Institute Vision and Mission

Institute Vision

"To impart quality technical education with ethical values, employable skills and research to achieve excellence"

Institute Mission

- *To attract and retain highly qualified, experienced & committed faculty.*
- *To create relevant infrastructure.*
- *Network with industry & premier institutions to encourage emergence of new ideas by providing research & development facilities to strive for academic excellence.*
- *To inculcate the professional & ethical values among young students with employable skills & knowledge acquired to transform the society.*

Introduction To Operating Systems

Vision and Mission of Department

Vision:

"To create **competent professionals** in Computer Science and Engineering with adequate **skills** to drive the IT industry"

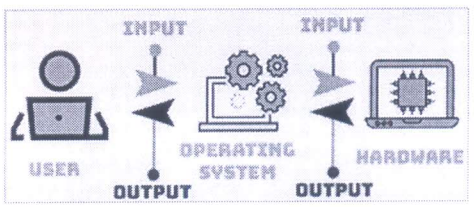
Mission:

M1: Impart sound **technical knowledge** and quest for **continuous learning**.

M2: To equip students to furnish Computer Applications for the society through **experiential learning** and research with **professional ethics**.

M3: Encourage **team work** through inter-disciplinary project and evolve as leaders with **social concerns**.

What is Operating System?



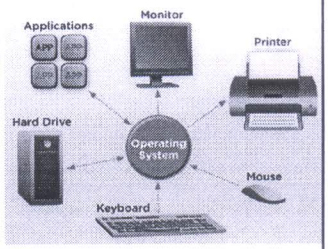
Operating System is a program that acts as an intermediary between user and computer hardware.

Goals of Operating System:

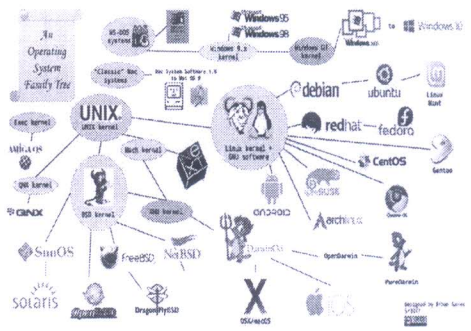
- Execute user programs.
- Make computer system convenient to use.
- Use the computer hardware in efficient manner.

Operating System as Resource Manager.

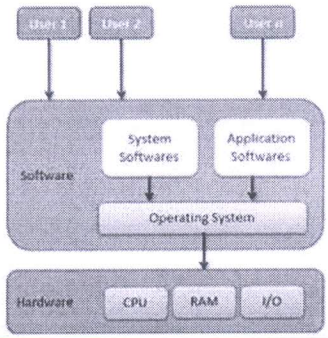
- An operating system (OS) is a **resource manager**.
- It takes the form of a **set of software routines** that allow users and application programs to access system resources in a **safe, efficient** and **abstract** way.



Many Flavors: The OS Family Tree

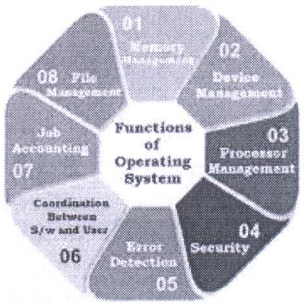


Operating System Structure

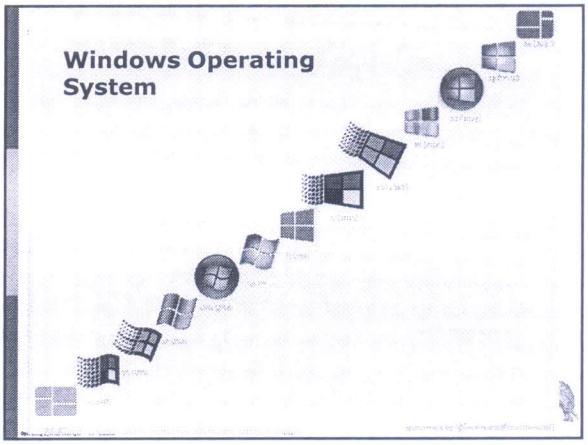


30 Companies and Devices Running on GNU/Linux

Functions/ Services of Operating System



Windows Operating System



Course Outcome

CO1	Identify the need and various types of Operating Systems.	Applying (K3)
CO2	Apply suitable techniques for process scheduling, synchronization and thread management.	Applying (K3)
CO3	Make use of deadlock and memory management schemes for managing the operating system	Applying (K3)
CO4	Determine the need of demand paging, file and directory management.	Applying (K3)
CO5	Utilize different scheduling schemes for disk scheduling	Applying (K3)

1.1 What is an Operating System?

- A program that acts as an **intermediary** between a user of a computer and the computer hardware
- Operating system goals:
  - **Execute** user programs and make solving user problems easier
  - Make the computer system **convenient** to use
  - Use the computer hardware in an **efficient** manner

Module 1 Syllabus

Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments.

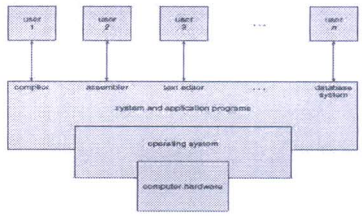
Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot.

Process Management Process concept; Process scheduling; Operations on processes; Inter process communication

Text book 1: Chapter 1, 2.1, 2.3, 2.4, 2.5, 2.6, 2.8, 2.9, 2.10, 3.1, 3.2, 3.3, 3.4

1.1 What is an Operating System do?

Four components of a computer :
1) Hardware
2) OS
3) Application programs and
4) Users



Abstract view of the components of a computer system

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition,Wiley-India, 2006

Reference Books:

1. Ann McHoes Ida M Flynn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition,PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

1.1.1 User View

The user's view of the computer depends on the interface being used:

1. **PC consisting of a monitor**, keyboard and system-unit. The OS is designed mostly for ease of use.
   - Some attention is paid to performance.
   - No attention is paid to resource utilization.
   - The OS is optimized for the single-user experience.

2. Some users use a **terminal connected** to a mainframe or (a minicomputer). The OS is designed
   - to maximize resource utilization.
   - to assure that no individual user takes more than her fair share.

3. Some users use a **workstation connected to network.**
The users have dedicated resources such as networking and servers.

4. Some users use a **handheld computer.**
The OS is designed mostly for individual usability.
Performance per unit of battery life is a very important factor.

1.1.2System View

An OS as a resource allocator
Resources used to solve a computing-problem:

- CPU time
- memory-space
- file-storage space and
- I/0 devices.

The OS manages and allocates the above resources to programs and the users.

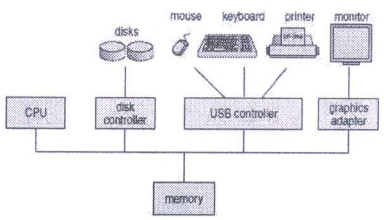
An OS is a control program
The OS is needed to control:

- operations of I/0 devices and
- execution of user-programs to prevent errors.

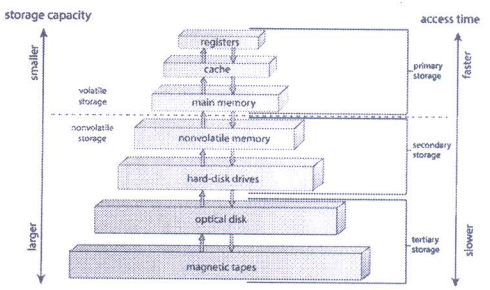
Common Functions of Interrupts

- Interrupt transfers control to the ISR generally, through the **interrupt-vector**, which contains the addresses of all the service routines.
- Interrupt architecture must **save the address** of the interrupted-instruction.
- Incoming interrupts are **disabled** while another interrupt is being processed to prevent a lost interrupt.
- A **trap** is a software-generated interrupt caused either by an error or a user request.

1.2.1 Computer System Organization



1.2.2 Storage-Device Hierarchy



Following events occur for a computer to start running:

- **Bootstrap program** is an initial program that runs when computer is powered-up.
  - Bootstrap program: initializes all the system registers memory-contents
  - loads OS into memory.
- Then, OS
  - starts executing the first process (such as "init") and
  - waits for some event to occur.

- **Hardware interrupt or the software interrupt**

i) Hardware may trigger an interrupt by **sending a signal to the CPU.**
ii) Software may trigger an interrupt by **executing a system-call.**
When CPU is interrupted, the CPU
stops current computation and
transfers control to ISR (interrupt service routine).

Finally, the ISR executes; on completion, the CPU resumes the interrupted computation

1.2.3 I/O Structure
A computer consists of CPUs and multiple device controllers (Figure 1.5).

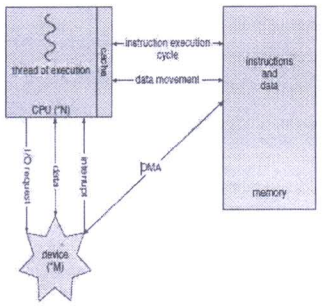
A **controller** is in charge of a specific type of device.
The controller maintains

- some **local buffer** and
- set of special-purpose **registers**.

Typically, OS has a device-driver for each controller.
Interrupt-driven I/O:

- Driver **loads** the appropriate registers within the controller.
- Controller **examines** the contents of registers to determine what action to take.
- Controller **transfers** data from the device to its local buffer.
- Controller informs the driver via an **interrupt** that it has finished its operation.
- Driver then **returns** control to the OS.

Problem: Interrupt-driven I/O produces high overhead when used for bulk data-transfer.
Solution: Use DMA (direct memory access).



Advantages:
1) Increased Throughput
By increasing no. of processors, we expect to get more work done in less time.

2) Economy of Scale
These systems are cheaper because they can share
→ peripherals → mass-storage → power-supply.

If many programs operate on same data, they will be stored on one disk & all processors can share them.
3) Increased Reliability
The failure of one processor will not halt the system.

- **Two types of multiple-processor systems:**

1)Asymmetric multiprocessing (AMP) and

2)Symmetric multiprocessing (SMP)

1.3 Computer System Architecture

1. Single-Processor Systems
2. Multiprocessor Systems
3. Clustered Systems

1.3.1 Single Processor Systems
The system has only one **general-purpose CPU**.
The CPU is capable of executing a general-purpose instruction-set.
These systems range from PDAs through mainframes.
Almost all systems have following processors: Include disk, keyboard, and graphics controllers.
Special-purpose processors run a limited instruction set and do not run user-processes.

1.3.2 Multi-Processor Systems

- These systems have two or more processors which can share:

→ bus → clock → memory/peripheral devices

1) Asymmetric Multiprocessing

- This uses **master-slave** relationship
- Each processor is assigned **a specific task.**
- A master-processor **controls** the system.

The master-processor schedules and allocates work to the slave-processors.
2) Symmetric Multiprocessing
- Each processor runs an **identical** copy of OS.

All processors are **peers**; no master-slave relationship exists between processors.
Advantages:
- Many processes can run simultaneously.
- Processes and resources are shared dynamically among the various processors.

Disadvantages
- Since CP s are separate, one CPU may be sitting **idle** while another CPU is overloaded. This results in inefficiencies.

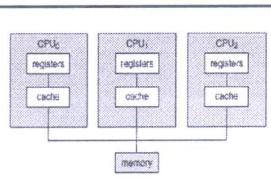
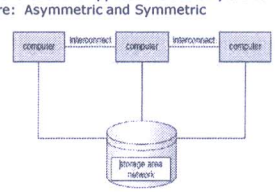


Figure 1.6 Symmetric multiprocessing architecture

1.3.3 Clustered Systems
These systems consist of **two or more systems coupled** together. These systems **share** storage & closely linked via LAN.

Advantage:
Used to provide **high-availability** service.
High-availability is obtained by adding a level of **redundancy** in the system.
Working procedure:
- A **cluster-software** runs on the cluster-nodes.
- Each node can **monitor** one or more other nodes (over the LAN).
- If the monitored-node fails, the monitoring-node can take ownership of failed-node"s storage and restart the applications running on the failed-node.

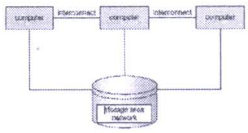
The users and clients of the applications see only a brief interruption of service.
- Two types are: Asymmetric and Symmetric



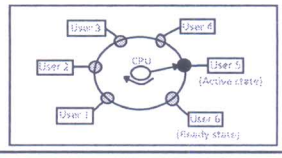
Asymmetric Clustering
One node is in **hot-standby mode** while the other nodes are running the applications.
The hot-standby node does nothing but **monitor** the active-server.

If the **server fails**, the hot-standby node becomes the active server.
Two or more nodes are running applications, and are monitoring each other.
Advantage:

This mode is more **efficient**, as it uses all of the available hardware.
It does require that more than one application be available to run.



1.4.3 Time Sharing Systems
- Time sharing (or multitasking) is a **logical extension of multiprogramming**.
- The CPU executes **multiple jobs by switching** between them.
- Switching between jobs occur so frequently that the users can interact with each program while it is running.
- Many users are allowed to **share the computer** simultaneously.
- CPU scheduling and multiprogramming are used to provide each user with a **small portion of a time-shared computer.**
- To obtain a good response time, jobs may have to be **swapped** in and out of main memory to the disk (called as backing store).



1.4 Operating System Structure

1. Batch Systems
2. Multi-Programmed Systems
3. Time-Sharing Systems

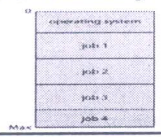
1.4.1 Batch Systems

- Early computers were physically enormous machines run from a console.
- The common **input devices were card readers** and tape drives.
- The common **output devices** were line printers, tape drives, and card punches.
- The user prepared a job which consisted of the program, the data, and control information submitted the job to the **computer-operator.**
- The job was usually in the form of **punch cards.**
- At some later time (after minutes, hours, or days), the **output appeared.**
- To speed up processing, operators **batched** together jobs with similar needs and ran them through the computer as a **group**.
- **Disadvantage**: 1) The CPU is often idle, because the speeds of the mechanical I/O devices.

