K.S.INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COURSE FILE

NAME OF THE STAFF	:	**Dr. Rekha B Venkatapur**
SUBJECT CODE/NAME	:	**18CS56/Unix Programming**
SEMESTER/YEAR	:	**V/III**
ACADEMIC YEAR	:	**2020 – 2021**
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
Bangalura -560 109

K. S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-5600109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course:UNIX PROGRAMMING				
Type: Core	**Course Code:18CS56**	**Sem:5th**	**Sec: B**	**Year: III**
Academic Year: 2020-2021				
Faculty Name: Dr Rekha B Venkatapur				

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: The course objective is to make students familiarize with UNIX operating system features, shell programming and system programming.

Course Learning Outcomes:

After completing the course, the students will be able to,

18CS56.1	**Identify** the UNIX features, architecture, structure and organization of UNIX file system.	Applying (K3)
18CS56.2	**Construct** the regular expression for grep commands and implement shell programs.	Applying (K3)
18CS56.3	**Develop** system programs using different categories of API's.	Applying (K3)
18CS56.4	**Build** Interprocess communication using various techniques.	Applying (K3)
18CS56.5	**Utilize** POSIX API for implementing signals.	Applying (K3)

Syllabus Content	
Module1: **Introduction:** Unix Components/Architecture. Features of Unix. The UNIX Environmentand UNIX Structure, Posix and Single Unix specification. General features of Unixcommands/ command structure. Command arguments and options. Basic Unix commandssuch as echo, printf, ls, who, date,passwd, cal, Combining commands. Meaning of Internaland external commands. The type command: knowing the type of a command and locating it.The root login. Becoming the super user: su command. **Unix files:** Naming files. Basic file types/categories. Organization of files. Hidden files.Standard directories. Parent child relationship. The home directory and the HOME variable.Reaching required files- the PATH variable, manipulating the PATH, Relative and absolutepathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and doubledots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. LO: At the end of this session the student will be able to, 1. Understand architecture, features and structure of UNIX operating system. 2. Know the categories and organization of files. 3. Learn the file, directory and environment variables.	CO1 **10hrs** PO1-3 PO2-1 PO3-1 PO5- 2 PO9-2 PSO1-2
Module 2: **File attributes and permissions:** The ls command with options. Changing file permissions:the relative and absolute permissions changing methods. Recursively changing filepermissions. Directory permissions. **The shells interpretive cycle:** Wild cards. Removing the special meanings of wild cards.Three standard files and redirection. **Connecting commands:** Pipe. Basic and Extendedregular expressions. The grep, egrep. Typical examples involving different regularexpressions. **Shell programming:** Ordinary and environment variables. The .profile. Read and readonlycommands. Command line arguments. exit and exit status of a command. Logical operatorsfor conditional execution. The test command and its shortcut. The if, while, for and casecontrol statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples. LO: At the end of this session the student will be able to, 1. Understand the file attributes and permissions. 2. Know the use of regular expression in grep, egrep commands. 3. Learn the shell programming.	CO2 **10 hrs.** PO1-3 PO2-2 PO3-2 PO9-2 PSO1-2 PSO2-2
Module 3: **UNIX File APIs:** General File APIs, File and Record Locking, Directory File APIs, DeviceFile APIs, FIFO File APIs, Symbolic Link File APIs. **UNIX Processes and Process Control:** **The Environment of a UNIX Process:** Introduction, main function, Process Termination,Command-Line Arguments, Environment List, Memory Layout of a C Program, SharedLibraries, Memory Allocation, Environment Variables, setjmp and	CO3 **10 hrs** PO1-3

longjmp Functions,getrlimit, setrlimit Functions, UNIX Kernel Support for Processes. **Process Control:** Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions. LO: At the end of this session the student will be able to, 1. Understand all categories of UNIX file API's. 2. Learn the API related to UNIX process.	PO2-2 PO3-2 PO5-2 PO9-2 PSO1-2 PSO2-2
Module 4: Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting,User Identification, Process Times, I/O Redirection. **Overview of IPC Methods**, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System VIPC, Message Queues, Semaphores. **Shared Memory**, Client-Server Properties, Stream Pipes, Passing File Descriptors, An OpenServer-Version 1, Client-Server Connection Functions. LO: At the end of this session the student will be able to, 1. Learn the different types Interprocess communication techniques. 2. Understand the shared memory, message queues and semaphore method used for IPC.	CO4 **10 hrs** PO1-3 PO2-2 PO3-2 PO5-2 PSO1-2 PSO2-2
Module 5: **Signals and Daemon Processes:** Signals: The UNIX Kernel Support for Signals, signal,Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp andsiglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers. Daemon Processes:Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model. LO: At the end of this session the student will be able to, 1. Learn the API used for implementing signals. 2. Understand daemon process and its characteristics.	**CO5** **10** hrs PO1-3 PO2-1 PO3-1 PSO1-2 PSO2-2
Text Books: - 1. Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill (Chapter 1,2,3,4,5,6,8,13,14) 2. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005 (Chapter 3,7,8,10,13,15) 3. Unix System Programming Using C++ - Terrence Chan, PHI, 1999. (Chapter 7,8,9,10)	
Reference Books: 1. M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education. 2. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2ndEdition, Wiley,2014learning	
Useful Websites: https://nptel.ac.in/content/storage2/courses/106108101/pdf/PPTs/Mod_13.pdf http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html	

Useful Journals

- https://link.springer.com/book/10.1007/978-3-319-92429-8

Teaching and Learning Methods:

1. Lecture class: 48 hrs.
2. Self-study: --
3. Field visits/Group Discussions/Seminars: --02
4. Practical classes: --

Assessment:

Type of test/examination: Written examination

Continuous Internal Evaluation(CIE) : 40 marks (Average of three tests will be considered)
Test duration: 1 :30 hr

Semester End Exam(SEE) : 60 marks (students have to answer all main questions)
Examination duration: 3 hrs

CO to PO Mapping

PO1: Science and engineering Knowledge **PO2:** Problem Analysis **PO3:** Design & Development **PO4:**Investigations of Complex Problems **PO5:** Modern Tool Usage **PO6:** Engineer & Society	**PO7:**Environment and Society **PO8:**Ethics **PO9:**Individual & Team Work **PO10:** Communication **PO11:**Project Mgmt. & 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 analyse data to address real world challenges.

CO 18CS56	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
18CS56.1	3	1	1	-	2	-	-	-	2	-	-	-
18CS56.2	3	2	2	-	2	-	-	-	2	-	-	-
18CS56.3	3	2	2	-	2	-	-	-	2	-	-	-
18CS56.4	3	2	2	-	2	-	-	-	2	-	-	-
18CS56.5	3	1	1	-	-	-		-	-	-	-	-
18CS56	3	1.6	1.6	-	2	-	-	-	2	-	--	-

CO	PSO1	PSO2
18CS56.1	2	-
18CS56.2	2	2
18CS56.3	2	2
18CS56.4	2	2
18CS56.5	2	2
18CS56	2	2

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

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, BENGALURU-560109

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

SESSION: SEP 2020 - DEC 2020

Week No.	Month	Day						Days	Activities
		Mon	Tue	Wed	Thu	Fri	Sat		
1	SEP		1*	2	3	4	5	5	1*-Commencement of Higher Semester
2	SEP	7	8	9	10	11	12	6	
3	SEP	14	15	16	17 DH	18	19	5	17- Mahalaya Amavasya
4	SEP	21	22	23	24	25	26TA	6	
5	SEP / OCT	**28 T1**	**29 T1**	**30 T1**	1	2 DH	3	5	2- Mahatma Gandhi Jayanthi
6	OCT	5	6BV	7ASD	8	9	10	6	5-10 First Feed Back
7	OCT	12	13	14	15	16	17 H	6	
8	OCT	19	20	21	22	23	24	6	24 - Monday Time Table
9	OCT	26 DH	27	28	29TA	30 DH	31 DH	3	26- Vijayadashami 30- Eid-Milad 31- Maharishi Valmiki Jayanti
10	NOV	2	3	4	5	6	7	6	7 - Wednesday Time Table
11	NOV	**9 T2**	**10 T2**	**11 T2**	12	13	14 H	6	
12	NOV	16 DH	17 BV	18 ASD	19	20	21	5	16 - Balipadyami Deepavalli 18 - 21 Second Feed Back 21 - Friday Time Table
13	NOV	23	24	25	26	27	28 H	6	
14	NOV /DEC	30	1	2	3 DH	4	5TA	5	3- Kanakadasa Jayanti 5 - Monday Time Table
15	DEC	7	8	**9 LT**	**10 LT**	**11 LT**	12 H	6	
16	DEC	**14 T3**	**15 T3**	**16 T3**	17 * BV			4	17* -Last Working Day
Total No of Working Days : 82									

Total Number of working days (Excluding holidays and Tests)=70

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

Monday	12
Tuesday	13
Wednesday	13
Thursday	13
Friday	13
Saturday	6
Total	70

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

K. S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CALENDAR OF EVENTS: ODD SEMESTER (2020-2021)

SESSION: SEP 2020 - JAN 2021

Week No.	Month	Day						Days	Activities	Department Activities
		Mon	Tue	Wed	Thu	Fri	Sat			
1	SEP		1*	2	3	4	5	5	1*-Commencement of Higher Semester	
2	SEP	7	8	9	10	11	12	6		
3	SEP	14	15	16	17 H	18	19	5	17- Mahalaya	
4	SEP	21	22	23	24	25	26TA	6		
5	SEP / OCT	28 T1	29 T1	30 T1	1	2 H	3	5	2- Mahatma Gandhi Jayanthi	
6	OCT	5	6BV	7ASD	8	9	10	6	5-10 First Feed Back	
7	OCT	12	13	14	15	16	17 DH	5		
8	OCT	19	20	21	22	23	24	6	24 - Monday Time Table	
9	OCT	26 H	27	28	29	30 H	31 H	3	26- Vijayadashami 30- Eid-Milad 31- Maharishi Valmiki Jayanti	Project Zeroth Review Presentation
10	NOV	2	3	4	5	6	7 TA	6	7 - Wednesday Time Table	
11	NOV	9	10	11	12	13	14 DH	5		13-11-2020 Webinar on An Insight into Web Application Development
12	NOV	16 H	17 T2	18 T2	19 T2	20	21	5	16 - Balipadyami Deepavalli 18 - 21 Second Feed Back 21 - Friday Time Table	
13	NOV	23	24BV	25ASD	26	27	28 DH	5		
14	NOV /DEC	30	1	2	3 H	4	5	5	3- Kanakadasa Jayanti 5 - Monday Time Table	Project Zeroth Phase Re-Presentation
15	DEC	7	8	9	10	11	12 DH	5		
16	DEC	14	15	16	17	18	19	6	19- Monday Time	
17	DEC	21	22	23	24	25H	26DH	4	25-Christmas	22-12-2020 & 23-12-2020 34th CSI Karnataka State Student Convention
18	DEC/ JAN	28	29	30	31	1 TA	2	6	2Thursday Time Table	Project Phase - 1 Review Presentation
19	JAN	4 LT	5 LT	6 LT	7 LT	8	9 DH	5		
20	JAN	11 T3	12 T3	13 T3	14 H	15	16 *	5	14- Makara sankaranthi	
Total No of Working Days : 106										

Total Number of working days (Excluding holidays and Tests)=87

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

Monday	16
Tuesday	16
Wednesday	16
Thursday	16
Friday	17
Saturday	6
Total	**87**

Nuccarapu
24/8/2020

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

V SEM B SECTION STUDENT DETAILS

Sl. No	USN	Student Name	Email ID	Mobile Number
1	1KS18CS063	NITISH KUMAR.M.R		
2	1KS18CS064	NOOR SUMAIYA	nlight110@gmail.com	8861130383
3	1KS18CS065	P SAI RAM		
4	1KS18CS066	PAVAN P		
5	1KS18CS067	POOJASHREE K	poojakshree@gmail.com	9590962135
6	1KS18CS069	PRANAV M S	pranavmsofficial@gmail.com	6361563076
7	1KS18CS070	PRATEEK HAVALE	prateekhavale27@gmail.com	7353590170
8	1KS18CS071	PRAVEEN KUMAR K		
9	1KS18CS072	PREETHI K	preethisuma9@gmail.com	9591607795
10	1KS18CS073	PUJARI VISHNU PRIYA	pujarivishnupriya2000@gmail.com	9380026399
11	1KS18CS074	PULLUR PAVAN KUMAR	pavankumar.pullur@gmail.com	9573588937
12	1KS18CS075	R DEKSHITHA	dekshi17@gmail.com	9901931905
13	1KS18CS076	R PRATIKSHA	r.pratiksha1713@gmail.com	9739558476
14	1KS18CS077	RAMYA R	ramyareddy9488@gmail.com	9480903288
15	1KS18CS078	RAHUL.P	rahulgowdarg1@gmail.com	9739601531
16	1KS18CS079	RAIPALLE SHREYAA	shreyaraipalle@gmail.com	9916087673
17	1KS18CS080	RAKSHA S	rakshakaranth.s@gmail.com	8792491252
18	1KS18CS081	RAKSHITH KUMAR.N	rakshithkumar66666@gmail.com	9482423818
19	1KS18CS082	REKHA N C	rekhancsagar107@gmail.com	6361823624
20	1KS18CS083	RITHANA.N.RAJ	rithananraj@gmail.com	7892428272
21	1KS18CS084	RUBA ABDUL RAHMAN	rubaabdulrahman456@gmail.com	8861189361
22	1KS18CS086	SAMHITHA	samhithav1999@gmail.com	8861399452
23	1KS18CS087	SANDEEP KUMAR	sandeepkr2909@gmail.com	7004481701
24	1KS18CS088	SAURAV KUMAR	sauravkumar8k@gmail.com	8877648894
25	1KS18CS089	SAURAV S MAKAM	sauravsmakam3010@gmail.com	8762171300
26	1KS18CS090	SHALINI S	shalinis24102000@gmail.com	8884686261
27	1KS18CS091	SHASHANK G	shashankganesh689@gmail.com	9148452122
28	1KS18CS092	SHASHANK MISHRA	shashankmishra.0598@gmail.com	8808464930
29	1KS18CS093	SHIVANGI SRIVASTAVA	shivi.aol16@gmail.com	9108320847
30	1KS18CS094	SHIVA PRAKASH T	shivaprakash832000@gmail.com	7795289234
31	1KS18CS095	SHUBHASHINI.R	shubhashinir05204@gmail.com	9606856231

32	1KS18CS096	SINDU A S	0908.sindu@gmail.com	8792983854
33	1KS18CS097	SOURABH SANTOSH KAMB	sourabhsk112@gmail.com	9741997486
34	1KS18CS098	SRI CHANDANA P	srichandana1020@gmail.com	7795233667
35	1KS18CS099	SRIVIDYA H R	srividyahr21@gmail.com	9980795846
36	1KS18CS100	SUBRAMANYA N	subramanyagowda8123@gmail.com	8123724341
37	1KS18CS101	SUDHAKAR YASWANTH	sudhakaryaswanth001@gmail.com	9597197465
38	1KS18CS102	SUDHANSHU JOSHI	sudhanshujoshi019@gmail.com	9945059849
39	1KS18CS103	SUJAY G S	sujay.suresh28@gmail.com	9380074049
40	1KS18CS104	SUNAINA NAYAK	nayaksunaina88@gmail.com	9916564268
41	1KS18CS105	SURAJ C JAWOOR	Jawoorsuraj@gmail.com	9606647157
42	1KS18CS106	SUSHMITHA S	sushmithas.bdvt@gmail.com	8296679369
43	1KS18CS107	SWETHA BIJANAPALLI	swethabijanapalli@gmail.com	8971678175
44	1KS18CS108	THAMMINENI HEMANTH C	hemanthchowdary743@gmail.com	7702006368
45	1KS18CS109	THIRUMALAI SHAKTIVEL C	thirumalaishaktivel@gmail.com	9663841156
46	1KS18CS110	VAISHAK P	vaishak.puttaswamy@gmail.com	8296087114
47	1KS18CS111	VARIDHI MADHURANATH	varidhim08@gmail.com	7760847056
48	1KS18CS112	VEDAVEDYA B H	vedavedyabh@gmail.com	9611672575
49	1KS18CS113	VEERA SREENIDHI.R	veerasreenidhi@gmail.com	8310043988
50	1KS18CS114	VENKATESH M N	venkateshmnvenki@gmail.com	6366013511
51	1KS18CS115	VIJAY.N.S	vijaysingh13091999@gmail.com	8095553691
52	1KS18CS116	VIJAYASHREE.N.R	123vijayashree@gmail.com	8310637251
53	1KS18CS117	VIJETHA	vijetha.k.byndoor@gmail.com	9380797561
54	1KS18CS118	VIINOD H MALALI	vinodmalali2000@gmail.com	6362131012
55	1KS18CS119	VISHNUPRIYA D	dvishnupriya1112@gmail.com	7259619113
56	1KS18CS120	VYJAYANTHI K S	vyju72000@gmail.com	7019745430
57	1KS18CS121	YASHWANTH.K	yashwanthk273@gmail.com	9663688426
58	1KS18CS122	YOGITA RAIKAR	yogitaramesh11@gmail.com	7892492206
59	1KS18CS123	ZAINA KHAN	Khanzaina307@gmail.com	9916300823
60	1KS18CS124	SHEWANI CHIB	shewanichib01@gmail.com	9149890467
61	1KS17CS015	B R GAGAN	gaganravi1104@gmail.com	7019128572
62	1KS18CS125	R SOUMYA	soumyahegde984@gmail.com	8762891459
63	1KS18CS126	RAYYAAN MOHIADDIN	rayyaan.m786@gmail.com	9880088171
64	1KS18CS127	ARVIND PATHAK	arvindpathak96445@gmail.com	7830167597
65	1KS18CS128	BHAGYASHREE.V	bhagyamohan3@gmail.com	7795198289
66	1KS18CS129	BI BI AYESHA	aishakulsum0404@gmail.com	7348816338
67	1KS18CS130	LIKITHA.S	likitha6065@gmail.com	9060747570
68	1KS18CS131	SHALINI.K.P	shalinikp96@gmail.com	7619391256

69	1KS19CS401	ARPITHA.G	arpithagopal505@gmail.com	9110824612
70	1KS19CS404	BHAVYASHREE.R	Bhavyajanu77@gmail.com	9535605609
71	1KS19CS413	SAHANA.V	sahanav5940@gmail.com	8050023594
72	1KS19CS414	Y.MRUDULA JAIN	mruduyjain@gmail.com	9632095864

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU-109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIVIDUAL ONLINE TIME TABLE FOR THE YEAR 2020-21 (ODD SEMESTER)

W.E.F: 01-09-2020

NAME OF THE FACULTY: Dr. REKHA.B.VENKATAPUR **DESIGNATION: PROF. & HOD**

PERIOD	1	2	11:00 AM-11.30 AM	3	4		5	6
TIME / DAY	9:00 AM-10:00 AM	10:00 AM-11.00 AM		11:30 AM-12:30 PM	12:30 PM-1.30 PM	1:30 PM-2:00 PM	02:00 PM - 03:00 PM	03:00 PM- 04:00 PM
MON			TEA BREAK		UP(B)	LUNCH BREAK	ML LAB(A1,A2&A3)	
TUE		UP(B)						
WED	UP(B)						PROJECT PHASE 1 + SEMINAR	
THUR				UP(B)			PROJECT PHASE 1 + SEMINAR	
FRI								