- **Virtual memory** is a technique that allows the execution of a job that may not be completely in memory.
- **Advantage of virtual-memory**:
  - Programs can be larger than physical memory, even then programs can be executed.

1.4.2 Multi-Programmed Systems

- Multiprogramming **increases CPU utilization** by organizing jobs so that the CPU always has one to execute.
- The idea is as follows:
  - OS keeps **several jobs** in memory simultaneously
  - OS picks and begins to execute **one of the jobs** in the memory.
  - OS simply switches to, and executes, another job.
  - When that job needs to wait, the CPU is **switched** to another job, and so on.
  - As long as at least one job needs to execute, the **CPU is never idle.**
- If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision is **job scheduling.**
- If several jobs are ready to run at the same time, the system must choose among them. Making this decision is **CPU scheduling.**



1.6 Operating System Operations

- Modern OS is **interrupt** driven. Events are always signaled by the occurrence of an interrupt or a trap.
- Operating system and the users **share** the hardware and software resources of the computer system
- **Error** in a user program could cause problems only for the one program that was running.
- **For example,** if a process gets stuck in an **infinite** loop, this loop could prevent the correct operation of many other processes.
- One erroneous program might **modify** another program, the data of another program, or even the operating system itself.
- A **properly designed operating system** must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.

To execute a program:

- The program will be
  - **loaded** into memory and
  - **mapped** to absolute addresses.
- Then, program accesses instructions & data from memory by generating **absolute addresses**.
- Finally, when program **terminates**, its memory-space is freed.
- To improve CPU utilization, **keep several programs** will be kept in memory

1.8.2 Mass Storage Management

The OS is **responsible** for following activities:

- Free-space management
- Storage allocation and
- Disk scheduling.

Usually, disks used to **store**

- data that **does not fit** in main memory or
- data that must be kept for a **"long"** period of time.

Most programs are **stored** on disk until loaded into memory.

The programs include

- compilers
- word processors and
- editors.

The programs use the disk as both the source and destination of their processing.

Entire speed of computer operation depends on disk and its algorithms.

1.8 Storage Management

1. File-System Management
2. Mass-Storage Management
3. Caching

1.8.1 File System Management

The OS is responsible for following activities:

- Creating and deleting **files**.
- Creating and deleting **directories**.
- Supporting primitives for **manipulating** files & directories.
- Mapping files onto **secondary** storage.
- **Backing** up files on stable (non-volatile) storage media.

Computer stores information on different types of physical media. For ex: magnetic disk, optical disk.

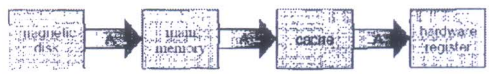
Each medium is controlled by a device (e.g. disk drive).

The **OS**

- **maps** files onto physical media and
- **accesses** the files via the storage devices

1.8.3 Caching

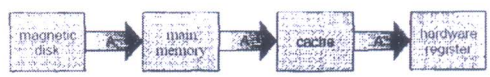
- Information is normally kept in some storage system (such as main memory).
- As it is used, it is copied into a **faster storage system called as the cache** on a temporary basis.
- When we need a particular piece of information:
  - **Check** whether the information is in the cache.
  - If information is in cache, **use the information** directly from the cache.
  - If information is not in cache, use the information from the **source**, putting a copy in the cache under the assumption that we will need it again soon.



File is a logical collection of related information.

- File consists of both program & data.
- Data files may be numeric, alphabets or binary.
- When multiple users have access to files, access control (read, write) must be specified.

- Data transfer from cache to CPU and registers is usually a hardware function, with no operating-system intervention.
- In contrast, transfer of data from disk to memory is usually
- controlled by the operating system.
- **cache coherency** occurs in multiprocessing.
- Most systems have an **instruction cache** to hold the instructions expected to be executed next.
- Most systems have **one or more high-speed data** caches in the memory hierarchy
- Because caches have limited size, cache management is an important design problem

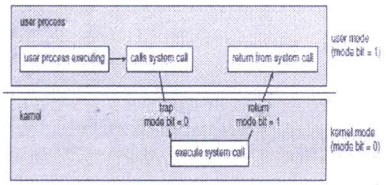


1.6.1 Dual Mode Operation

Problem: Ability to differentiate between the execution of OS code and user-defined code.

Two modes of operation: 1) User mode and 2) Kernel mode

A mode bit is a bit added to the hardware of the computer to indicate the current mode: i.e. kernel (0) or user (1)



Working principle:

- At system **boot time**, the hardware starts in kernel-mode.
- The **OS is then loaded** and starts user applications in user-mode.
- Whenever a **trap or interrupt occurs**, the hardware switches from user-mode to kernel-mode (that is, changes the state of the mode bit to 0).
- The system always switches to user-mode (by setting the mode bit to 1) before passing control to a user-program.

Dual mode protects

- OS from errant users and
- errant users from one another.

- Privileged instruction is executed only in kernel-mode.
- If an attempt is made to execute a privileged instruction in user-mode, the hardware treats it as **illegal and traps it to the OS.**
- A **system calls** are called by user-program to ask the OS to perform the tasks on behalf of the user program.

1.5.2 Timer

Problem: Allow a user-program to get stuck in an infinite loop and never return control to the OS.
Solution: use a timer.

A timer can be set to interrupt the computer after a **specific period**. The period may be fixed (**for ex: 1/60 second**) or variable (for ex: from 1ns to 1ms).

A variable timer is implemented by a fixed-rate clock and a counter.
Working procedure:

- The OS sets the **counter**.
- Every time the clock ticks, the counter is **decremented**.
- When the counter reaches **0**, an interrupt occurs.

- The instructions that modify the content of the timer are **privileged instructions.**

1.6 Process Management

The OS is responsible for the following activities:

- **Creating** and deleting both user and system processes
- **Suspending** and resuming processes
- Providing mechanisms for process **synchronization**
- Providing mechanisms for process **communication**
- Providing mechanisms for **deadlock** handling

A process needs following **resources** to do a task:

- CPU
- memory and
- files.

The resources are **allocated** to process

- when the process is **created** or
- while the process is **running**.

When the process terminates, the OS **reclaims** all the reusable resources.

A program by itself is not a process;
A program is a passive entity (such as the contents of a file stored on disk).
A process is an active entity.
Two types of process:

Single-threaded process has one PC(program counter) which specifies location of the next instruction to be executed.

Multi-threaded process has one PC per thread which specifies location of next instruction to execute in each thread

1.7 Memory Management

The OS is responsible for the following activities:

- Keeping **track** of which parts of memory are currently being used and by whom
- **Deciding** which processes are to be loaded into memory when memory space becomes available
- **Allocating** and de-allocating memory space as needed.

- Main memory is the array of bytes ranging from hundreds to billions.
- Each byte has its own address.
- The CPU
  - **reads** instructions from main memory during the **instruction-fetch cycle.**
  - reads/writes **data** from/to main-memory during the **data-fetch cycle.**

OPERATING SYSTEMS

MODULE 1: INTRODUCTION OPERATING-SYSTEM STRUCTURES PROCESSES

1.1 Operating System

1.1 What Operating Systems Do
- 1.1.1 User View
- 1.1.2 System View

1.2 Computer-System Organization
- 1.2.1 Computer System Organization
- 1.2.2 Storage Structure
- 1.2.3 I/O Structure

1.3 Computer-System Architecture
- 1.3.1 Single Processor Systems
- 1.3.2 Multiprocessor Systems
- 1.3.3 Clustered Systems

1.4 Operating-System Structure
- 1.4.1 Batch systems
- 1.4.2 Multi-Programmed Systems
- 1.4.3 Time-Sharing systems

1.5 Operating-System Operations
- 1.5.1 Dual Mode Operation
- 1.5.2 Timer

1.6 Process Management

1.7 Memory Management

1.8 Storage Management
- 1.8.1 File System Management
- 1.8.2 Mass Storage Management
- 1.8.3 Caching
- 1.8.4 I/O Systems

1.9 Protection and Security

1.10 Distributed System

1.11Special-Purpose Systems
- 1.11.1 Real-Time Embedded Systems
- 1.11.2 Multimedia Systems
- 1.11.3 Handheld Systems

1.12 Computing Environments
- 1.12.1 Traditional Computing
- 1.12.2 Client-Server Computing
- 1.12.3 Peer-to-Peer Computing
- 1.12.4 Web Based Computing

Chapter 2

2.1 Operating-System Services

2.2 User and Operating-System Interface

2.3 System Calls

2.4 Types of System Calls

- 2.4.1 Process Control
- 2.4.2 File Management
- 2.4.3 Device Management
- 2.4.4 Information Maintenance

- 2.4.5 Communication
  - 2.4.5.1 Message Passing Model
  - 2.4.5.2 Shared Memory Model

2.5 System Programs
2.6 Operating-System Design and Implementation

- 2.6.1 Design Goals
- 2.6.2 Mechanisms & Policies
- 2.6.3 Implementation

2.7 Operating-System Structure

- 2.7.1 Simple Structure
- 2.7.2 Layered Approach
- 2.7.3 Micro-Kernels
- 2.7.4 Modules

2.8 Virtual Machines
2.9 Operating-System Generation
2.10 System Boot

Chapter 3:

3.1 Process Concept

- 3.1.1 The Process
- 3.1.2 Process State
- 3.1.3 Process Control Block

3.2 Process Scheduling

- 3.2.1 Scheduling Queues
- 3.2.2 Schedulers

- 3.2.3 Context Switch

3.3 Operations on Processes

- 3.3.1 Process Creation
- 3.3.2 Process Termination

3.4 Inter-process Communication

- 3.4.1 Shared-Memory Systems
- 3.4.2 Message-Passing Systems
  - 3.4.2.1 Naming

  - 3.4.2.2 Synchronization
  - 3.4.2.3 Buffering

MODULE 1: CHAPTER 1: INTRODUCTION

Operating System

- An OS is a program that acts as an intermediary between
  → computer-user and
  → computer-hardware.

- It also provides a basis for application-programs
- Goals of OS:
  - To execute programs.
  - To make solving user-problems easier.
  - To make the computer convenient to use.
- The OS (also called kernel) is the one program running at all times on the computer.

- Different types of OS:
  - Mainframe OS is designed to optimize utilization of hardware.
  - Personal computer (PC) OS supports complex game, business application.
  - Handheld computer OS is designed to provide an environment in which a user can easily interface with the computer to execute programs.

1.1What Operating Systems do?

- Four components of a computer (Figure 1.1):
  1) Hardware
  2) OS
  3) Application programs and
  4) Users

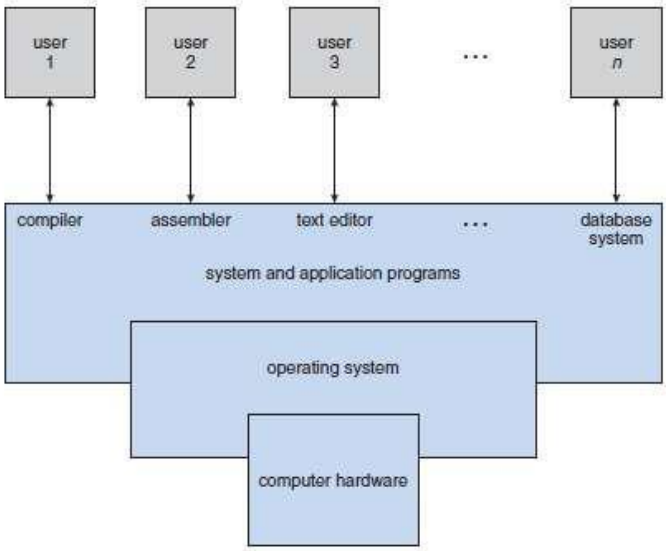


Figure 1.1 Abstract view of the components of a computer system

- Hardware provides basic computing-resources:
  → CPU
  → memory and
  → I/O devices.

- Application-program defines how the resources are used to solve computing-problems of the users. Ex: word processors, spread sheets, compilers.

• The OS controls & co-ordinates the use of hardware among various application-program for various users.

• Two views of OS: 1) User and
2) System.

1.1.1 User View

• The user's view of the computer depends on the interface being used:

- Most users use a PC consisting of a monitor, keyboard and ystem-unit.
- The OS is designed mostly for ease of use.
- Some attention is paid to performance.
- No attention is paid to resource utilization.

The OS is optimized for the single-user experience.

1) Some users use a terminal connected to a mainframe or (a minicomputer). The OS is designed

→ to maximize resource utilization.
→ to assure that no individual user takes more than her fair share.

2) Some users use a workstation connected to network.
- The users have dedicated resources such as networking and servers.
- The OS is designed to compromise between
→ individual usability and
→ resource utilization.

4) Some users use a handheld computer.

- The OS is designed mostly for individual usability.
- Performance per unit of battery life is a very important factor.

1.1.2 System View

1) An OS as a resource allocator
- Resources used to solve a computing-problem:
  - CPU time
  - memory-space
  - file-storage space and
  - I/0 devices.
- The OS manages and allocates the above resources to programs and the users.

2) An OS is a control program
The OS is needed to control:

→ operations of I/0 devices and
→ execution of user-programs to prevent errors.

1.2 Computer System Organization

1.2.1 Computer System Organization

A computer consists of : one or more CPUs and no. of device-controllers (Figure 1.2).

• Controller is in charge of a specific type of device (for ex: audio devices).

- CPU and controllers can execute concurrently.
- A memory-controller is used to synchronize access to the shared-memory.

- Following events occur for a computer to start running:
  1) Bootstrap program is an initial program that runs when computer is powered-up
     → Bootstrap program: initializes all the system from registers to memory contents
     → loads OS into memory.

  2) Then, OS
     → starts executing the first process (such as "init") and
     → waits for some event to occur.

  3) The occurrence of an event is signaled by an interrupt from either the hardware or the software (Figure 1.3).

     i) Hardware may trigger an interrupt by sending a signal to the CPU. ii) Software may trigger an interrupt by executing a system-call.

  4) When CPU is interrupted, the CPU
     → stops current computation and
     → transfers control to ISR (interrupt service routine).

  5) Finally, the ISR executes; on completion, the CPU resumes the interrupted computation.

Common Functions of Interrupts

- Interrupt transfers control to the ISR generally, through the interrupt-vector, which contains the addresses of all the service routines. Interrupt architecture must save the address of the interrupted-instruction.

- Incoming interrupts are disabled while another interrupt is being processed to prevent a lost interrupt.

- A trap is a software-generated interrupt caused either by an error or a user request.
- A modern OS is interrupt-driven.

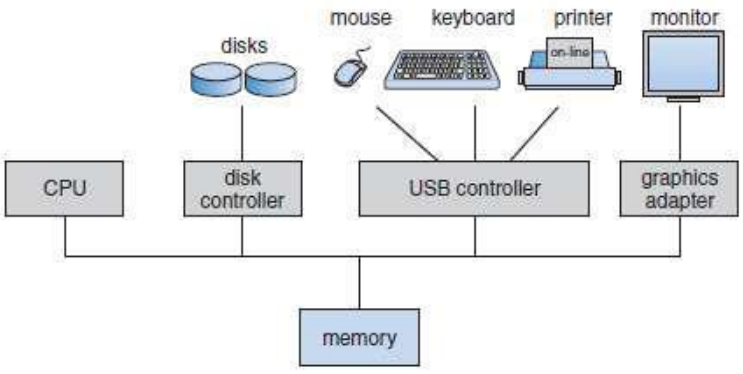


Figure 1.2 A modern computer system

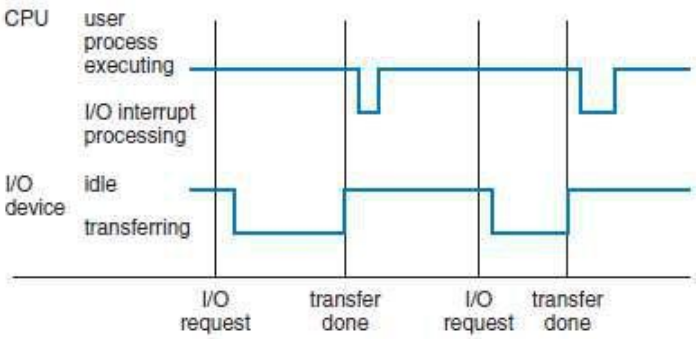


Figure 1.3 Interrupt time line for a single process doing output

1.2.2 Storage Structure

- Programs must be in main-memory (also called RAM) to be executed.

- Interaction with memory is done through a series of load or store instructions.
  Moves a word from main-memory to an internal register within the CPU
  Moves the content of a register to main-memory.
- Also, the CPU automatically loads instructions from main-memory for execution.
- Ideally, we want the programs & data to reside in main-memory permanently. This is not possible for

2 reasons:

1) Main-memory is small.

2) Main-memory is volatile i.e. it loses its contents when powered-off.

- Most computers provide secondary-storage as an extension of main-memory.
- Main requirement:

The wide range of storage-systems can be organized in a hierarchy (Figure 1.4).

- The higher levels are expensive, but they are fast.
  The lower levels are inexpensive, but they are slow.

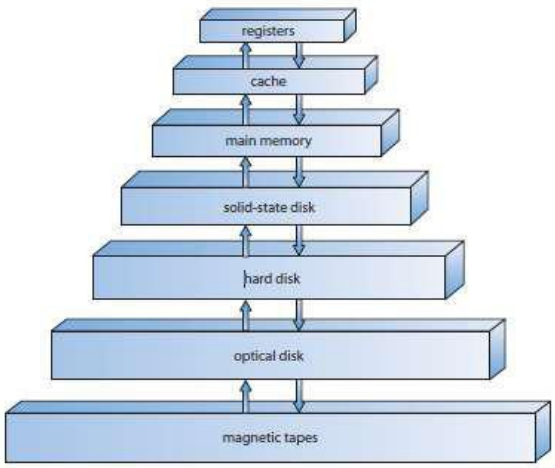


Figure 1.4 Storage-device hierarchy

1.2.3 I/O Structure

- A computer consists of CPUs and multiple device controllers (Figure 1.5).

- A controller is in charge of a specific type of device.
- The controller maintains
  → some local buffer and

  → set of special-purpose registers.

- Typically, OS has a device-driver for each controller.
- Interrupt-driven I/O:

1) Driver loads the appropriate registers within the controller.
2) Controller examines the contents of registers to determine what action to take.
3) Controller transfers data from the device to its local buffer.
4) Controller informs the driver via an interrupt that it has finished its operation.

5) Driver then returns control to the OS.

- Problem: Interrupt-driven I/O produces high overhead when used for bulk data-transfer.
  Solution: Use DMA (direct memory access).

- In DMA, the controller transfers blocks of data from buffer-storage directly to main memory without CPU intervention.

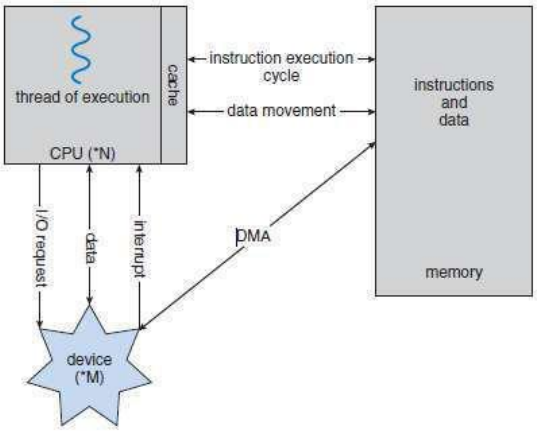


Figure 1.5 How a modern computer system works

1.3 Computer System Architecture

1) Single-Processor Systems

2) Multiprocessor Systems
3) Clustered Systems

1.3.1 Single Processor Systems

- The system has only one general-purpose CPU.
- The CPU is capable of executing a general-purpose instruction-set.
- These systems range from PDAs through mainframes.
- Almost all systems have following processors:

Include disk, keyboard, and graphics controllers.

2) General Purpose Processors

Include I/O processors.

- Special-purpose processors run a limited instruction set and do not run user-processes.

1.3.2 Multi-Processor Systems

- These systems have two or more processors which can share:

→ bus → clock → memory/peripheral devices

• Advantages:

1) Increased Throughput

By increasing no. of processors, we expect to get more work done in less time.

2) Economy of Scale

These systems are cheaper because they can share
→ peripherals → mass-storage → power-supply.

If many programs operate on same data, they will be stored on one disk & all processors can share them.

3) Increased Reliability

The failure of one processor will not halt the system.

• Two types of multiple-processor systems: 1) Asymmetric multiprocessing (AMP) and

2) Symmetric multiprocessing (SMP)

1) Asymmetric Multiprocessing

- This uses master-slave relationship (Figure 1.6).
- Each processor is assigned a specific task.
- A master-processor controls the system.

- The master-processor schedules and allocates work to the slave-processors.

2) Symmetric Multiprocessing

• Each processor runs an identical copy of S.

- All processors are peers; no master-slave relationship exists between processors.
- Advantages:
  1) Many processes can run simultaneously.
  2) Processes and resources are shared dynamically among the various processors.

• Disadvantage:

1) Since CP s are separate, one CPU may be sitting idle while another CPU is overloaded. This results in inefficiencies.

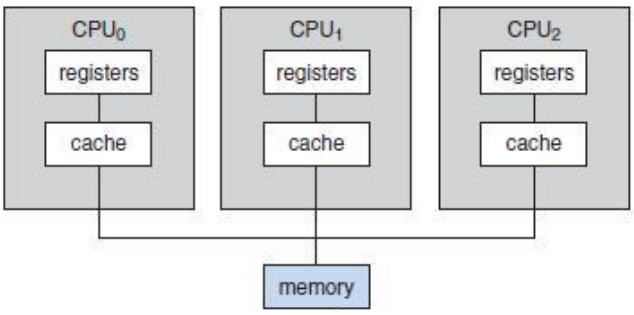


Figure 1.6 Symmetric multiprocessing architecture

1.3.3 Clustered Systems

- These systems consist of two or more systems coupled together (Figure 1.7).

- These systems share storage & closely linked via LAN.
- Advantage:
    1) Used to provide high-availability service.
- High-availability is obtained by adding a level of redundancy in the system.
- Working procedure:
    - A cluster-software runs on the cluster-nodes.
    - Each node can monitor one or more other nodes (over the LAN).
    - If the monitored-node fails, the monitoring-node can
        - take ownership of failed-node"s storage and
        - restart the applications running on the failed-node.

    - The users and clients of the applications see only a brief interruption of service.

- Two types are: Asymmetric and Symmetric

1) Asymmetric Clustering
- One node is in hot-standby mode while the other nodes are running the applications.
- The hot-standby node does nothing but monitor the active-server.

- If the server fails, the hot-standby node becomes the active server.
- Two or more nodes are running applications, and are monitoring each other.
- Advantage:

    1) This mode is more efficient, as it uses all of the available hardware.
- It does require that more than one application be available to run.

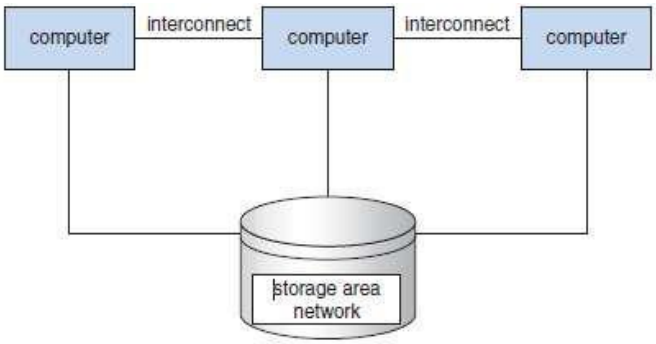


Figure 1.7 General structure of a clustered system

1.4 Operating System Structure

1) Batch Systems

2) Multi-Programmed Systems
3. Time-Sharing Systems

1.4.1 Batch Systems

- Early computers were physically enormous machines run from a console.
- The common input devices were card readers and tape drives.
- The common output devices were line printers, tape drives, and card punches.
- The user
    → prepared a job which consisted of the program, the data, and control information
    → submitted the job to the computer-operator.

• The job was usually in the form of punch cards.
• At some later time (after minutes, hours, or days), the output appeared.

• To speed up processing, operators batched together jobs with similar needs and ran them through the computer as a group.

- Disadvantage: 1) The CPU is often idle, because the speeds of the mechanical I/O devices.

1.4.2 Multi-Programmed Systems

• Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.
• The idea is as follows:

1) OS keeps several jobs in memory simultaneously (Figure 1.8).
2) OS picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete.
3) OS simply switches to, and executes, another job.
4) When that job needs to wait, the CPU is switched to another job, and so on.
5) As long as at least one job needs to execute, the CPU is never idle.

• If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision is job scheduling.
• If several jobs are ready to run at the same time, the system must choose among them. Making this decision is CPU scheduling.

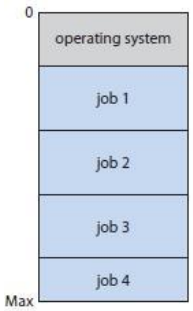


Figure 1.8 Memory layout for a multiprogramming system

1.4.3 Time Sharing Systems

• Time sharing (or multitasking) is a logical extension of multiprogramming.

• The CPU executes multiple jobs by switching between them.

• Switching between jobs occur so frequently that the users can interact with each program while it is running.
• Many users are allowed to share the computer simultaneously.

• CPU scheduling and multiprogramming are used to provide each user with a small portion of a time-shared computer.

• To obtain a good response time, jobs may have to be swapped in and out of main memory to the disk (called as backing store).

• Virtual memory is a technique that allows the execution of a job that may not be completely in memory.

• Advantage of virtual-memory:

1) Programs can be larger than physical memory.

• Main requirements:
  • The system must provide a file-system.

  • The system must provide disk-management.
  • The system must provide CPU-scheduling to support concurrent execution.
  • The system must provide job-synchronization to ensure orderly execution.

1.5 Operating System Operations

• Modern OS is interrupt driven.
• Events are always signaled by the occurrence of an interrupt or a trap.

• ISR (Interrupt Service Routine) is provided that is responsible for dealing with the interrupt.

• Since the operating system and the users share the hardware and software resources of the computer system, we need to make sure that an error in a user program could cause problems only for the one program that was running.
• **For example,** if a process gets stuck in an infinite loop, this loop could prevent the correct operation of many other processes.
• One erroneous program might modify another program, the data of another program, or even the operating system itself.
• A properly designed operating system must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.