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

V SEMESTER ONLINE CLASS TIME TABLE FOR THE YEAR 2020-21 (ODD SEMESTER)(TENTATIVE)

W.E.F: 01-09-2020 **CLASS TEACHER: Mr. Kumar K**

SEC: 'B'

PERIOD	1	2	11:00 AM-11.30 AM	3	4			6
TIME DAY	9:00 AM-10:00 AM	10:00 AM-11.00 AM		11:30 AM-12:30 PM	12:30 PM-1.30 PM	1:30 PM-2:00 PM	2:00 PM - 3:00 PM	03:00 PM- 04:00 PM
MON	PADP	DBMS	BREAK	ME	UP	LUNCH BREAK	TUTORIAL / PEDAGOGY ACTIVITIES	
TUE	ATC	UP		DBMS	CN		ES	
WED	UP	ME		PADP	CN		PLACEMENT ACTIVITIES ← →	
THUR	CN	PADP		UP	ATC		CN LAB ← →	
FRI	DBMS	ME		CN	ATC		DBMS LAB ← →	
SAT	ME	DBMS		ATC	PADP		TUTORIAL / PEDAGOGY ACTIVITIES	

Subject Code	Subject Name	Faculty Name
18CS51	MANAGEMENT AND ENTREPRENEURSHIP FOR IT INDUSTRY	Mr. Krishna Gudi
18CS52	COMPUTER NETWORKS	Dr. Ram P Rustagi
18CS53	DATABASE MANAGEMENT SYSTEM	Mr. Kumar K
18CS54	AUTOMATA THEORY AND COMPUTABILITY	Mr. Venkata Rao K
18CS55	APPLICATION DEVELOPMENT USING PYTHON	Mr. Prashanth H S
18CS56	UNIX PROGRAMMING	Dr. Rekha B Venkatapur
18CSL57	COMPUTER NETWORK LABORATORY	Mr. Krishna Gudi & Mrs. Swathi K
18CSL58	DBMS LABORATORY WITH MINI PROJECT	Mr. Kumar K & Dr. Dayananda R B
18CIV59	ENVIRONMENTAL STUDIES	Mrs. Radhika

BN 25/8/2020
TIME TABLE INCHARGE

HOD 25/8/2020

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

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

UNIX PROGRAMMING (Effective from the academic year 2018 -2019) SEMESTER – V			
Course Code	**18CS56**	**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 (18CS56) will enable students to

- Interpret the features of UNIX and basic commands.
- Demonstrate different UNIX files and permissions
- Implement shell programs.
- Explain UNIX process, IPC and signals.

Module 1	**Contact Hours**
Introduction: Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. General features of Unix commands/ command structure. Command arguments and options. Basic Unix commands such as echo, printf, ls, who, date,passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The root login. Becoming the super user: su command. **Unix files:** Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. **RBT: L1, L2**	08
Module 2	
File attributes and permissions: The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions. **The shells interpretive cycle:** Wild cards. Removing the special meanings of wild cards. Three standard files and redirection. **Connecting commands:** Pipe. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions. **Shell programming:** Ordinary and environment variables. The .profile. Read and readonly commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples. **RBT: L1, L2**	08
Module 3	
UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs. **UNIX Processes and Process Control:** **The Environment of a UNIX Process:** Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes. **Process Control:** Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3,	08

wait4 Functions, Race Conditions, exec Functions **RBT: L1, L2, L3**	
Module 4	
Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. **Overview of IPC Methods**, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores. **Shared Memory**, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions. **RBT: L1, L2, L3**	08
Module 5	
Signals and Daemon Processes: Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers. Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model. **RBT: L1, L2, L3**	08

Course Outcomes: The student will be able to :

- Explain Unix Architecture, File system and use of Basic Commands
- Illustrate Shell Programming and to write Shell Scripts
- Categorize, compare and make use of Unix System Calls
- Build an application/service over a Unix system.

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. Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill (Chapter 1,2 ,3,4,5,6,8,13,14)
2. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005 (Chapter 3,7,8,10,13,15)
3. Unix System Programming Using C++ - Terrence Chan, PHI, 1999. (Chapter 7,8,9,10)

Reference Books:

1. M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.
2. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2ndEdition, Wiley,2014.

Faculty can utilize open source tools to make teaching and learning more interactive.

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 AND ENGINEERING

NAME OF THE STAFF : Dr. REKHA B VENKATAPUR

SUBJECT CODE/NAME : 18CS56 UNIX PROGRAMMING

SEMESTER/YEAR : V 'B' Section

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: Introduction						
1	**Introduction**. Unix Components/Architecture. Features of Unix, The UNIX Environment and UNIX Structure,	L+D	Zoom App	1	1	1-9-2020
2	POSIX and single Unix specification. The login prompt. General features of Unix commands/ command structure.	L+ D	Zoom App	1	2	2-9-2020
3	Command arguments and options. Understanding of some basic commands such as echo, printf, ls, who, date, passwd, cal.	L+ D	Zoom App, Cygwin POSIX Compatible platform	1	3	3-9-2020
4	Combining commands. Meaning of Internal and external commands.	L+D	Zoom App, Cygwin POSIX Compatible platform	1	4	7-9-2020
5	The type command: knowing the type of a command and locating it. The root Login , Becoming Super user: su command	L+D	Zoom App, Cygwin POSIX Compatible platform	1	5	8-9-2020

6	**Unix Files**:Naming files, Basic file types/categories, Organization of files.	L+D	Zoom App, Cygwin POSIX Compatible platform	1	6	9-9-2020
7	Hidden files, Standard directories. Parent Child relationship. Home directory and HOME variable	L+D	Zoom App, Cygwin POSIX Compatible platform	1	7	10-9-2020
8	The PATH variable, Manipulating the PATH, Realtive and absolute path names. Directory commands – pwd,cd, mkdir,rmdir,the (.) dotand double dots(..) notations to represent	L+D	Zoom App, Cygwin POSIX Compatible platform	1	8	14-9-2020
9	File related commands –cat,mv.rm,cp and od commands	L+D	Zoom App, Cygwin POSIX Compatible platform	1	9	15-9-2020
10	Pedagogy activity – I	Quiz	Google forms online			
MODULE 2:File Attributes and Permissions						
11	Afile attributes and permissions : The ls Command with options, Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions, Directory permissions.	L+D	Zoom App, Cygwin POSIX Compatible platform	1	10	16-9-2020
12	**The Shells interpretive cycle**: Wild cards. Removing the special meaning of wild cards. Three standard files and redirection.	L+D	Zoom App, Cygwin POSIX Compatible platform	1	11	21-9-2020
13	**Connecting Commands** pipe.Basic and extended regular expressions The grep,egrep. Typical ex. Involving diff. regular expressions	L+ D	POSIX Compatible platform	1	12	22-9-2020
14	**Shell Programming** : Ordinary and environment variables	L+D	POSIX Compatible platform	1	13	23-9-2020

15	The .profile. Read and readonly commands. Command line arguments.	L+D	POSIX Compatible platform	1	14	24-9-2020
16	**Internal Assessment -I**					
17	exit and exit status of command. Logical operators for conditional execution. The test command & short cuts	L+D	POSIX Compatible platform	1	15	5-10-2020
18	The if, while, for and case control statements	L+D	POSIX Compatible platform	1	16	06-10-2020
19	The set and shift commands	L+D	POSIX Compatible platform	1	17	07-10-2020
20	Handling positional parameters. The HERE (<<) document and trap command, Simple shell program examples	L+D	POSIX Compatible platform	1	18	08-10-2020
21	Pedagogy Activity - II	Online assignment Zoom App		1		10-10-2020
MODULE 3: UNIX File APIs						
22	UNIX File APIs: General File APIs	L+D	POSIX Compatible platform	1	19	12-10-2020
23	File and Record locking, Directory File APIs, Device File APIs	L+D	POSIX Compatible platform	1	20	13-10-2020
24	FIFO File APIs, Symbolic Link APIs	L+D	POSIX Compatible platform	1	21	14-10-2020
25	**UNIX PROCESSES and Process Control :** **The Environment of a UNIX Process:** Introduction, main function, Process termination, Command-line arguments	L+D	POSIX Compatible platform	1	22	15-10-2020
26	Environment List, Memory layout of a C program, Shared Libraries, Memory Allocation	L+D	POSIX Compatible platform	1	23	19-10-2020
27	Environmental Variables, setjmp, longjmp Functions, getrlimit, setrlimit functions	L+D	POSIX Compatible platform	1	24	20-10-2020
28	Unix Kernel Support for Processes,	L+D	POSIX	1	25	21-10-2020

	Process Control: Introduction		Compatible platform			
29	Process identifier,fork,vfork	L+D	POSIX Compatible platform	1	26	22-10-2020
30	wait,waitpid,wait3,wait4 Functions	L+D	POSIX Compatible platform	1	27	24-10-2020
31	Race Conditions, exec Functions	L+D	POSIX Compatible platform	1	28	27-10-2020
Module 4: Changing User IDs and Group IDs						
32	Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting,	L+D	POSIX Compatible platform	1	29	28-10-2020
33	User Identification, Process Times, I/O Redirection.	L+D	POSIX Compatible platform	1	30	29-10-2020
34	Overview of IPC Methods,	L+D	POSIX Compatible platform	1	31	02-11-2020
35	Pipes, popen, pclose Functions,	L+D	POSIX Compatible platform	1	32	03-11-2020
36	Coprocesses, FIFOs,	L+D	POSIX Compatible platform	1	33	04-11-2020
37	System V IPC, Message Queues, Semaphores.	L+D	POSIX Compatible platform	1	34	05-11-2020
38	Shared Memory, Client-Server Properties,	L+D	POSIX Compatible platform	1	35	07-11-2020
39	**Internal Assessment - II**					
40	Stream Pipes, Passing File Descriptors	L+D	POSIX Compatible platform	1	36	12-11-2020
41	An Open Server-Version 1,	L+D	BB+LCD	1	37	17-11-2020
42	Client-Server Connection Functions.	L+D	BB+LCD	1	38	18-11-2020