1.5.1 Dual Mode Operation

• Problem: We must be able to differentiate between the execution of
  → OS code and user-defined code.
• Two modes of operation (Figure 1.9): 1) User mode and
  2) Kernel mode

• A mode bit is a bit added to the hardware of the computer to indicate the current mode: i.e. kernel (0) or user (1)

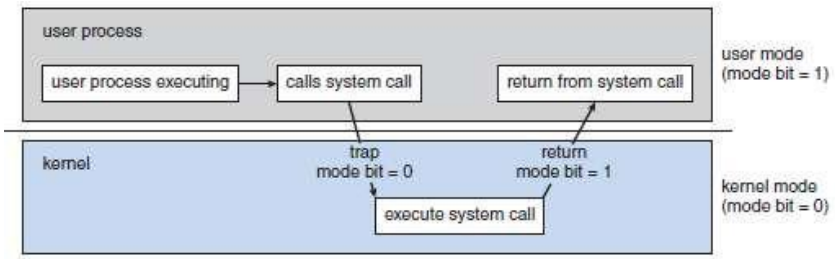


Figure 1.9 Transition from user to kernel mode

• Working principle:
  1) At system boot time, the hardware starts in kernel-mode.
  2) The OS is then loaded and starts user applications in user-mode.

  3) Whenever a trap or interrupt occurs, the hardware switches from user-mode to kernel-mode (that is, changes the state of the mode bit to 0).

4) The system always switches to user-mode (by setting the mode bit to 1) before passing control to a user-program.

- Dual mode protects
  → OS from errant users and
  → errant users from one another.

- Privileged instruction is executed only in kernel-mode.

- If an attempt is made to execute a privileged instruction in user-mode, the hardware treats it as illegal and traps it to the OS.

- A system calls are called by user-program to ask the OS to perform the tasks on behalf of the user program.

1.5.2 Timer

- Problem: We cannot allow a user-program to get stuck in an infinite loop and never return control to the OS.
  Solution: We can use a timer.

- A timer can be set to interrupt the computer after a specific period.
- The period may be fixed (for ex: 1/60 second) or variable (for ex: from 1ns to 1ms).

- A variable timer is implemented by a fixed-rate clock and a counter.
- Working procedure:
  1) The OS sets the counter.
  2) Every time the clock ticks, the counter is decremented.
  3) When the counter reaches 0, an interrupt occurs.
- The instructions that modify the content of the timer are privileged instructions.

1.6 Process Management

- The OS is responsible for the following activities:
  1) Creating and deleting both user and system processes
  2) Suspending and resuming processes
  3) Providing mechanisms for process synchronization
  4) Providing mechanisms for process communication
  5) Providing mechanisms for deadlock handling
- A process needs following resources to do a task:
  → CPU
  → memory and
  → files.

- The resources are allocated to process
  → when the process is created or
  → while the process is running.

- When the process terminates, the OS reclaims all the reusable resources.

- A program by itself is not a process;
  1) A program is a passive entity (such as the contents of a file stored on disk).
  2) A process is an active entity.
- Two types of process:

  1) Single-threaded process has one PC(program counter) which specifies location of the next instruction to be executed.

2) Multi-threaded process has one PC per thread which specifies location of next instruction to execute in each thread

1.7 Memory Management

- The OS is responsible for the following activities:

  1) Keeping track of which parts of memory are currently being used and by whom

  2) Deciding which processes are to be loaded into memory when memory space becomes available
  3) Allocating and de-allocating memory space as needed.

- Main memory is the array of bytes ranging from hundreds to billions.
- Each byte has its own address.
- The CPU
  → reads instructions from main memory during the instruction-fetch cycle.
  → reads/writes data from/to main-memory during the data-fetch cycle.

- To execute a program:

  1) The program will be
     → loaded into memory and
     → mapped to absolute addresses.

  2) Then, program accesses instructions & data from memory by generating absolute addresses.
  3) Finally, when program terminates, its memory-space is freed.
- To improve CPU utilization, keep several programs will be kept in memory
- Selection of a memory-management scheme depends on hardware-design of the system.

1.8 Storage Management

1) File-System Management
2) Mass-Storage Management
3) Caching

1.8.1 File System Management

- The OS is responsible for following activities:
  1) Creating and deleting files.
  2) Creating and deleting directories.
  3) Supporting primitives for manipulating files & directories.

  4) Mapping files onto secondary storage.
  5) Backing up files on stable (non-volatile) storage media.

- Computer stores information on different types of physical media. For ex: magnetic disk, optical disk.
- Each medium is controlled by a device (e.g. disk drive).
- The OS
  → maps files onto physical media and

  → accesses the files via the storage devices

- File is a logical collection of related information.

- File consists of both program & data.
- Data files may be numeric, alphabets or binary.
- When multiple users have access to files, access control (read, write) must be specified.

1.8.2 Mass Storage Management

- The OS is responsible for following activities:
  1) Free-space management
  2) Storage allocation and
  3) Disk scheduling.
- Usually, disks used to store
  → data that does not fit in main memory or
  → data that must be kept for a "long" period of time.
- Most programs are stored on disk until loaded into memory.
- The programs include
  → compilers
  → word processors and
  → editors.
- The programs use the disk as both the source and destination of their processing.
- Entire speed of computer operation depends on disk and its algorithms.

1.8.3 Caching

- Caching is an important principle of computer systems.
- Information is normally kept in some storage system (such as main memory).
- As it is used, it is copied into a faster storage system called as the cache on a temporary basis.
- When we need a particular piece of information:
  1) We first check whether the information is in the cache.
  2) If information is in cache, we use the information directly from the cache.
  3) If information is not in cache, we use the information from the source, putting a copy in the cache under the assumption that we will need it again soon.
- In addition, internal programmable registers, such as index registers, provide high-speed cache for main memory.
- The compiler implements the register-allocation and register-replacement algorithms to decide which information to keep in registers and which to keep in main memory.
- Data transfer from cache to CPU and registers is usually a hardware function, with no operating-system intervention.
- In contrast, transfer of data from disk to memory is usually controlled by the operating system.
- **cache coherency** occurs in multiprocessing**.**
- Most systems have an instruction cache to hold the instructions expected to be executed next.
- Most systems have one or more high-speed data caches in the memory hierarchy
- Because caches have limited size, cache management is an important design problem
  Careful selection of cache size & of a replacement policy can result in greatly increased performance

Level	1	2	3	4
Name	registers	cache	raaJft memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000,000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operatingsystem	operating system
Backed by	cache	main memory	disk	CD or tape

1.8.4 I/O Systems

- The OS must hide peculiarities of hardware devices from users.

- In UNIX, the peculiarities of I/O devices are hidden from the bulk of the OS itself by the I/O subsystem.
- The I/O subsystem consists of
    1) A memory-management component that includes buffering, caching, and spooling.
    2) A general device-driver interface.
    3) Drivers for specific hardware devices.
- Only the device driver knows the peculiarities of the specific device to which it is assigned.

1.9 Protection and Security

- Protection is a mechanism for controlling access of processes or users to resources defined by OS.
- This mechanism must provide
    → means for specification of the controls to be imposed and
    → means for enforcement.

- Protection can improve reliability by detecting latent errors at the interfaces between subsystems.
- An unprotected resource cannot defend against use (or misuse) by an unauthorized or incompetent user.
- A protection-oriented system provides a means to distinguish between authorized and unauthorized usage.

Security

Security

- Security means defense of the system against internal and external attacks.
- The attacks include
    → viruses and worms
    → DOS(denial-of-service)
    → identity theft.

- Protection and security require the system to be able to distinguish among all its users.
    1) User identities (user IDs) include name and associated number, one per user.User IDs are associated with all files (or processes) of that user to determine access control.

    2) Group identifier (group ID): can be used to define a group name and the set of users belonging to that group.

A user can be in one or more groups, depending on operating-system design decisions.

1.10 Distributed System

- This is a collection of physically separate, possibly heterogeneous computer-systems.
- The computer-systems are networked to provide the users with access to the various resources.
- Access to a shared resource increases

→ computation speed
→ functionality

→ data availability and
→ reliability

- A network is a communication path between two or more systems.
- Networks vary by the
  → protocols used

  → distances between nodes and
  → transport media.

- Common network protocol are
  → TCP/IP
  → ATM.

- Networks are **characterized based on the distances** between their nodes.
  → A **local-area network (LAN)** connects computers within a building.
  → A wide-area network (WAN) usually links buildings, cities, or countries.

  → A metropolitan-area network (MAN) could link buildings within a city.

- The media to carry networks are equally varied. They include
  → copper wires,
  → fiber strands, and
  → wireless transmissions.

1.11 Special Purpose Systems

1) Real-Time Embedded Systems

2) Multimedia Systems
3) Handheld Systems

1.11.1 Real-Time Embedded Systems

- Embedded computers are the most prevalent form of computers in existence.

- These devices are found everywhere, from car engines and manufacturing robots to VCRs and microwave ovens.
- They tend to have very specific tasks.

- The systems they run on are usually primitive, and so the operating systems provide limited features.
- Usually, they prefer to spend their time monitoring & managing hardware devices such as
  → automobile engines and
  → robotic arms.

- Embedded systems almost always run real-time operating systems.

- A real-time system is used when rigid time requirements have been placed on the operation of a processor.

1.11.2 Multimedia Systems

- Multimedia data consist of audio and video files as well as conventional files.

- These data differ from conventional data in that multimedia data must be delivered(streamed) according to certain time restrictions.

• The nature of these data affects the design of operating systems that support the requirements of multimedia systems

• Multimedia describes a wide range of applications. These include
→ audio files such as MP3
→ DVD movies

→ video conferencing
→ live webcasts of speeches

1.11.3 Handheld Systems

• Handheld systems include
→ PDAs and
→ cellular telephones.
These systems use special purpose embedded operating systems.

• Main challenge faced by developers of handheld systems: Limited size of devices.

Because of small size 5 inches in height and 3 inches in width, and it weighs less than one-half pound, most handheld devices have a

• small amount of memory(512 KB and 128 MB)
• slow processors, Processors for most handheld devices run at a fraction of the speed of a processor in a PC. Most handheld devices use smaller, slower processors that consume less power. Therefore, the operating system and applications must be designed not to tax the processor.

→ small display screens home computer may measure up to 30 inches, the display for a handheld device is often no more than 3 inches square. Tasks, such as reading e-mail and browsing web pages, must be condensed into smaller displays. One approach for displaying the content in web pages is **web clipping,** where only a small subset of a web page is delivered and displayed on the handheld device.

1.12 Computing Environments

1) Traditional Computing

2) Client-Server Computing
3) Peer-to-Peer Computing
4) Web-Based Computing

1.12.1 Traditional Computing

• Used in office environment:
PCs connected to a network, with servers providing file and print services.
Terminals attached to mainframes were prevalent at many companies

• Web technologies are stretching the boundaries of traditional computing.
Companies establish portals, which provide web accessibility to their internal servers.
Network computers are terminals that understand web computing.
Handheld PDAs can connect to wireless networks to use company's web portal.

• Used in home networks:
At home, most users had a single computer with a slow modem.
Some homes have firewalls to protect their networks from security breaches.

• Systems were either batch or interactive.
1) Batch system processed jobs in bulk, with predetermined input.
2) Interactive systems waited for input from users.

1.12.2 Client-Server Computing

• Servers can be broadly categorized as (Figure 1.10): 1) Compute servers and

2) File servers

1) Compute-server system provides an interface to which a client can send a request to perform an action (for example, read data).

- In response, the server executes the action and sends back results to the client.

2) File-server system provides a file-system interface where clients can create, read, and delete files.

- For example: web server that delivers files to clients running web browsers.

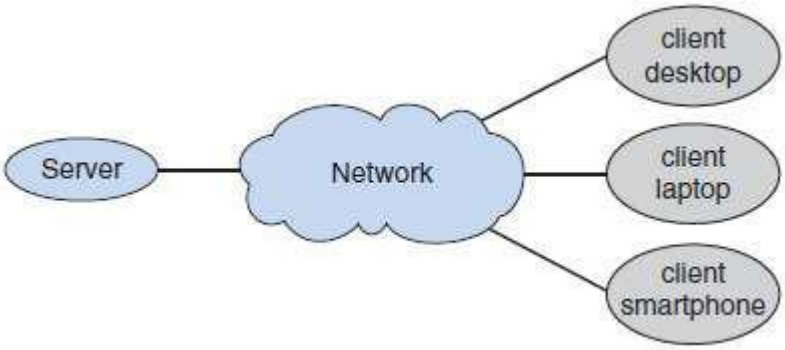


Figure 1.10 General structure of a client–server system.

1.12.3 Peer-to-Peer Computing

• All nodes are considered peers, and each may act as either a client or a server(Figure 1.11).

• Advantage:

1) In a client-server system, the server is a bottleneck;

but in a peer-to-peer system, services can be provided by several nodes distributed throughout the network.

• A node must first join the network of peers.
• Determining what services are available is done in one of two general ways:

1) When a node joins a network, it registers its service with a centralized lookup service on the network.

Any node desiring a specific service first contacts this centralized lookup service to determine which node provides the service.

2) A peer broadcasts a request for the service to all other nodes in the network. The node (or nodes) providing that service responds to the peer.

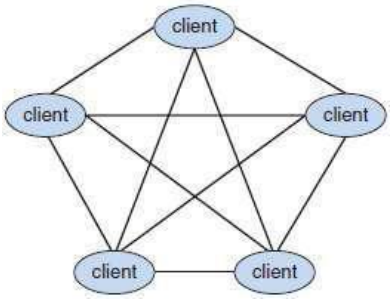


Figure 1.11 Peer-to-peer system with no centralized service.

1.12.4 Web-Based Computing

- The Web has become ubiquitous, leading to more access by a wider variety of devices
- This includes
  → PC
  → handheld PDA &
  → cell phones

Web computing has increased the emphasis on networking. Devices that were not previously networked now include wired or wireless access, have faster network connectivity, provided by either improved networking technology, optimized network implementation code,or both.

- Load balancer is a new category of devices to manage web traffic among similar servers.
- In load balancing, network connection is distributed among a pool of similar servers.

- More devices becoming networked to allow web access

- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows XP, which can be clients and servers

MODULE 1 (CONT.): CHAPTER 2

OPERATING-SYSTEM STRUCTURES

2.1 Operating System Services

- An OS provides an environment for the execution of programs.
- It provides services to
  → programs and
  → users.

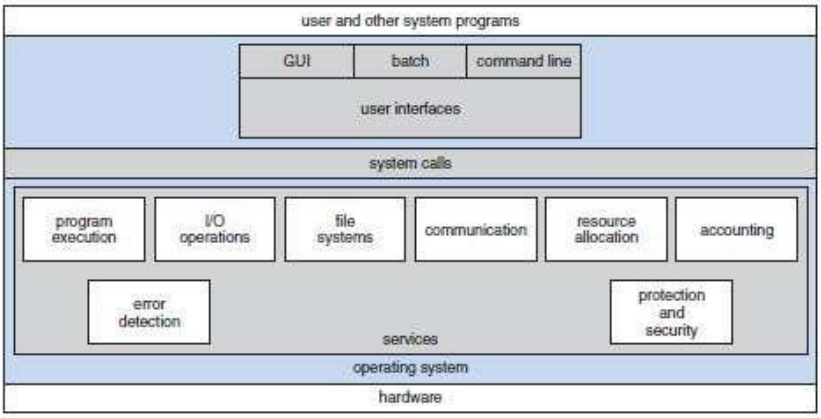


Figure 2.1 A view of OS services

- Common functions helpful to the user are (Figure 2.1):
  Almost all OS have a user-interface (UI).

  Different interfaces are:

  i) CLI (Command Line Interface) ¤ This uses
  → text commands and
  → method for entering the text commands.

ii) Batch Interface

¤ Commands & directives to control those commands are entered into files, and those files are executed.

iii) GUI (Graphical User Interface)

¤ The interface is a window-system with a pointing-device to
→ direct I/0
→ choose from menus and

→ make selections.

2) Program Execution

The system must be able to
→ load a program into memory and
→ run the program.

The program must be able to end its execution, either normally or abnormally.

3) I/O Operations

The OS must provide a means to do I/O operations because users cannot control I/O devices directly.

For efficiency and protection, users usually cannot control I/O devices directly. Therefore, the operating system must provide a means to do I/O. For specific devices, special functions may be desired (ex: to blank CRT screen).

4) File-System Manipulation

Programs need to

→ read & write files (or directories)

→ create & delete files
→ search for a given file and
→ allow or deny access to files.

5) Communications

In some situations, one process needs to communicate with another process.

Communications may be implemented via

1. Shared memory or
2. Message passing

In message passing, packets of information are moved between processes by OS.

6) Error Detection

Errors may occur in

→ CPU & memory-hardware (ex: power failure)
→ I/O devices (ex: lack of paper in the printer) and
→ user program (ex: arithmetic overflow)

For each type of error, OS should take appropriate action to ensure correct & consistent computing.

• Common functions for efficient operation of the system are:

- When multiple users are logged on the system at the same time, resources must be allocated to each of them.
- The OS manages different types of resources.
- Some resources (say CPU cycles) may have special allocation code.
    - Other resources (say I/O devices) may have general request & release code.
- We want to keep track of
- which users use how many resources and
- which kinds of resources.
- This record keeping may be used for
- accounting (so that users can be billed) or
- gathering usage-statistics.
- When several separate processes execute concurrently, it should not be possible for one process to interfere with the others or with the OS itself.
- Protection involves ensuring that all access to resources is controlled.
    - Security starts with each user having authenticated to the system by means of a password.

2.2 User Operating System Interface

• Two ways that users interface with the OS: 1) Command Interpreter (Command-line interface)

2) Graphical User Interface (GUI)

1) Command Interpreter

• Main function:

To get and execute the next user-specified command (Figure 1.13).

• The commands are used to manipulate files i.e. create, copy, print, execute, etc.

• Two general ways to implement:

1) Command interpreter itself contains code to execute command.
2) Commands are implemented through system programs. This is used by UNIX.

2) Graphical User Interfaces

• No entering of commands but the use of a mouse-based window and menu system (Figure 2.2).
• The mouse is used to move a pointer to the position of an icon that represents
→ file
→ program or
→ folder

• By clicking on the icon, the program is invoked

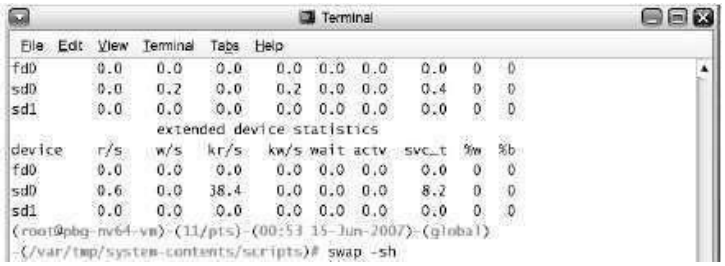


Figure 2.2 The Bourne shell command interpreter in Solaris 10.

2.3 System Calls

- These provide an interface to the OS services.
- These are available as routines written in C and C++.
- The programmers design programs according to an API. (API=application programming interface).
- The API
  → defines a set of functions that are available to the programmer (Figure 2.3).
  → includes the parameters passed to functions and the return values.
- The functions that make up an API invoke the actual system-calls on behalf of the programmer.
- **Benefits of API:**
  1) Program portability.
  2) Actual system-calls are more detailed (and difficult) to work with than the API available to the programmer.
- Three general methods are used to pass parameters to the OS:
  1) via registers.
  2) Using a table in memory & the address is passed as a parameter in a register (Figure 2.3).
  3) The use of a stack is also possible where parameters are pushed onto a stack and popped off the stack by the OS.

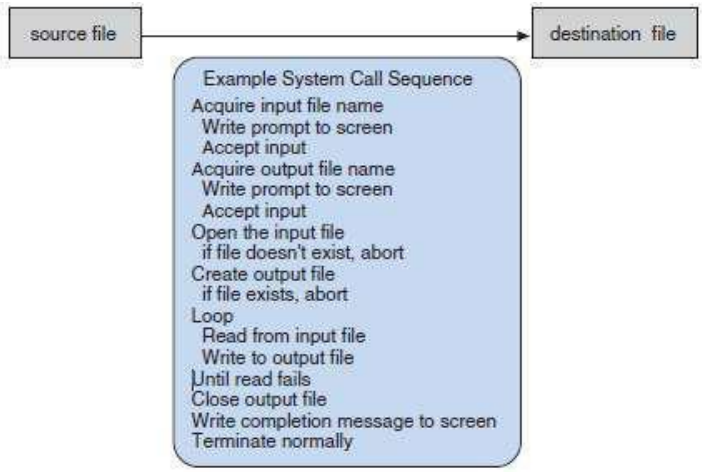


Figure 2.3 Example of how system calls are used.

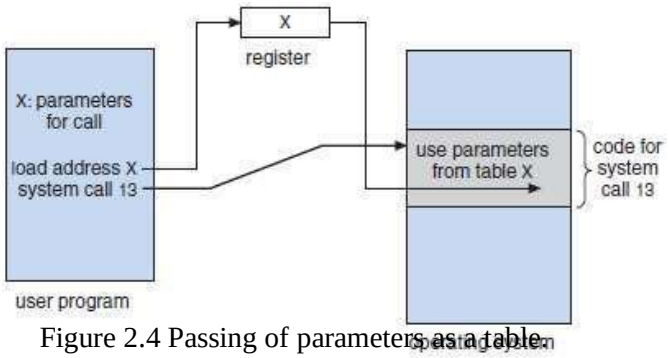


Figure 2.4 Passing of parameters as a table.

2.4 Types of System Calls

1) Process control
2) File management
3) Device management
4) Information maintenance
5) Communications

2.4.1 Process Control

- System calls used:
  - end, abort
  - load, execute
  - create process, terminate process
  - get process attributes, set process attributes
  - wait for time
  - wait event, signal event
  - allocate and free memory

- A running program needs to be able to halt its execution either normally (**end**) or abnormally (**abort**).

- If program runs into a problem, error message may be generated and dumped into a file.
  This file can be examined by a debugger to determine the cause of the problem.

- The OS must transfer control to the next invoking command interpreter.
  - Command interpreter then reads next command.
  - In interactive system, the command interpreter simply continues with next command.
  - In GUI system, a pop-up window will request action from user.
- A process executing one program can **load** and **execute** another program.
- Where to return control when the loaded program terminates?

1) If control returns to the existing program when the new program terminates, we must save the memory image of the existing program. (Thus, we have effectively created a mechanism for one program to call another program).

2) If both programs continue concurrently, we **create**d a new process to be multiprogrammed.

- We should be able to control the execution of a process. i.e. we should be able to determine and reset the attributes of a process such as:
  → job's priority or

  → maximum execution time

- We may also want to **terminate** process that we created if we find that it
  → is incorrect or

  → is no longer needed.

- We may need to **wait** for processes to finish their execution.

  We may want to wait for a specific event to occur.

- The processes should then signal when that event has occurred.

2.4.2 File Management

- System calls used:
    - create file, delete file
    - open, close
    - read, write, reposition
    - get file attributes, set file attributes

- Working procedure:
    1) We need to **create** and **delete** files.
    2) Once the file is created,
        → we need to **open** it and to use it.
        → we may also **read** or **write**.

    3) Finally, we need to **close** the file.
- We need to be able to
    → determine the values of file-attributes and

    → reset the file-attributes if necessary.

- File attributes include
    → file name

    → file type
    → protection codes and

    → accounting information.

2.4.3 Device Management

- System calls used:
    - request device, release device;
    - read, write, reposition;
    - get device attributes, set device attributes;
    - logically attach or detach devices.
- A program may need additional resources to execute.

- Additional resources may be
    → memory

    → tape drives or

    → files.

- If the resources are available, they can be granted, and control can be returned to the user program;

    If the resources are unavailable, the program may have to wait until sufficient resources are available.

- Files can be thought of as virtual devices. Thus, many of the system calls used for files are also used for devices.

- In multi-user environment,
    1) We must first **request** the device, to ensure exclusive use of it.
    2) After we are finished with the device, we must **release** it.

- Once the device has been requested (and allocated), we can **read** and **write** the device.

- Due to lot of similarity between I/O devices and files, OS (like UNIX) merges the two into a combined file-device structure**.**

• UNIX merges I/O devices and files into a combined file-device structure**.**

2.4.4 Information Maintenance

• System calls used:
- get time or date, set time or date
- get system data, set system data
- get process, file, or device attributes
- set process, file, or device attributes

• Many system calls exist simply for the purpose of transferring information between the user program and the OS.

For ex,

1) Most systems have a system call to return
→ current time and

→ current date.

2) Other system calls may return information about the system, such as
→ number of current users

→ version number of the OS
→ amount of free memory or disk space.

3) The OS keeps information about all its processes, and there are system calls to access this information.

2.4.5 Communication

• System calls used:
- create, delete communication connection
- send, receive messages
- transfer status information
- attach or detach remote devices

• Two models of communication.
1) Message-passing model and 2) Shared Memory Model

2.4.5.1 Message Passing Model

• Information is exchanged through an IPC provided by OS. (IPC=inter process communication).

• Steps for communication:
1) Firstly, a connection must be opened using **open connection** system-call.
2) Each computer has a host-name, such as an IP name.
Similarly, each process has a process-name, which is translated into an equivalent identifier.

The **get hostid** & **get processid** system-calls do this translation.

3) Then, identifiers are passed to the **open** and **close** system-calls.

4) The recipient-process must give its permission for communication to take place with an **accept connection** system-call.
(The processes that will be receiving connections are called daemons processes).

5) Daemon processes
→ execute a **wait for connection** system-call and

→ are awakened when a connection is made.

6) Then, client & server exchange messages by **read message** and **write message** system calls.
7) Finally, the **close connection** system-call terminates the communication.

- Advantages:
  1) Useful when smaller numbers of data need to be exchanged.
  2) It is also easier to implement than is shared memory.

2.4.5.2 Shared Memory Model

- Processes use map memory system-calls to gain access to regions of memory owned by other processes.
- Several processes exchange information by reading and writing data in the shared memory.
- The shared memory
  → is determined by the processes and
  → are not under the control of S.

- The processes are also responsible for ensuring that they are not writing to the same location simultaneously.
- Advantage:
  1) Shared memory allows maximum speed and convenience of communication,

- Disadvantage:
  1) Problems exist in the areas of protection and synchronization.

2.5 System Programs

- They provide a convenient environment for program development and execution. (System programs also known as system utilities).
- They can be divided into these categories:
- Six categories of system-programs:

1) File Management

 These programs manipulate files i.e. create, delete, copy, and rename files.

2) Status Information

 Some programs ask the system for

→ date (or time)
→ amount of memory(or disk space) or

→ no. of users.

→These information is then printed to the terminal (or output-device or file).

3) File Modification

→Text editors can be used to create and modify the content of files stored on disk.

4) Programming Language Support

→ Compilers, assemblers, and interpreters for common programming-languages (such as C, C++) are provided to the user.

5) Program Loading & Execution

The system may provide

→ absolute loaders
→ relocatable loaders
→ linkage editors and

→ overlay loaders.

→Debugging-systems are also needed.

→These programs are used for creating virtual connections between
→ processes

→ users and
→ computer-systems.

They allow users to
→ browse web-pages
→ send email or

→ log-in remotely.

- Most OSs are supplied with programs that
  → solve common problems or
  → perform common operations. Such programs include
  → web-browsers

  → word-processors
  → spreadsheets and

  → games.

  These programs are known as application programs**.**

2.6 Operating System Design & Implementation

2.6.1 Design Goals

- The first problem in designing a system is to
  → define goals and
  → define specifications.

- The design of the system will be affected by
  → choice of hardware and
  → type of system such as

  1) batch or time shared
  2) single user or multiuser

- Two basic groups of requirements: 1) User goals and

  2) System goals

1) User Goals

- The system should be
  → convenient to use

→ easy to learn and to use
→ reliable, safe, and fast.