Module5: Signals and Daemon Processes						
43	Signals: The UNIX Kernel Support for Signals	L+D	BB+LCD	1	39	19-11-2020
44	signal, Signal Mask, sigaction	L+D	BB+LCD	1	40	23-11-2020
45	The SIGCHLD Signal and the waitpid Function	L+D	BB+LCD	1	41	24-11-2020
46	The sigsetjmp and siglongjmp Functions	L+D	BB+LCD	1	42	25-11-2020
47	Kill, Alarm, Interval Timers	L+D	BB+LCD	1	43	26-11-2020
48	POSIX.lb Timers	L+D	BB+LCD	1	44	30-11-2020
49	Daemon Processes: Introduction	L+D	BB+LCD	1	45	01-12-2020
50	Daemon Characteristics	L+D	BB+LCD	1	46	02-12-2020
51	Coding Rules, Error Logging	L+D	BB+LCD	1	47	07-12-2020
52	Client-Server Model.	L+D	BB+LCD	1	48	08-12-2020
53	**Internal Assessment - III**					

Signature of Faculty

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

Signature of 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	**2018-2022**		
Year/Semester/section	**III/V/ B**		
Course Code-Title	**18CS56- UNIX PROGRAMMING**		
Name of the Instructor	**Dr Rekha B Venkatapur**	**Dept**	**CSE**

Assignment No: 1 **Total marks:10**
Date of Issue: 28/9/2020 **Date of Submission: 3/10/2020**

Sl.No.	Assignment Questions	K Level	CO	Marks
1.	Identify the differences between internal and external commands	3	1	1
2.	Construct the absolute and relative path names considering an example and explain in brief.		1	1
3.	Experiment with of the following commands: PATH b) HOME c) who d)ls e)printf f) PWD g)mkdir h) rmdir		1	2
4.	Describe command line arguments with suitable examples		1	1
5.	Online Quiz using google form (10 Qns in each test) on 22-9-2020 & 8-10-2020 Ex Qn Current file permission of a regular file "Attendance.txt" are **rw- -w-r-x** write the chmod expression required to change it to following: a) rwxrw-r-x b) –xrw-rwx c) rwxrwxrwx Using both relative and absolute methods of assigning permissions		1, 2	5

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

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

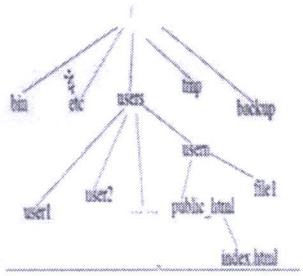
K S I T Bangalore

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ASSIGNMENT – 1 KEYS

Degree	**:**	**B.E**	**Semester**	**:**	**V A & B**
Branch	**:**	**Computer Science and Engineering**	**Course Code**	**:**	**18CS56**
Course Title	**:**	**Unix Programming**	**Max Marks**	**:**	**10**

Sl.No	Answers to Assignment Questions	Marks
1.	Differences between internal and external commands Internal command – Execution directly through shell as Shell built in , More priority, Faster execution, In case of occurrence of a command both in shell and user files, Shell command will be executed Example $type echo echo is a shell builtin External command – Less priority, slower, If a command exists both as an internal command of the shell as well as an external one (in /bin or /usr/bin), the shell will accord top priority to its own internal command with the same name. $type cat cat is /bin/ls	1
2.	Absolute and relative paths **ABSOLUTE PATHNAME** • Directories are arranged in a hierarchy with root (/) at the top. The position of any file within the hierarchy is described by its pathname. • Elements of a pathname are separated by a /. A pathname is absolute, if it is described in relation to root, thus absolute pathnames always begin with a /. • An **absolute path** is a complete **path** from start of actual **file** system from / directory. • Following are some examples of absolute filenames. /etc/passwd /users/kumar/progs/cprogs / dev/rdsk/Os3 **Example** • **date command can executed in two ways as** **$date // Relative path** Thu Sep 7 10:20:29 IST 2017 **$/bin/date // Absolute path** Thu Sep 7 10:20:29 IST 2017 **RELATIVE PATHNAME** • A pathname can also be relative to your current working directory. Relative pathnames never begin with /. Relative to user home directory, some pathnames might look like this – • **Relative path** is defined as the **path** related to the present working directly(pwd). progs/cprogs rdsk/Os3	2

	○ Path means a position in the directory tree. ○ Relative path ▪ starts from current working directory ▪ If you are already in the users directory, the Relative pathname for file1 is usern/file1 ○ Absolute path ▪ start from root (/) and follow the tree ▪ The absolute pathname for file1 is /users/usern/file1 	
3.	**a) PATH Variable (Environmental variable)** • Environmental variables are used to provide information to the programs in use. • A command runs in UNIX by executing a disk file. • When a command is specified say *date*, the system will locate the associated file from a list of directories specified in the PATH variable and then executes it. • The PATH variable normally includes the current directory also. • Specifies the locations in which Shell has to search for files • List of directories searched by the shell to locate a command **$echo $PATH** **/bin:/bin/usr:** **b) HOME Variable** • A shell variable HOME indicates home directory of current user • This variable is set for user by admin in /etc/passwd **Ex.$echo $HOME** **/home/kumar** **c)WHO** **who command** : who are the users? • Unix maintains an account of all users who are logged on to the system Who command displays an informative listing of there users • $ who **root** console **aug 1 07:51** (:0) **kumar**pts/10 **aug 1 07:56** (pc123.heavens.com) **Sharma**pts/6 **aug 1 02:10** (pc123.heavens.com) **d) ls** **ls command** : listing directory contents • Use to obtain a list of all filename in the current directory **Syntax:** ls [options] [argument] • Ex: ls **e)printf** **printf command** : An alternate to echo It exists as an external command but only in bash shell it is built in Usage $printf " Good Morning\n" Good Morning $_	1
4.	UNIX commands take the following general form: verb [options] [arguments] where verb is the command name that can take a set of optional options and one or more optional arguments. Commands, options and arguments have to be separated by spaces or tabs to enable the shell to interpret them as words. A contiguous string of spaces and tabs together is called a	1

	whitespace. The shell compresses multiple occurrences of whitespace into a single whitespace. Unix arguments range from simple to complex. They consists of options, expressions, instructions, file names etc. **Options** An option is preceded by a minus sign (-) to distinguish it from filenames. Example: $ ls –l There must not be any whitespaces between – and l. Options are also arguments, but given a special name because they are predetermined. Options can be normally compined with only one – sign. i.e., instead of using $ ls –l –a –t we can as well use, $ ls –lat Because UNIX was developed by people who had their own ideas as to what options should look like, there will be variations in the options. Some commands use + as an option prefix instead of -. f)PWD Any time user can know the current working directory using pwd command. **$ pwd** **/home/kumar** Like HOME it displays the absolute path. g)mkdir • Directories are created with **mkdir**(make directory) command. The command is followed by names of the directories to be created. A directory patch is created under current directory like this: **$mkdir progs progs/cprogsprogs/javaprogs** • This creates three subdirectories – progs, cprogs and javaprogs under progs. • The order of specifying arguments is important. You cannot create subdirectories before creation of parent directory. h)rmdir • The **rmdir**(remove directory) command removes the directories. You have to do this to remove progs: **$rmdir progs** • If **progs**is empty directory then it will be removed form system. • **rmdir**expect the arguments reverse of mkdir. Note: only empty directories can be removed by this command (without –r)	
5.	Quiz – 10 questions (reduced to 5 Marks)	5

Course In charge

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

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

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
ASSIGNMENT QUESTIONS

Academic Year	**2020-21**		
Batch	**2018-2022**		
Year/Semester/section	**III/V/A & B**		
Course Code-Title	**18CS56- UNIX PROGRAMMING**		
Name of the Instructor	**Dr Rekha B Venkatapur**	**Dept**	**CSE**

Assignment No: 2 **Total marks:10**
Date of Issue: 1/11/2020 **Date of Submission: 10/11/2020**

Sl. No	Assignment Questions	K Level	CO	Marks
1.	Explain shell features of while and for with syntax	K3	CO2	1
2.	What would be the effect of the following commands: (a) grep"^[A - Z]" file1 (b) egrep "UNIX|Unix|unix" file1 (c) grep "UNIX$" file1 (d) grep "UNIX. UNIX" file1 (e) grep "J.*" file1 > file2	K3	CO2	1
3.	Write, Execute and Explain to others 1. Write and execute a program to check the type of a file (i.e. regular/directory/FIFO/Symbolic link etc) 2. Write & Execute a program to create a FIFO file called FIFO5 with access permission of read-write-execute for everyone. 3. Write a C Program to illustrate the use of mkfifo,open, read and close APIs for a FIFO file. Pedagogy Activity – Collaborative study through Team wise Execution of file APIS – Total 3 teams time 30 Mins Date of execution : 20-11-2020	K3	CO3	4
4	Explain the following system calls in detail: i) _exit ii) exit iii) atexit functions iv)fork v) vfork vi) wait()	K3	CO3	1
5	Explain attributes inherited by child process and attributes that are different between the parent and child processes	K3	CO3	1
6	Explain functions used for changing user ID and group ID	K3	CO4	1
7	Explain System function with an example program	K3	CO4	1

Course In charge

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

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 ODD SEMESTER

SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**V & VI**
Branch	:	**Computer Science & Engineering**	**Course Code**	:	**18CS56**
Course Title	:	**UNIX Programming**	**Max Marks**	:	**10**

Q. NO	POINTS	Marks
1	**while: looping** The general syntax of this command is as follows: while condition is true do commands done Example Program **for: looping with a list** The general syntax of for loop is as follows for variable in list do commands done Example Program	**1M**
2	**Identify the output of the following grep commands:** (a) grep"^[A - Z]" file1 Serach for the lines that does not begin with a capital character in file1 (b) egrep "UNIX|Unix|unix" file1 Search for line that contain UNIX or Unix or unix in file1 (c) grep "UNIX$" file1 Search for lines that end with UNIX in file1 (d) grep "UNIX. UNIX" file1 Search for lines with strings as UNIXany character space UNIX in file1 (e) grep "J.*" file1 > file2 Search for lines with strings that start with J followed by zero or more characters in file1 and redirect the output to file2	**1M**
3.	**Pedagogy Activity** – Collaborative study through Team wise Execution of file APIS – Total 3 teams time 30 Mins Date of execution : 20-11-2020 1. Write and execute a program to check the type of a file (i.e. regular/directory/FIFO/Symbolic link etc) 2. Write & Execute a program to create a FIFO file called FIFO5 with access permission of read-write-execute for everyone. 3. Write a C Program to illustrate the use of mkfifo,open, read and close APIs for a FIFO file.	**4M**

4.	**Exit Functions** Three functions terminate a program normally: _exit and _Exit, which return to the kernel immediately, and exit, which performs certain cleanup processing and then returns to the kernel. #include <stdlib.h> void exit(int status); #include <unistd.h> void _exit(intstatus); All three exit functions expect a single integer argument, called the exit status. **atexitFunction** With ISO C, a process can register up to 32 functions that are automatically called by exit. These are called exit handlers and are registered by calling the atexitfunction. #include <stdlib.h> intatexit(void (*func)(void)); Returns: 0 if OK, nonzero on error **fork Function** An existing process can create a new one by calling the fork function. #include <unistd.h> pid_t fork(void); Returns: 0 in child, process ID of child in parent, 1 on error. **vfork Function** The function vforkhas the same calling sequence and same return values as fork. ✓ The vfork function is intended to create a new process when the purpose of the new process is to exec a newprogram. The vfork function creates the new process, without copying the address space of the parent into the child, as the child won't reference that address space; the child simply calls exec (or exit) right after thevfork. **Wait function** • Block, if all of its children are stillrunning ✓ Return immediately with the termination status of a child, if a child has terminated and is waiting for its termination status to befetched. #include <sys/wait.h> pid_t wait(int *statloc); return: process ID if OK, 0 (see later), or 1 on error.	**1M**
5.	**There are numerous other properties of the parent that are inherited by the child:** o Real user ID, real group ID, effective user ID, effective group ID o Supplementary group IDs o Process group ID o Session ID oControlling terminal	**1M**

6.	o The set-user-ID and set-group-ID flags o Current working directory o Root directory o File mode creation mask o Signal mask and dispositions o The close-on-exec flag for any open file descriptors o Environment o Attached shared memory segments o Memory mappings o Resource limits **The differences between the parent and child are** • The return value from fork • The process IDs are different • The two processes have different parent process IDs: the parent process ID of the child is the parent; the parent process ID of the parent doesn't change • The child's tms_utime, tms_stime, tms_cutime, and tms_cstime values are set to 0 • File locks set by the parent are not inherited by the child • Pending alarms are cleared for the child • The set of pending signals for the child is set to the empty set	**1M**
7.	**CHANGING USER IDs AND GROUP IDs** When our programs need additional privileges or need to gain access to resources that they currently aren't allowed to access, they need to change their user or group ID to an ID that has the appropriate privilege or access. Similarly, when our programs need to lower their privileges or prevent access to certain resources, they do so by changing either their user ID or group ID to an ID without the privilege or ability access to the resource #include <unistd.h> intsetuid(uid_tuid); intsetgid(gid_tgid); Both return: 0 if OK, 1 on error setreuidand setregidFunctions #include <unistd.h> intsetreuid(uid_truid, uid_teuid); intsetregid(gid_trgid, gid_tegid); Both return : 0 if OK, -1 on error seteuid and setegid functions #include <unistd.h> intseteuid(uid_tuid); intsetegid(gid_tgid); Both return : 0 if OK, 1 on error **system Function** #include <stdlib.h> int system(const char *cmdstring); If cmdstring is a null pointer, system returns nonzero only if a command processor is available. This feature determines whether the system function is supported on a given operating system. Under the UNIX System, system is always available. Because system is implemented by calling fork, exec, and waitpid, there are three	**1M**

types of return values.

- If either the fork fails or waitpid returns an error other than EINTR, system returns 1 with errno set to indicate the error.
- If the exec fails, implying that the shell can't be executed, the return value is as if the shell had executed exit(127).
- Otherwise, all three functions fork, exec, and waitpid succeed, and the return value from system is the termination status of the shell, in the format specified for waitpid.

Program: The system function, without signal handling

```
#include <sys/wait.h>
 #include <errno.h>
#include <unistd.h>
int system(const char *cmdstring) /* version without signal handling */
{ pid_tpid; int status;
if (cmdstring == NULL) return(1); /* always a command processor with UNIX */
if ((pid = fork()) < 0) { status = -1; /* probably out of processes */
 }
 else if (pid == 0)
 { /* child */
execl("/bin/sh", "sh", "-c", cmdstring, (char *)0);
_exit(127); /* execl error */ }
else { /* parent */
while (waitpid(pid, &status, 0) < 0)
 { if (errno != EINTR)
 { status = -1; /* error other than EINTR from waitpid() */ break; }
}
}
return(status);
}
```

Course incharge | **Module Coordinator** | **HOD-CSE**

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

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

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	**2018-2022**		
Year/Semester/section	**III/V/A & B**		
Course Code-Title	**18CS56/UNIX PROGRAMMING**		
Name of the Instructor	**Dr Rekha B Venkatapur**	**Dept**	**CSE**

Assignment No: 3 **Total marks:10**
Date of Issue: 1/1/2021 **Date of Submission: 6/1/2021**

Sl. No	Assignment Questions	K Level	CO	Marks
1.	**Make use of** relevant data structure and explain the semaphore semget and semctl API's used for IPC.	K3	**CO4**	1
2.	**Construct** a code snippet that the parent sends "Hello world "message to child process through the pipe. Child on receiving this message should display it on output screen	K3	**CO4**	1
3.	**Utilize** FIFO and explain client server structure with a neat diagram	K3	**CO4**	1
4.	**Develop** IPC using: a. Streams pipe b. Passing file descriptors c. Co-Processes d. popen and pclose	K3	**CO4**	1
5.	**Identify** the ways in which the process can handle signals and also explain UNIX kernel support provided for handling signals.	K3	**CO5**	1
6.	**Build** the signal APIs with their prototypes and uses for sigprocmask, sigpending and sigaction.	K3	**CO5**	1
7.	**Identify** the timer manipulation API's in POSIX.1b	K3	**CO5**	1
8.	**Make use of** an example program and explain the kill and alarm functions.	K3	**CO5**	1
9.	**Identify** the Daemon characteristics and coding rules with an example.	K3	**CO5**	1
10.	**Interview** the deamon processes and explain with a neat diagram the error logging facility for a deamon process.	K3	CO5	1

Course In charge

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

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 ODD SEMESTER

SCHEME AND SOLUTION

Degree	**:**	**B.E**	**Semester**	**:**	**V A**
Branch	**:**	**Computer Science & Engineering**	**Course Code**	**:**	**18CS56**
Course Title	**:**	**UNIX Programming**	**Max Marks**	**:**	**10**

Q. NO	POINTS	Marks
1	The kernel maintains a semid_dsstructure for each semaphore set: **structsemid_ds {** **structpc_permsem_perm;** **unsignedshort sem_nsems; /* # of semaphores in set */** **time_t sem_otime; /* last-semop() time*/** **time_t sem_ctime; /* last-change time*/** **};** Each semaphore is represented by an anonymous structure containing at least the following members: **struct {** **unsignedshort semval; /* emaphore value, always >= 0 */** **pid_t sempid; /* pid for last operation*/** **unsignedshort semncnt; /* # processes awaiting semval>curval */** **unsignedshort semzcnt; /* # processes awaiting semval==0*/** **};** The first function to call is semget to obtain a semaphore ID. **#include <sys/sem.h>** **intsemget(key_t key, intnsems, int flag);** Returns: semaphore ID if OK, 1 on error When a new set is created, the following members of the semid_ds structure are initialized. • The ipc_perm structure is initialized. The mode member of this structure is set to the corresponding permission bits offlag. • sem_otime is set to0. • sem_ctime is set to the currenttime. • sem_nsems is set tonsems. The number of semaphores in the set is nsems. If a new set is being created (typically in the server), we must specify nsems. If we are referencing an existing set (a client), we can specify nsems as 0. The semctl function is the catchall for various semaphore operations. **#include <sys/sem.h>** **intsemctl(intsemid, intsemnum,int cmd,... /* union semunarg*/);** The fourth argument is optional, depending on the command requested, and if present, is of type semun, a union ofvarious command-specific arguments:	**1M**

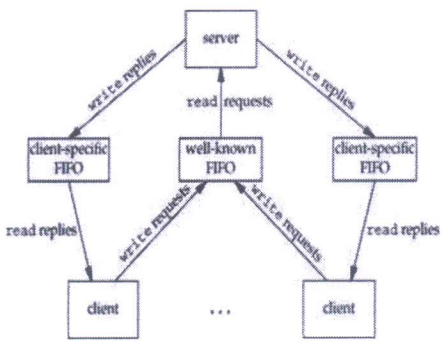
```
union semun
{
  int     val;                  /* for SETVAL*/
  structsemid_ds*buf;           /* for IPC_STAT and IPC_SET */ unsignedsh
};
```

The *cmd*argument specifies one of the above ten commands to be performed on the set specified by semid. The function semopatomically performs an array of operations on a semaphore set.