2) System Goals

- The system should be
  → easy to design

  → implement, and maintain
  → flexible, reliable, error free, and efficient.

2.6.2 Mechanisms & Policies

- Mechanisms determine how to do something.
- Policies determine what will be done.
- For example, the timer construct is a mechanism for ensuring CPU protection, but deciding how long the timer is to be set for a particular user is a policy decision.
- Separating policy and mechanism is important for flexibility.
- Policies change over time; mechanisms should be general.
- Change in policy would then require redefinition of only certain parameters of the system. For instance, consider a mechanism for giving priority to certain types of programs over others
- If the mechanism is properly separated from policy, it can be used to support a policy decision that I/O-intensive programs should have priority over CPU-intensive ones or to support the opposite policy

•2.**6.3 Implementation**

- Traditionally, operating systems have been written in assembly language
- OS's are nowadays written in higher-level languages like C/C++
- The Linux and Windows XP operating systems are written mostly in C,
- Some small sections of assembly code for device drivers and for saving and restoring the state of registers
- Advantages of higher-level languages:

  1) Faster development and
  2) OS is easier to port.
- Disadvantages of higher-level languages:
  1) Reduced speed and

  2) Increased storage requirements.
- Major performance improvements are the result of better **data structures** and **algorithms** than of excellent assembly-language code.
- Although operating systems are large, only a small amount of the code is critical to **high performance**;
- Memory manager and the CPU scheduler are probably the most **critical routines**.
- After the system is written and is working correctly, **bottleneck** routines can be identified and can be replaced with **assembly-language equivalents**.

2.7 Operating System Structure

1) Simple Structure

2) Layered Approach
3) Micro-kernels

4) Modules

2.7.1 Simple Structure

- These OSs are small, simple, and limited system.
- For example: MS-DOS and UNIX.

- **MS-DOS** was written to provide the most functionality in the least space.
- **Disadvantages**:

i) It was not divided into modules carefully (Figure 2.5).
ii) The interfaces and levels of functionality are not well separated.

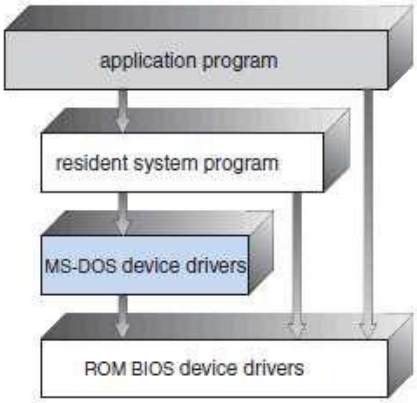


Figure 2.5 MS-DOS layer structure

- Application programs are able to access the basic I/O routines
- MS-DOS vulnerable to errant (or malicious) programs, causing entire system crashes when user programs fail.
- MS-DOS was also limited by the hardware of its era.
- The Intel 8088 for which it was written provides no dual mode and no hardware protection, the designers of MS-DOS had no choice but to leave the base hardware accessible.

2) **UNIX** was initially limited by hardware functionality.
Two parts of UNIX (Figure 2.6): Kernel and System programs.

- The kernel is further separated into a series of interfaces and device drivers.

- Everything below the system-call interface and above the physical hardware is the kernel.
- The kernel provides following functions through system calls:
  - file system
  - CPU scheduling and
  - memory management.

- **Disadvantage**:
  1) Difficult to enhance, as changes in one section badly affects other areas.

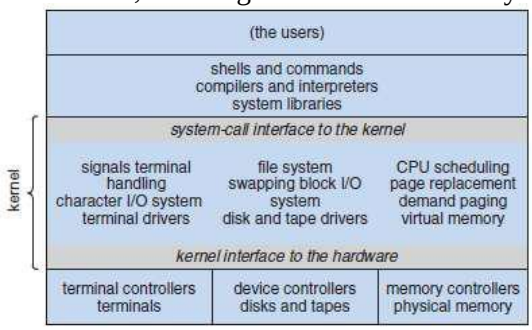


Figure 2.6 Traditional UNIX system structure

2.7.2 Layered Approach

- The OS is divided into a number of layers.
- Each layer is built on the top of another layer.
- The bottom layer is the hardware.
- A layer is an implementation of an abstract-object. i.e. The object is made up of
  → data and
  → operations that can manipulate the data.
- The layer consists of a set of routines that can be invoked by higher-layers.
- Higher-layer
  → does not need to know how lower-layer operations are implemented
  → needs to know only what lower-layer operations do.
- Advantage:
  1) Simplicity of construction and debugging.
- Disadvantages:
  1) Less efficient than other types.
  2) Appropriately defining the various layers.(„.“ a layer can use only lower-layers, careful planning is necessary).
- For example, the device driver for the backing store (disk space used by virtual-memory algorithms) must be at a lower level than the memory-management routines, because memory management requires the ability to use the backing store.

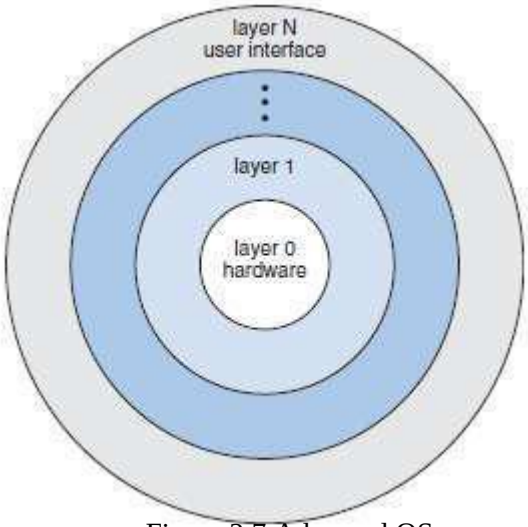


Figure 2.7 A layered OS

2.7.3 Micro-Kernels

- As UNIX expanded, the kernel became large and difficult to manage.
- In the mid-1980s, researchers at Carnegie Mellon University developed an operating system called **Mach** that modularized the kernel using the **microkernel** approach.
- This method structures the operating system by removing all nonessential components from the kernel and implementing them as system and user-level programs.
- The result is a **smaller kernel**.

- There is little **concern** regarding which services should remain in the kernel and which should be implemented in user space.
- Typically, however, microkernels provide minimal **process and memory management, in addition to a communication facility.**

Main function: To provide a communication facility between

→ client program and
→ various services running in user-space.

- Communication is provided by message passing (Figure 2.8).
- All non-essential components are
  → removed from the kernel and

  → implemented as system- & user-programs.

- Advantages:
  1) Ease of extending the OS. (New services are added to user space w/o modification of kernel).

  2) Easier to port from one hardware design to another.
  3) Provides more security & reliability.(If a service fails, rest of the OS remains untouched.).
  4) Provides minimal process and memory management.

- Disadvantage:
  1) Performance decreases due to increased system function overhead.

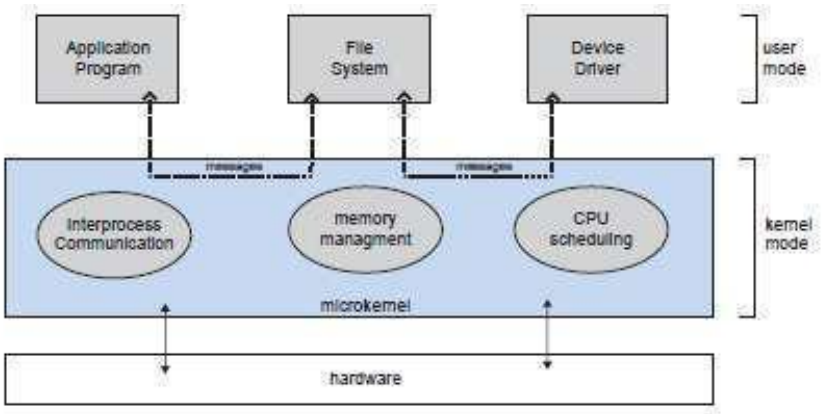


Figure 2.8 Architecture of a typical microkernel

2.7.4 Modules

- The kernel has
  → set of core components and
  → dynamic links in additional services during boot time(or run time).

- Seven types of modules in the kernel (Figure 2.9):

  1) Scheduling classes
  2) File systems
  3) Loadable system calls

  4) Executable formats
  5) STREAMS modules

6) Miscellaneous
7) Device and bus drivers

- The top layers include
  → application environments and
  → set of services providing a graphical interface to applications.

- Kernel environment consists primarily of
  → Mach microkernel and

  → BSD kernel.

- Mach provides
  → memory management;

  → support for RPCs & IPC and
  → thread scheduling.

- BSD component provides
  → BSD command line interface
  → support for networking and file systems and

  → implementation of POSIX APIs

- The kernel environment provides an I/O kit for development of
  → device drivers and

  → dynamic loadable modules (which Mac OS X refers to as kernel extensions).

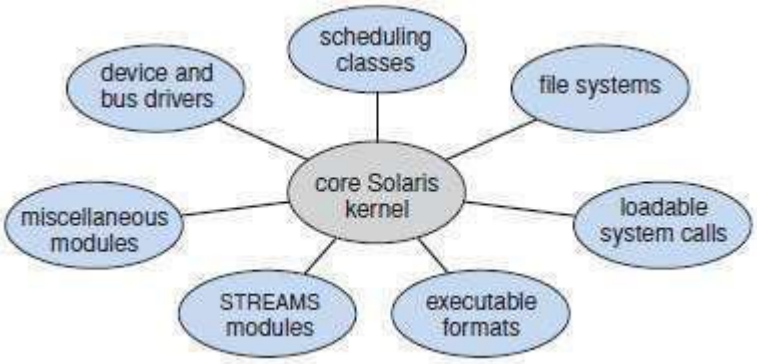


Figure 2.9 Solaris loadable modules

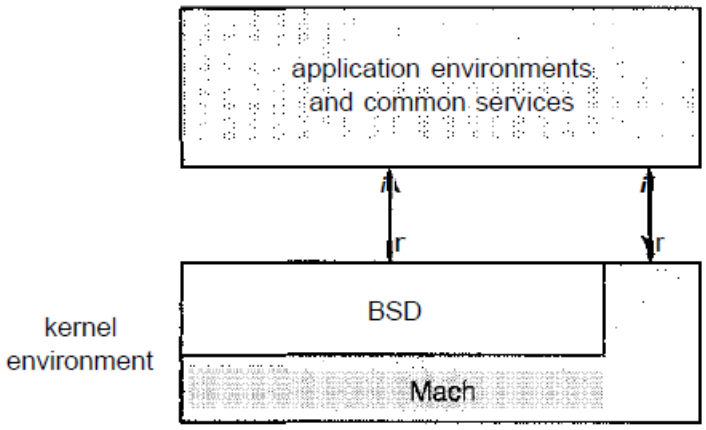


Figure 2.10 Mac OS X Structure

2.8 Virtual Machines

• **Main idea:** To abstract hardware of a single computer into several different execution environments. An OS creates the illusion that a process has

→ own processor &

→ own (virtual) memory.

• The virtual-machine provides
→ an interface that is identical to the underlying hardware (Figure 2.11).

→ a (virtual) copy of the underlying computer to each process.

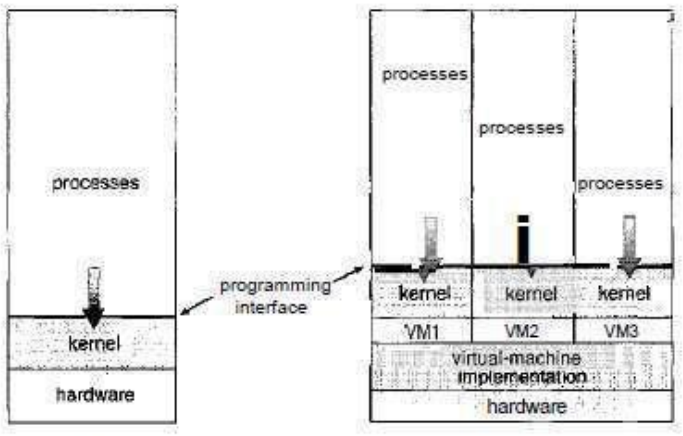


Figure 2.11 System models, (a) Nonvirtual machine, (b) Virtual machine.

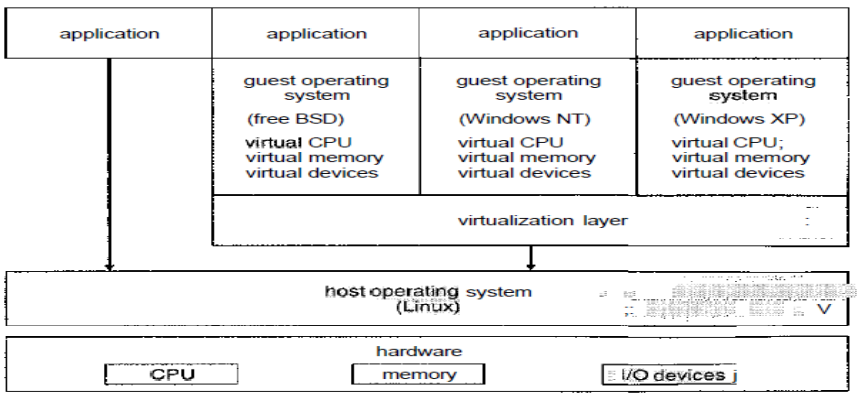
• Problem: Virtual-machine software itself will need substantial disk space to provide virtual memory. Solution: provide virtual disks that are identical in all respects except size.

• **Advantages**:
1) Complete protection of the various system resources. It is a perfect vehicle for OS"s R&D.