2. Construct a code snippet that the parent sends "Hello world "message to child process through the pipe. Child on receiving this message should display it on output screen **1M**

```
#include <stdio.h>
#include<fnctl.h>
#include<unistd.h>
    int main(void)
    {
    int n;
    intfd[2];
    pid_tpid;
    char line[MAXLINE];
    if (pipe(fd) < 0)
    err_sys("pipeerror");
  if ((pid = fork()) < 0 ) {
    err_sys("forkerror");
    }
     else if (pid>0){                    /* parent */
    close(fd[0]);
        write(fd[1], "hello world\n", 12);
    }else{                              /* child */
    close(fd[1]);
        n = read(fd[0], line, MAXLINE);
        write(STDOUT_FILENO, line, n);
    }
    exit(0);
}
```

3. **1M**



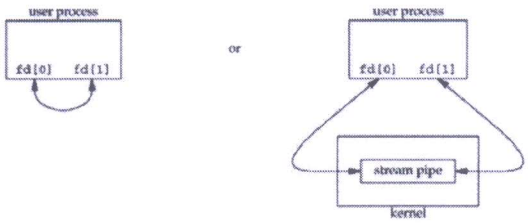
- FIFO's can be used to send data between a client and a server. If we have a server that is contacted by numerous clients, each client can write its request to a well-known FIFO that the server creates. Since there are multiple writers for the FIFO, the requests sent by the clients to the server need to be less than PIPE_BUF bytes insize.
- This prevents any interleaving of the client writes. The problem in using FIFOs for this type of client server communication is how to send replies back from the server to eachclient.
- A single FIFO can't be used, as the clients would never know when to read their response versus responses for other clients. One solution is for each client to send its process ID with the request. The server then creates a unique FIFO

for each client, using a pathname based on the client's processID.

- For example, the server can create a FIFO with the name /vtu/ ser.XXXXX, where XXXXX is replaced with the client's process ID. This arrangement works, although it is impossible for the server to tell whether a client crashes. This causes the client-specific FIFOs to be left in the filesystem.

4. **a. Stream Pipes:** Pipes now provide a **bidirectional** mechanism for process communication. When a pipe is created by the **pipe** system call, two Streams are opened and connected together, thus providing a full-duplex mechanism. **1M**

- A STREAMS-based pipe is a **bidirectional** (full-duplex) pipe. To obtain bidirectional data flow between a parent and a child, only a single STREAMS pipe is required.
- Following figure shows the two ways to view a STREAMS pipe.



b. Passing File Descriptors

- The ability to pass an open file descriptor between processes is **powerful**.
- It can lead to different ways of designing client server applications. It allows **one process** (typically a server) to do everything that is required to open a file (involving such details as translating a network name to a network address, dialing a modem, negotiating locks for the file, etc.) and simply **pass back** to the calling process a descriptor that can be used with all the I/O functions.
- All the details involved in opening the file or device are **hidden** from the client.
- When we pass an open file descriptor from one process to another, the passing process and the receiving process should **share the same file table entry.** Following figure shows the desired arrangement.

Technically, we are passing a pointer to an open file table entry from one process to another. This pointer is assigned the first available descriptor in the receiving process

intsend_fd(intfd, intfd_to_send);
intsend_err(intfd, int status, const char *errmsg);
intrecv_fd(intfd, ssize_t (*userfunc)(int, const
void *, size_t));

c. COPROCESSES

A UNIX system filter is a program that reads from standard input and writes to standard output. Filters are normally connected linearly in shell pipelines. A filter becomes a coprocess when the same program generates the filter's input and reads the filter's output. A coprocess normally runs in the background from a shell, and its standard input and standard output are connected to another program using apipe.

The process creates two pipes: one is the standard input of the coprocess, and the other is the standard output of the coprocess. Figure 15.16 shows thisarrangement.

d. popenAND pcloseFUNCTIONS

Since a common operation is to create a pipe to another process, to either read its output or send it input, the standard I/O library has historically provided the popen and pclose functions. These two functions handle all the dirty work that we've been doing ourselves: creating a pipe, forking a child, closing the unused ends of the pipe, executing a shell to run the command, and waiting for the command to terminate.

```
#include <stdio.h>
FILE *popen(const char *cmdstring, const char *type);
```

Returns: file pointer if OK, NULLon error

```
intpclose(FILE *fp);
```

Returns: termination status of cmdstring, or 1 on error

The function popendoes a forkand execto execute the cmdstring, and returns a standard I/O file pointer. If type is "r", the file pointer is connected to the standard output of cmdstring

5 The ways in which the process can handle signals **1M**

- Accept the **default action** of the signal, which for most signals will terminate the process.
- **Ignore** the signal. The signal will be discarded and it has no effect whatsoever on the recipient process.
- Invoke a **user-defined** function. The function is known as a signal handler routine and the signal is said to be ***caught*** when this function is called.

The unix kernel support of signals

- When a signal is generated for a process, the kernel will set the corresponding signal flag in the process table slot of the recipientprocess.
- If the recipient process is asleep, the kernel will awaken the process by schedulingit.
- When the recipient process runs, the kernel will check the process U-area that contains an array of signal handlingspecifications.
- If array entry contains a zero value, the process will accept the default action of thesignal.
- If array entry contains a 1 value, the process will ignore the signal and kernel will discardit.
- If array entry contains any other value, it is used as the function pointer for a user-defined signal handler routine.

6. **a. sigprocmask** **1M**

A process initially inherits the parent's signal mask when it is created, but any pending signals for the parent process are not passed on. A process may query or set its signal mask via the sigprocmask API:

```
#include <signal.h>
intsigprocmask(intcmd, constsigset_t *new_mask, sigset_t
*old_mask);
```

Returns: 0 if OK, 1 on error

b. Sigpending

A process can query which signals are pending for it via the sigpending API:

```
#include<signal.h>

intsigpending(sigset_t* sigmask);
```

Returns 0 if OK, -1 if fails.

The sigpending API can be useful to find out whether one or more signals are pending for a process and to set up special signal handling methods for these signals before the process calls the sigprocmask API to unblock them.

c. Sigaction

The sigaction API blocks the signal it is catching allowing a process to specify

```
#include<signal.h>

intsigaction(intsignal_num, structsigaction* action,
```

additional signals to be blocked when the API is handling a signal.Thesigaction API prototype is:

Returns: 0 if OK, 1 on error

7. **Identify** the timer manipulation API's in POSIX.1b **1M**

#include<signal.h> #include<time.h>

inttimer_create(clockid_t clock, structsigevent* spec, timer_t* timer_hdrp);

inttimer_settime(timer_ttimer_hdr, int flag, structitimerspec* val, structitimerspec* old);

inttimer_gettime(timer_ttimer_hdr, structitimerspec* old);

inttimer_getoverrun(timer_ttimer_hdr);

inttimer_delete(timer_ttimer_hdr);

8. **KILL** **1M**

A process can send a signal to a related process via the kill API. This is a simple means of inter-process communication or control. The function prototype of the APIis:

```
#include<signal.h>

int kill(pid_tpid, intsignal_num);
```

Returns: 0 on success, -1 on failure.

The signal_num argument is the integer value of a signal to be sent to one or more processes designated by pid. The possible values of pid and its use by the kill API are:

ALARM

The alarm API can be called by a process to request the kernel to send the SIGALRM signal after a certain number of real clock seconds. The function prototype of the API is:

```
#include<signal.h>

Unsigned intalarm(unsigned inttime_interval);
```

Returns: 0 or number of seconds until previously set alarm

9.	**Characteristics of daemons are:** ▪ Daemons run inbackground. ▪ Daemons have super-userprivilege. ▪ Daemons don't have controllingterminal. ▪ Daemons are session and groupleaders. **Coding rules** Some basic rules to coding a daemon prevent unwanted interactions from happening. • Call umaskto set the file mode creation mask to 0. ▪ Call fork and have the parent exit. ▪ Call setsidto create a new session. ▪ Change the current working directory to the root directory. ▪ Unneeded file descriptors should be closed. ▪ Some daemons open file descriptors 0, 1, and 2 to /dev/null so that any library routines that try to read from standard input or write to standard output or standard error will have no effect.	**1M**
10.	Daemons are processes that live for a long time. They are often started when the system is bootstrapped and terminate only when the system is shut down. **Error logging** ▪ One problem a daemon has is how to handle error messages. It can't simply write to standard error, since it shouldn't have a controlling terminal. We don't want all the daemons writing to the console device, since on many workstations, the console device runs a windowing system. A central daemon error-logging facility is required. There are three ways to generate log messages: ▪ Kernelroutinescancallthelogfunction.Thesemessagescanbereadbyanyuserprocessthatopen sand reads the /dev/klogdevice. ▪ Most user processes (daemons) call the syslog(3) function to generate log messages. This causes the message to be sent to the UNIX domain datagram socket/dev/log. ▪ A user process on this host, or on some other host that is connected to this host by a TCP/IP network, can send log messages to UDP port 514. Note that the syslog function never generates these UDP datagrams: they require explicit network programming by the process generating the logmessage.	**1M**

Course incharge

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

Module Coordinator

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

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

I SESSIONAL TEST QUESTION PAPER 2020 – 21ODDSEMESTER

Set A

USN										

Degree : B.E
Branch : Computer Science &Engineering
Course Title :UNIX PROGRAMMING
Duration :90Minutes

Semester: V A& B
CourseCode: 18CS56
Date: 07-10-2020
MaxMarks: 30

Note: Answer ONE full question from each part.