• **Disadvantage**: Difficult to implement due to effort required to provide an exact duplicate to underlying machine.

2.8.3.1 VMware Example Virtual Machine

VMware is a popular commercial application that abstracts Intel 80X86 hardware into isolated virtual machines. VMware runs as an application on a host operating system such as Windows or Linux and allows this host system to concurrently run several different **guest operating systems** as independent

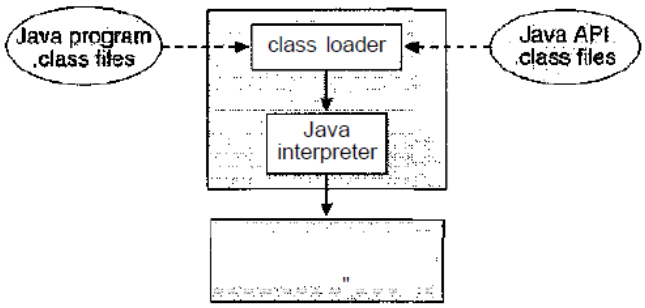


virtual machines.

Fig 2.12 VMWare Virtual Machine

2.8.3.2 The Java Virtual Machine *

Java is a popular object-oriented programming language introduced by Sun Microsystems in 1995. In addition to a language specification and a large API library, Java also provides a specification for a Java virtual machine—or JVM. Java objects are specified with the class construct; a Java program consists of one or more classes. For each Java class, the compiler produces an architecture-neutral **bytecode** output (.class) file that will run on any implementation of the JVM.



2.13 Java Virtual Machine

2.9 Operating System Generation

- It is possible to design, code, and implement an operating system specifically for one machine at one site.
- Operating systems are designed to run on any of a class of machines at a variety of sites with a variety of peripheral configurations.
- The system must then be configured or generated for each specific computer site, a process sometimes known as **system generation** (SYSGEN).
- SYSGEN program must determine:

1) What CPU will be used?
2) How will boot disk be formatted?
3) How much memory is available?
4) What devices are available?
5) What OS options are desired?

- A system-administrator can use the above information to modify a copy of the source code of the OS The operating system then is completely compiled.
- Data declarations, initializations, and constants, along with conditional compilation, produce an output object version of the operating system
- At the other extreme, it is possible to construct a system that is completely table driven.
- All the code is always part of the system, and selection occurs at execution time, rather than at compile or link time.
- System generation involves simply creating the appropriate tables to describe the system.

2.10 System Boot

- After an operating system is **generated**, it must be made **available** for use by the **hardware**.
- **But how does** the hardware know where the kernel is or how to load that kernel?
- The procedure of starting a computer by loading the kernel is known as ***booting*** the system.
- A small piece of code known as the **bootstrap program** or **bootstrap loader** locates the kernel, loads it into main memory, and starts its execution.
- This program is in the form of **read-only memory (ROM),** because the RAM is in an unknown state at system startup.

• ROM is convenient because it **needs no initialization and cannot be infected by a computer virus**.

• Some systems—such as cellular phones, PDAs, and game consoles—**store the entire operating system in ROM.**

• But changing the bootstrap code requires changing the ROM hardware chips.

• Some systems resolve this problem by using **erasable programmable read-only memory** (EPROM), which is readonly except when explicitly given a command to become writable.

• All forms of ROM are also known as **firmware,** since their characteristics fall somewhere between those of hardware and those of software.

• A problem with firmware in general is that executing code there is **slower** than executing code in RAM. Some systems store the operating system in firmware and **copy it to RAM for fast execution**.

• For **large operating systems** (including most general-purpose operating systems like Windows, Mac OS X, and UNIX) or for systems that change frequently, the bootstrap loader is stored in firmware, and the operating system is on disk.

MODULE 1 (CONT.): CHAPTER 3-PROCESSES

3.1 Process Concept

➢A process is the unit-of-work.

➢A system consists of a collection of processes:

1) **OS process** can execute system-code and
2) **User process** can execute user-code.

3.1.1 The Process

• A process is a program in execution.
• It also includes (Figure 1.23):

1) **Program Counter** to indicate the current activity.
2) **Registers Content** of the processor.
3) **Process Stack** contains temporary data.
4) **Data Section** contains global variables.
5) **Heap** is memory that is dynamically allocated during process run time.

• A program by itself is not a process.

1) A process is an active-entity.
2) A program is a passive-entity such as an executable-file stored on disk.

• A program becomes a process when an executable-file is loaded into memory.
• If you run many copies of a program, each is a separate process.
The text-sections are equivalent, but the data-sections vary.

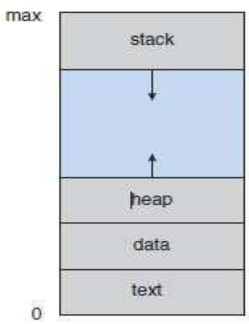


Figure 3.1 Process in memory

3.1.2 Process State

- As a process executes, it changes state.
- Each process may be in one of the following states (Figure 1.24):

1) **New**: The process is being created.
2) **Ready**: The process is waiting to be assigned to a processor.
3) **Running**: Instructions are being executed.
4) **Waiting:** The process is waiting for some event to occur (such as I/0 completions).
5) **Terminated**: The process has finished execution.

- Only one process can be running on any processor at any instant.

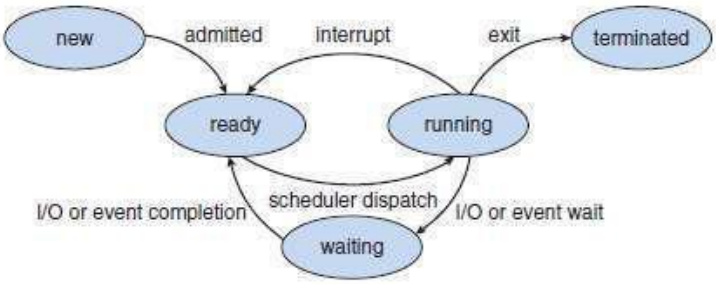


Figure 3.4 Diagram of process state

3.1.3 Process Control Block

• In OS, each process is represented by a PCB (Process Control Block).

Figure 3.5 Process control block (PCB)

- PCB contains following information about the process (Figure 3.5):
  - **Process State**

    The current state of process may be
    → new
    → ready
    → running
    → waiting or
    → halted.

    This indicates the address of the next instruction to be executed for the process.

3) CPU Registers

These include

→ accumulators (AX)
→ index registers (SI, DI)
→ stack pointers (SP) and
→ general-purpose registers (BX, CX, DX).

4) CPU Scheduling Information
This includes: priority of process

→ pointers to scheduling-queues and

→ scheduling-parameters.

5) Memory Management Information
This includes: value of base- & limit-registers and

→ value of page-tables(or segment-tables).

6) Accounting Information
→ This includes : amount of CPU time

→ time-limit and
→ process-number.

7) I/O Status Information

→ This includes: list of I/O devices

→ list of open files.

3.2 Process Scheduling

• **Objective of multiprogramming:** To have some process running at all times to maximize CPU utilization.

Objective of time-sharing: To switch the CPU between processes so frequently that users can interact with each program while it is running.

• To meet above 2 objectives: **Process scheduler** is used to select an available process for program-execution on the CPU.

3.2.1 Scheduling Queues

• Three types of scheduling-queues:

1) Job Queue : This consists of all processes in the system. As processes enter the system, they are put into a job-queue.

2) Ready Queue: This consists of the processes that are residing in main-memory and ready & waiting to execute (Figure 3.6).

→This queue is generally stored as a **linked list**.
→A ready-queue header contains pointers to the first and final PCBs in the list.
→Each PCB has a pointer to the next PCB in the ready-queue.
→This consists of the processes that are waiting for an I/O device.
→Each device has its own device-queue.

• When the process is executing, one of following events could occur (Figure 1.27):

1) The process could issue an I/0 request and then be placed in an I/0 queue.
2) The process could create a new subprocess and wait for the subprocess's termination.

3) The process could be interrupted and put back in the ready-queue.

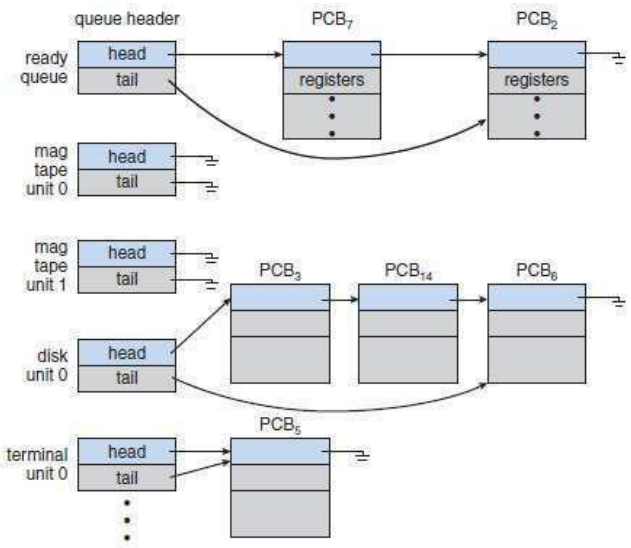


Figure 3.7 The ready-queue and various I/O device-queues

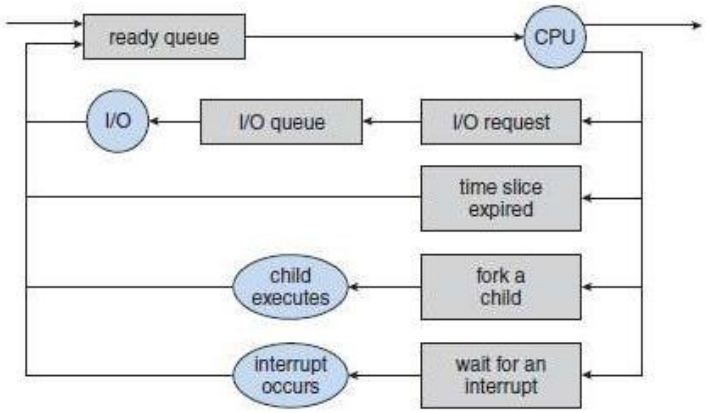


Figure 3.8Queueing-diagram representation of process scheduling

3.2.2 Schedulers

• Three types of schedulers:

1) Long-term scheduler
2) Short-term scheduler and
3) Medium-term schedulers

Long-Term Scheduler	Short-Term Scheduler
Also called job scheduler.	Also called CPU scheduler.
Selects which processes should be brought into the ready-queue.	Selects which process should be executed next and allocates CPU.
Need to be invoked only when a process leaves the system and therefore executes much less frequently.	Need to be invoked to select a new process for the CPU and therefore executes much more frequently.
May be slow „„ minutes may separate the creation of one new process and the next.	Must be fast „„ a process may execute for only a few milliseconds.
Controls the degree of multiprogramming.	

Processes can be described as either:

- **I/O-bound Process**
  - Spends more time doing I/O operation than doing computations.
  - Many short CPU bursts.
- **CPU bound processes**
  - Spends more time doing computations than doing I/O operation.
  - Few very long CPU bursts.

• Why long-term scheduler should select a good process mix of I/O-bound and CPU-bound processes ?
Ans: 1) If all processes are I/0 bound, then
i) Ready-queue will almost always be empty, and
ii) Short-term scheduler will have little to do.
2) If all processes are CPU bound, then
i) I/0 waiting queue will almost always be empty (devices will go unused) and
ii) System will be unbalanced.

• Some time-sharing systems have **medium-term scheduler** (Figure 3.9).
• The scheduler removes processes from memory and thus reduces the degree of multiprogramming.
• Later, the process can be reintroduced into memory, and its execution can be continued where it left off. This scheme is called **swapping**.

 The process is swapped out, and is later swapped in, by the scheduler.\

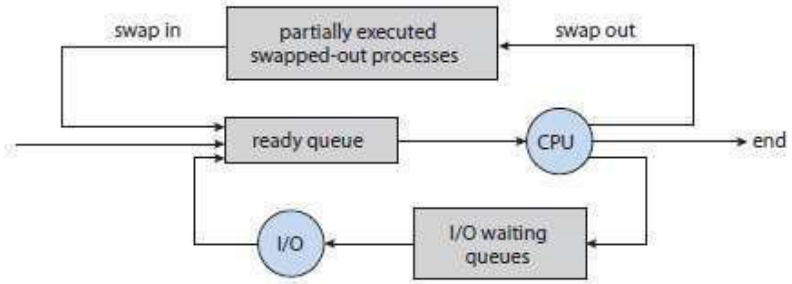


Figure 3.9 Addition of medium-term scheduling to the queueing diagram

3.2.3 Context Switch

- Context-switch means saving the state of the old process and switching the CPU to another process**.**
- The context of a process is represented in the PCB of the process; it includes
  → value of CPU registers

  → process-state and
  → memory-management information.

- Disadvantages:

  1) Context-switch time is pure overhead, because the system does no useful work while switching.

  2) Context-switch times are highly dependent on hardware support.

3.3 Operations on Processes

1) Process Creation and

2) Process Termination

3.3.1 Process Creation

- A process may create a new process via a **create-process** system-call.

- The creating process is called a **parent-process.**
- The new process created by the parent is called the **child-process** (Sub-process).

- OS identifies processes by pid (process identifier), which is typically an integer-number.
- A process needs following resources to accomplish the task:
  → CPU time
  → memory and

  → I/0 devices.

- Child-process may
  → get resources directly from the OS or

→ get resources of parent-process. This prevents any process from overloading the system.

- Two options exist when a process creates a new process:

  1) The parent & the children execute concurrently.
  2) The parent waits until all the children have terminated.

- Two options exist in terms of the address-space of the new process:

  1) The child-process is a duplicate of the parent-process (it has the same program and data as the parent).
  2) The child-process has a new program loaded into it.