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	Draw the tree structure of the file system created by the following commands (Assume you are in the directory /home/Organization). $mkdirBoard_of_Directors $cd Board_of_Directors $mkdir CEO $cd CEO $mkdirDirector_OperationDirector_Technical $cd Director_Operation $mkdirMgr_AdminMgr_Finance $cd ../Director_Technical $mkdirGM_Technical • Why is it not possible to issue the command $rmdir /usr/Organization/Director_Technical • Write a sequence of commands to create a file Hello.txt in Mgr_Admin and copy it to GM_Technical.	6	CO1	K3
(b)	Identify the differences between internal and external commands.	6	CO1	K3
(c)	Experiment with of the following commands: a) echo b) cp c) od d) rmdir e)mv f)pwd	6	CO1	K3
	OR			
2(a)	Construct a neat diagram of the UNIX file system and explain different types of files supported in UNIX.	6	CO1	K3
(b)	Identify the salient features of UNIX operating system	6	CO1	K3
(c)	Experiment with of the following commands: a) date b) who c)mkdir d) cat e)cal f)wc	6	CO1	K3
	PART-B			
3(a)	Interview the significance of the seven fields of 'ls –l'command.	6	CO2	K3

(b)	Current file permission of a regular file "Attendance .txt" are **rw- -w-r-x** write the chmod expression required to change it to following: a) rwxrw-r-x b) - -xrw-rwx c) rwxrwxrwx Using both relative and absolute methods of assigning permissions	**6**	**CO2**	**K3**
	OR			
4(a)	Make use of example and explain absolute and relative methods of assigning permissions to file.	**6**	**CO2**	**K3**
(b)	Current file permission of a regular file "Marks.txt" are **-w- rw-rwx** write the chmod expression required to change it to following: a) r-x-w-r-x b) - -xrw-r- - c) - - - - - - - - Using both relative and absolute methods of assigning permissions	**6**	**CO2**	**K3**

Signature of Course incharge **Signature of Module Coordinator** **Signature of HOD**

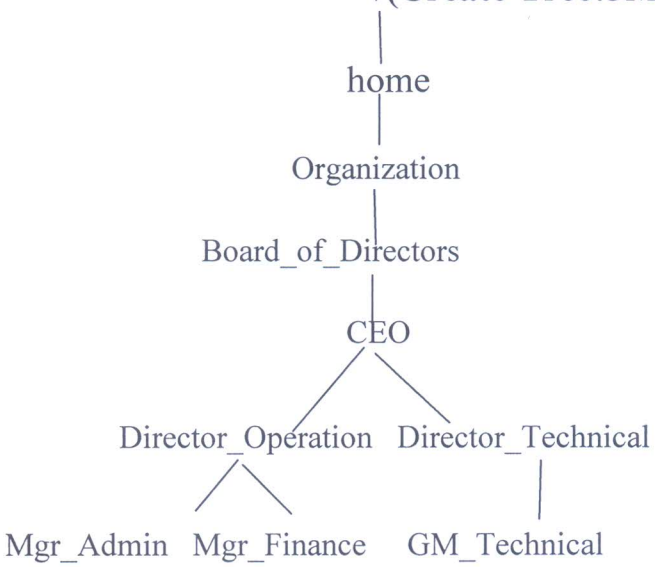
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2020-21ODD SEMESTER

SET A

SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**CSE**	**Course Code**	:	**18CS56**
Course Title	:	**UNIX Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
1(a)	**/(Create Tree:3M)**  • It is not possible to delete $rmdir /usr/Organization/Director_Technical as the directory is not empty**(1M)** • $cd /home/Organization/Board_of_Directors/CEO/Director_Operation/Mgr_Admin • $cat Hello.txt • $cp Hello.txt ../../Director_Technical/GM_Technical **(2M)**	3+1+2=6M
(b)	• **Internal Commands(3M)** – The shell execute command(file) from its own set of built-in commands. – Not stored as separate files in /bin directory. – **$ type echo** #Output as: echo is shell built-in – Execution speed is faster – Internal commands will have top priority compare • **External Commands(3M)** – The command (file) has an independent existence in the /bin	3+3=6M

	directory. – **$ type ls** # ls is an external command ls is /bin/ls – External commands less priority compare to internal command – Execution speed is slow compared to internal commands.	
(c)	Explain each command with example**: 1 Mark each** a) echo – Display content on screen b) cp – Copy Command c) od – Display text in octal numbers d) rmdir – Remove Directory e)mv – Move file f)pwd- Present working directory.	1*6=6
2(a)	UNIX File system diagram **(2M)** Explanation of standard directories in file system**(4M)** /bin, /home, /dev, /usr, /var, /sbin, /etc	2+4=6
(b)	Any 6 Features of UNIX: **1Mark each** Multiuser, Multitasking, Building Block, UNIX Toolkit, Pattern Matching, Programming Facility, Documentation, Portability, Organized file system	1*6= 6M
(c)	Explain each command with example**: 1 Mark each** a) date : Display date and time b) who – Display the users logged into system c)mkdir- make directory d) cat- create or display contents of file e)cal- Display calendar f)wc – Display number of lines, words and characters in a file	1*6=6M
3(a)	Give example and explain the seven attributes of all files in the current directory and they are: File type and Permissions, Links, Ownership, Group ownership, File size, Last Modification date and timeFile, name	1*6=6M
(b)	Current file permission of a regular file "Attendance .txt" are rw- -w-r-x write the chmod expression required to change it to following: **2Marks each** Considering **rw- -w-r-x** ascurrent permission. a) **rwxrw-r-x** Relative: $chmod u+x,g+r Attendance.txt Absolute:$chmod 765 Attendance.txt **b) - -xrw-rwx** Relative :$chmod u-rw,u+x,g+r,o+w Attendance.txt Absolute :$chmod 167 Attendance.txt **c) rwxrwxrwx** Relative :$chmod u+x,g+rx,o+w Attendance .txt	2+2+2= 6M

	Absolute :$chmod 777 Attendance .txt	
4(a)	**Relative Permissions(3M)**chmod only changes the permissions specified in the command line and leaves the other permissions unchanged. **Its syntax is: chmod category operation permission filename(s)** chmod takes an expression as its argument which contains: user category (user, group, others) operation to be performed (assign or remove a permission) type of permission (read, write, execute) **Category operation permission** u - user + assign r - read g - group - remove w - write o - others = absolute x - execute a - all (ugo) **Absolute Permissions(3M)**Here, we need not to know the current file permissions. We can set all nine permissions explicitly. A string of three octal digits is used as an expression. The permission can be represented by one octal digit for each category. For each category, we add octal digits. If we represent the permissions of each category by one octal digit, this is how the permission can be represented: Read permission – 4 (octal 100) Write permission – 2 (octal 010) Execute permission – 1 (octal 001)	3+3=6M
(b)	Current file permission of a regular file "Marks.txt" are -w- rw-rwx write the chmod expression required to change it to following: **2Marks each** **a) r-x-w-r-x** Relative: $chmod u+rx,u-w,g-r,o-w Marks.txt Absolute: $chmod 525 Marks.txt **b) - -xrw-r- -** Relative:$chmod u-w,u+x, o-wx Marks.txt Absolute:$chmod 164 Marks.txt **c) - - - - - - - -** Relative: $chmod u-w,g-rw,o-rwx Marks.txt Absolute: $chmod 000 Marks.txt	2+2+2=6M

Signature of Course incharge **Signature of Module Coordinator** **Signature of HOD**

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

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

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 – 21ODDSEMESTER

Set B

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

Degree : B.E — **Semester:** V A& B
Branch : Computer Science &Engineering — **CourseCode:** 18CS56
Course Title :UNIX PROGRAMMING — **Date:** 07-10-2020
Duration :90Minutes — **MaxMarks:**30

Note: Answer ONE full question from each part.

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	Draw the tree structure of the file system created by the following commands(Assume you are in the directory /usr/office). Why is it not possible to issue the command rmdir /usr/office/right $mkdirleft $mkdir middle $mkdir right $ cd left $mkdir left middle right $cd ../middle $mkdir dir1 dir2 /usr/office/right/dir3	6	CO1	(Applying) K3
(b)	Identify the salient features of UNIX operating system	6	CO1	(Applying) K3
(c)	Experiment with of the following commands: a) date b) who c)mkdir d) cat e)cal f)wc	6	CO1	(Applying) K3
	OR			
2(a)	Construct a neat diagram of the UNIX file system and explain different types of files supported in UNIX.	6	CO1	(Applying) K3
(b)	Identify the differences between internal and external commands.	6	CO1	(Applying) K3
(c)	Experiment with of the following commands: a) echo b) cp c) od d) rmdir e)mv f)pwd	6	CO1	(Applying) K3
	PART-B			

3(a)	Interview the significance of the seven fields of 'ls –l'command.	6	CO2	(Applying) K3
(b)	Current file permission of a regular file "Marks.txt" are **-w-rw-rwx** write the chmod expression required to change it to following: a) r-x-w-r-x b) - -xrw-r- - c) - - - - - - - - Using both relative and absolute methods of assigning permissions	6	CO2	(Applying) K3
	OR			
4(a)	Investigate the methods of assigning permissions to file using absolute and relative access.	6	CO2	(Applying) K3
(b)	Current file permission of a regular file "Attendance .txt" are **rw- -w-r-x** write the chmod expression required to change it to following: b) rwxrw-r-x b) –xrw-rwx c) rwxrwxrwx Using both relative and absolute methods of assigning permissions	6	CO2	(Applying) K3

Signature of Course incharge **Signature of Module Coordinator** **Signature of HOD**