- In UNIX, each process is identified by its process identifier (pid), which is a unique integer.

Example : In Solaris, the process at the top of the tree is the sched process, with pid of 0. The sched process creates several children processes—including pageout and f sf lush. These processes are responsible for managing memory and file systems. The sched process also creates the i n i t process, which serves as the root parent process for all user processes. In Figure 3.9, we see two children of i n i t — inetd and dtlogin. inetd is responsible for networking services such as t e l n e t and ftp; dtlogin is the process representing a user login screen. When a user logs in, dtlogin creates an X-windows session (Xsession), which in turns creates the sdt_shel process. Below sdt_shel, a user's command-line shell—the C-shell or csh—is created. It is this commandline interface where the user then invokes various child processes, such as the I s and cat commands. We also see a csh process with pid of 7778 representing a user who has logged onto the system using t e l n e t . This user has started the Netscape browser (pid of 7785) and the emacs editor (pid of 8105). On UNIX, a listing of processes can be obtained using the ps command.

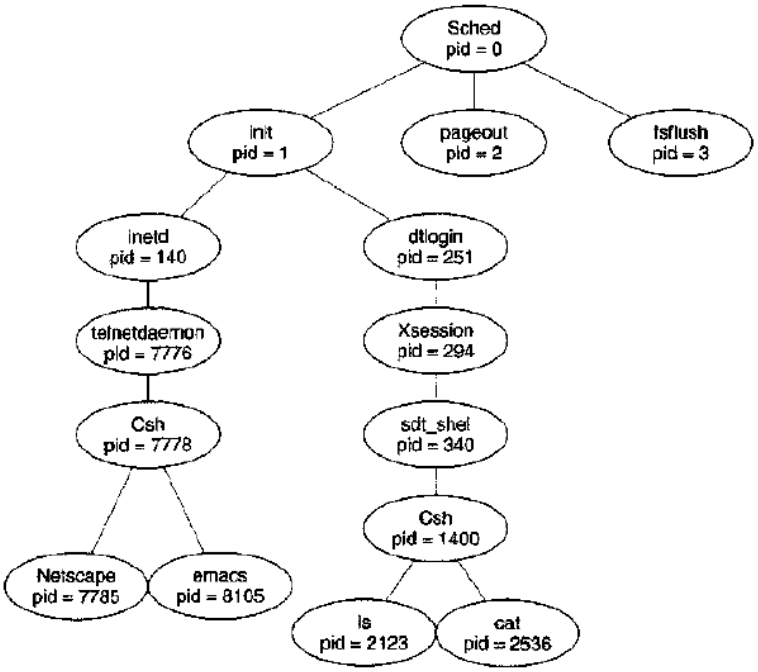


Figure 3.10 a Tree of processes on Solaris OS

- A new process is created by the **fork()** system-call (Figure 3.11 & 3.12).
- The new process consists of a copy of the address-space of the original process.
- Both the parent and the child continue execution with one difference:

  1) The return value for the fork() is **zero** for the new (child) process.
  2) The return value for the fork() is **nonzero** pid of the child for the parent-process.

- Typically, the **exec()** system-call is used after a fork() system-call by one of the two processes to replace the process's memory-space with a new program.
- The parent can issue **wait()** system-call to move itself off the ready-queue.

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
pid_t pid;

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
       fprintf(stderr, "Fork Failed");
       return 1;
    }
    else if (pid == 0) { /* child process */
       execlp("/bin/ls","ls",NULL);
    }
    else { /* parent process */
       /* parent will wait for the child to complete */
       wait(NULL);
       printf("Child Complete");
    }

    return 0;
}
```

Figure 3.10 Creating a separate process using the UNIX fork() system-call

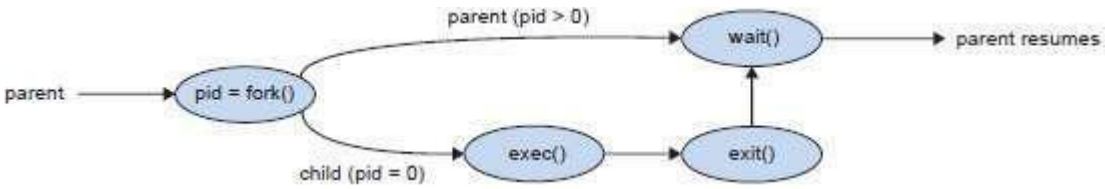


Figure 3.11 Process creation using the fork() system-call

```
#include <stdio.h>
#include <windows.h>

int main(VOID)
{
STARTUPINFO si;
PROCESS_INFORMATION pi;

    // allocate memory
    ZeroMemory(&si, sizeof(si));
    si.cb = sizeof(si);
    ZeroMemory(&pi, sizeof(pi));

    // create child process
    if (!CreateProcess(NULL, // use command line
     "C:\\WINDOWS\\system32\\mspaint.exe", // command line
     NULL, // don't inherit process handle
     NULL, // don't inherit thread handle
     FALSE, // disable handle inheritance
     0, //no creation flags
     NULL, // use parent's environment block
     NULL, // use parent's existing directory
     &si,
     &pi))
    {
      fprintf(stderr, "Create Process Failed");
      return -1;
    }
    // parent will wait for the child to complete
    WaitForSingleObject(pi.hProcess, INFINITE);
    printf("Child Complete");

    // close handles
    CloseHandle(pi.hProcess);
    CloseHandle(pi.hThread);
}
```

Fig 3.12 a Creating a separate process using WIN 32 API

3.3.2 Process Termination

- A process terminates when it executes the last statement (in the program).
- Then, the OS deletes the process by using **exit()** system-call.
- Then, the OS de-allocates all the resources of the process. The resources include
  → memory
  → open files and

  → I/0 buffers.

- Process termination can occur in following cases:
  → A process can cause the termination of another process via **TerminateProcess()** system-call.
  → Users could arbitrarily **kill** the processes.

- A parent terminates the execution of children for following reasons:

1) The child has exceeded its usage of some resources.
2) The task assigned to the child is no longer required.

3) The parent is exiting, and the OS does not allow a child to continue.

• In some systems, if a process terminates, then all its children must also be terminated. This phenomenon is referred to as **cascading termination**.

3.4 Inter Process Communication (IPC)

• Processes executing concurrently in the OS may be

1) Independent processes or 2) Co-operating processes.

1) A process is **independent** if
   i) The process cannot affect or be affected by the other processes.
   ii) The process does not share data with other processes.

2) A process is **co-operating** if
   i) The process can affect or be affected by the other processes.
   ii) The process shares data with other processes.

• **Several reasons for providing an environment that allows process**

1) cooperation**Information Sharing**

• Since many users may be interested in same piece of information (ex: shared file).

2) Computation Speedup

• We must break the task into subtasks.
• Each subtask should be executed in parallel with the other subtasks.
• The speed can be improved only if computer has multiple processing elements such as

→ CPUs or I/O channels.

3) Modularity

• Divide the system-functions into separate processes or threads.

4) Convenience

• An individual user may work on many tasks at the same time.
• For ex, a user may be editing, printing, and compiling in parallel.

• Two basic models of IPC (Figure 3.12): 1) Shared-memory and

2) Message passing.

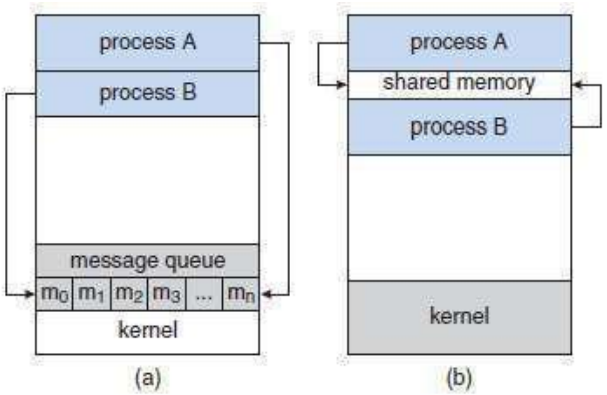


Figure 3.12 Communications models. (a) Message passing. (b) Shared-memory

3.4.1 Shared-Memory Systems

- Communicating-processes must establish a region of shared-memory.
- A shared-memory resides in address-space of the process creating the shared-memory. Other processes must attach their address-space to the shared-memory.
- The processes can then exchange information by reading and writing data in the shared-memory.
- The processes are also responsible for ensuring that they are not writing to the same location simultaneously.
- For ex, **Producer-Consumer Problem**:

Producer-process produces information that is consumed by a consumer-process

For example, a compiler may produce assembly code, which is consumed by an assembler. The assembler, in turn, may produce object modules, which are consumed by the loader.

- To allow producer and consumer processes to run concurrently, we must have available a buffer of items that can be filled by the producer and emptied by the consumer.
- This buffer will reside in a region of memory that is shared by the producer and consumer processes. A producer can produce one item while the consumer is consuming another item.
- The producer and consumer must be synchronized, so that the consumer does not try to consume an item that has not yet been produced.
- Two types of buffers can be used:
  1) **Unbounded-Buffer** places no practical limit on the size of the buffer.
  2) **Bounded-Buffer** assumes that there is a fixed buffer-size.

- The bounded buffer can be used to enable processes to share memory. The following variables reside in a region of memory shared by the producer and consumer processes:

```
#define BUFFER_SIZE 10

typedef struct {
    . . .
}item;

item buffer[BUFFER_SIZE];
int in = 0;
int out = 0;
```

Figure 3.13 Buffer Intialization

- The shared buffer is implemented as a circular array with two logical pointers: in and out. The variable in points to the next free position in the buffer; out points to the first full position in the buffer. The buffer is empty when in == out; the buffer is full when ((in + 1) % BUFFER_SIZE) == out.
- The producer process has a local variable nextProduced in which the new item to be produced is stored. The consumer process has a local variable nextConsumed in which the item to be consumed is stored.

```
item nextProduced;

while (true) {
     /* produce an item in nextProduced */
     while (((in + 1) % BUFFER-SIZE) == out)
        ; /* do nothing */
     buffer[in] = nextProduced;
     in = (in + 1) % BUFFER_SIZE;
}
```

Figure 3.14. Producer Process

```
item nextConsumed;

while (true) {
     while (in == out)
        ; //do nothing

     nextConsumed = buffer[out];
     out = (out + 1) % BUFFER_SIZE;
     /* consume the item in nextConsumed */
}
```

Figure 3.15 Consumer Process

when in == out; the buffer is full when ((in + 1) % BUFFER_SIZE) == out.

- Advantages:
  1) Allows maximum speed and convenience of communication.
  2) Faster.

3.4.2 Message-Passing Systems

- These allow processes to communicate and to synchronize their actions without sharing the same address-space.
- For example, a chat program used on the WWW.
- Messages can be of 2 types: Fixed size or Variable size.

1) If **fixed-sized messages** are used, the system-level implementation is simple. However, the programming task becomes more difficult.

2) If **variable-sized messages** are used, the system-level implementation is complex. However, the programming task becomes simpler.

- A communication-link must exist between processes to communicate
- Three methods for implementing a link:
  1) Direct or indirect communication.

  2) Symmetric or asymmetric communication.
  3) Automatic or explicit buffering.

Issues related to Inter Process Communication

3.4.2.1 Naming

- Processes that want to communicate must have a way to refer to each other. They can use either direct or indirect communication.
- Under direct communication, each process that wants to communicate must explicitly name the recipient or sender of the communication. In this scheme the send and receive primitives are defined as:
  - scheme, the send.0 and receive() primitives are defined as:

send(P,message): Send a message to process P.
receive(Q,message): Receive a message from process Q.

Direct Communication	Indirect Communication
Each process must explicitly name the recipient/sender.	Messages are sent to/received from mailboxes (or ports).
Properties of a communication link: • A link is established automatically between every pair of processes that want to communicate. The processes need to know only each other‟s identity to communicate. • A link is associated with exactly two processes. • Exactly one link exists between each pair of processes.	Properties of a communication link: • A link is established between a pair of processes only if both members have a shared mailbox. • A link may be associated with more than two processes. • A number of different links may exist between each pair of communicating processes.
Symmetric addressing: • Both sender and receiver processes must name the other to communicate Send(P,Message), Receive(Q,message)	**Mailbox owned by a process:** • The owner can only receive, and the user can only send. • The mailbox disappears when its owner process terminates.
Asymmetric addressing: • Only the sender names the recipient; the recipient needn't name the sender.Send(P,message), Receive(id,message)	**Mailbox owned by the OS:** The OS allows a process to: 1. Create a new mailbox 2. Send & receive messages via it 3. Delete a mailbox.

- Advantages:
  Useful for exchanging smaller amounts of data („.‟ No conflicts need be avoided).
  Easier to implement.
  Useful in a distributed environment.

3.4.2.2 Synchronization

• Message passing may be either blocking or non-blocking (also known as synchronous and asynchronous).

Synchronous Message Passing	Asynchronous Message Passing
Blocking send: The sending process is blocked until the message is received by the receiving process or by the mailbox.	Non-blocking send: The sending process sends the message and resumes operation.
Blocking receive: The receiver blocks until a message is available.	Non-blocking receive: The receiver retrieves either a valid message or a null.

3.4.2.3 Buffering

- Messages exchanged by processes reside in a temporary queue.
- Three ways to implement a queue:

1) Zero Capacity

- The queue-length is zero.
- The link can't have any messages waiting in it.
- The sender must block until the recipient receives the message.

2) Bounded Capacity

- The queue-length is finite.
- If the queue is not full, the new message is placed in the queue.
- The link capacity is finite.
- If the link is full, the sender must block until space is available in the queue.

3) Unbounded Capacity

- The queue-length is potentially infinite.
- Any number of messages can wait in the queue.
    - The sender never blocks.