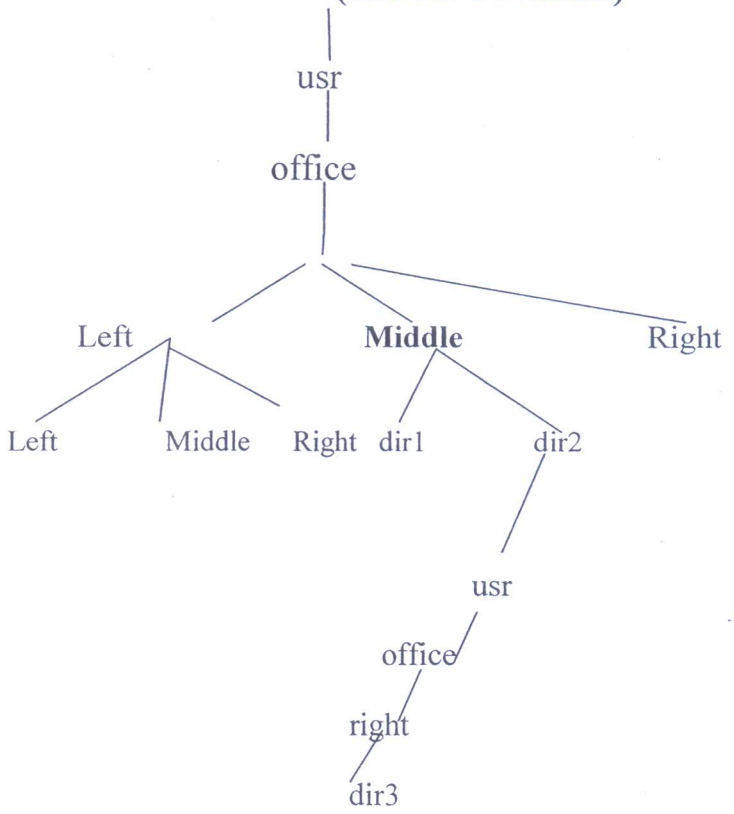
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2020-21

ODD SEMESTER

SCHEME AND SOLUTION –SET B

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**CSE**	**Course Code**	:	**18CS56**
Course Title	:	**UNIX Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
	PART-A	
1(a)	**/(Create Tree:3M)**  • It is not possible to delete rmdir /usr/office/right as the directory is not empty**(1M)**	3+1+2=6M
(b)	Any 6 Features of UNIX: **1Mark each** Multiuser, Multitasking, Building Block, UNIX Toolkit, Pattern Matching, Programming Facility, Documentation, Portability, Organized file system	1*6= 6M
(c)	Explain each command with example**: 1 Mark each** a) date : Display date and time b) who – Display the users logged into system c)mkdir- make directory d) cat- create or display contents of file e)cal- Display calendar	1*6=6M

	f)wc – Display number of lines, words and characters in a file	
2(a)	UNIX File system diagram **(2M)** Explanation of standard directories in file system**(4M)** /bin, /home, /dev, /usr, /var, /sbin, /etc	2+4=6
(b)	• **Internal Commands(3M)** – The shell execute command(file) from its own set of built-in commands. – Not stored as separate files in /bin directory. – **$ type echo** #Output as: echo is shell built-in – Execution speed is faster – Internal commands will have top priority compare • **External Commands(3M)** – The command (file) has an independent existence in the /bin directory. – **$ type ls** # ls is an external command ls is /bin/ls – External commands less priority compare to internal command – Execution speed is slow compared to internal commands.	3+3=6M
(c)	Explain each command with example**: 1 Mark each** a) echo – Display content on screen b) cp – Copy Command c) od – Display text in octal numbers d) rmdir – Remove Directory e)mv – Move file f)pwd- Present working directory.	1*6=6
PART- B		
3(a)	Give example and explain the seven attributes of all files in the current directory and they are: File type and Permissions, Links, Ownership, Group ownership, File size, Last Modification date and timeFile, name	1*6=6M
(b)	Current file permission of a regular file "Attendance .txt" are rw- -w-r-x write the chmod expression required to change it to following: **2Marks each** Considering **rw- -w-r-x** ascurrent permission. a) **rwxrw-r-x** Relative: $chmod u+x,g+r Attendance.txt Absolute:$chmod 765 Attendance.txt **b) - -xrw-rwx** Relative :$chmod u-rw,u+x,g+r,o+w Attendance.txt Absolute :$chmod 167 Attendance.txt **c) rwxrwxrwx** Relative :$chmod u+x,g+rx,o+w Attendance .txt Absolute :$chmod 777 Attendance .txt	2+2+2= 6M

4(a)	**Relative Permissions(3M)**chmod only changes the permissions specified in the command line and leaves the other permissions unchanged. **Its syntax is: chmod category operation permission filename(s)** chmod takes an expression as its argument which contains: user category (user, group, others) operation to be performed (assign or remove a permission) type of permission (read, write, execute) **Category operation permission** u - user + assign r - read g - group - remove w - write o - others = absolute x - execute a - all (ugo) **Absolute Permissions(3M)**Here, we need not to know the current file permissions. We can set all nine permissions explicitly. A string of three octal digits is used as an expression. The permission can be represented by one octal digit for each category. For each category, we add octal digits. If we represent the permissions of each category by one octal digit, this is how the permission can be represented: Read permission – 4 (octal 100) Write permission – 2 (octal 010) Execute permission – 1 (octal 001)	3+3=6M
(b)	Current file permission of a regular file "Marks.txt" are -w- rw-rwx write the chmod expression required to change it to following: **2Marks each** **a) r-x-w-r-x** Relative: $chmod u+rx,u-w,g-r,o-w Marks.txt Absolute: $chmod 525 Marks.txt **b) - -xrw-r- -** Relative:$chmod u-w,u+x, o-wx Marks.txt Absolute:$chmod 164 Marks.txt **c) - - - - - - - -** Relative: $chmod u-w,g-rw,o-rwx Marks.txt Absolute: $chmod 000 Marks.txt	2+2+2=6M

Signature of Course incharge

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

Signature of Module Coordinator

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

Signature of HOD

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

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

SET – A

USN										

Degree : **B.E** **Semester** : **V A& B**
Branch : **Computer Science and Engineering** **Course Code** : **18CS56**
Course Title : **UNIX PROGRAMMING** **Date** : **19-11-2020**
Duration : **90 Minutes** **Max Marks** : **30**

Note: Answer ONE full question from each part.

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Construct** the prototype and explain locking of files and records in detail.	6	CO3	K3
(b)	**Build** the prototype and explain in detail the following system calls: i) _exit ii) exit iii) atexit	6	CO3	K3
(C)	**Identify** the attributes inherited by child process and attributes that are different between the parent and child processes	6	CO3	K3
	OR			
2(a)	**Identify** hard link and soft link API and explain in detail.	6	CO3	K3
(b)	**Make use of** memory layout of c program and explain all the sections in detail.	6	CO3	K3
(C)	**Build** the prototype and explain in detail the following system calls: i)fork ii) vfork iii) wait	6	CO3	K3
	PART-B			
3(a)	**Utilize** while and for looping construct of shell script and explain its syntax and write example programs.	6	CO2	K3
(b)	**Make use of** System function and explain its prototype with an example program	6	CO4	K3
	OR			
4(a)	**Identify** the purpose of grep command and explain its all options	6	CO2	K3
(b)	**Construct** the prototype and explain all the functions used for changing user ID and group ID.	6	CO4	K3

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

II SESSIONAL TEST QUESTION PAPER 2020-21ODD SEMESTER

SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**CSE**	**Course Code**	:	**18CS56**
Course Titl	:	**UNIX Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
1(a)	Multiple processes performs read and write operation on the same fileconcurrently. This provides a means for data sharing among processes, but it also renders difficulty for any process in determining when the other process can override data in afile. So, in order to overcome this drawback UNIX and POSIX standard support file lockingmechanism. ▪ File locking is applicable for regularfiles. ▪ Only a process can impose a write lock or read lock on either a portion of a file or on the entirefile. Prototype(2M) #include<fcntl.h> int***fcntl***(**int**fdesc,**int**cmd_flag,.....); The first argument specifies the filedescriptor. The second argument cmd_flag specifies what operation has to beperformed. If fcntl is used for file locking then it can valuesas F_SETLK sets a file lock, do not block if this cannot succeed immediately. F_SETLKW sets a file lock and blocks the process until the lock is acquired. F_GETLKqueries as to which process locked a specified region offile. For file locking purpose, the third argument to fctnl is an address of a *struct flock* typevariable. This variable specifies a region of a file where lock is to be set, unset orqueried. **struct flock (2M)** **{** **short l_type; /* what lock to be set or to unlock file */** **short l_whence; /* Reference address for the next field */ off_t l_start;** **/*offset from the l_whence reference addr*/** **off_t l_len; /*how many bytes in the locked region*/** **pid_t l_pid; /*pid of a process which has locked the file*/** **};** ▪ The l_type field specifies the lock type to be set orunset. ▪ The possible values, which are defined in the <fcntl.h> header, and their uses are	2+2+1+1=6 M

	l_type value(1M) **Use** F_RDLCK — Set a read lock on a specified region F_WRLCK — Set a write lock on a specified region F_UNLCK — Set a write lock on a specified region The l_whence, l_start&l_len define a region of a file to be locked orunlocked. ▪ The possible values of l_whence and their usesare(1M) **l_whence value** — **Use** SEEK_CUR — The l_start value is added to current file pointer address SEEK_SET — Thel_start value is added to byte 0 of the file	
(b)	**Exit Functions (2 Marks each)** Three functions terminate a program normally: _exit and _Exit, which return to the kernel immediately, and exit, which performs certain cleanup processing and then returns to the kernel. #include <stdlib.h> void exit(int status); #include <unistd.h> void _exit(intstatus); All three exit functions expect a single integer argument, called the exit status. **atexitFunction** With ISO C, a process can register up to 32 functions that are automatically called by exit. These are called exit handlers and are registered by calling the atexitfunction. #include <stdlib.h> intatexit(void (*func)(void)); Returns: 0 if OK, nonzero on error	2*3=6M
(c)	Any 8 properties of the parent that are inherited by thechild(4M) Any 4 differences between the parent and child(2M)	4+2=6
2(a)	**Hard Link(3M)** • The link function creates a new link for the existing file. • The prototype of the link function is `#include <unistd.h>` `int link(const char *cur_link, const char *new_link);` Returns: If successful, the link function returns 0.If unsuccessful, link returns –1.	3+3=6

Soft Link(3M)

- A symbolic link is created with the symlink.

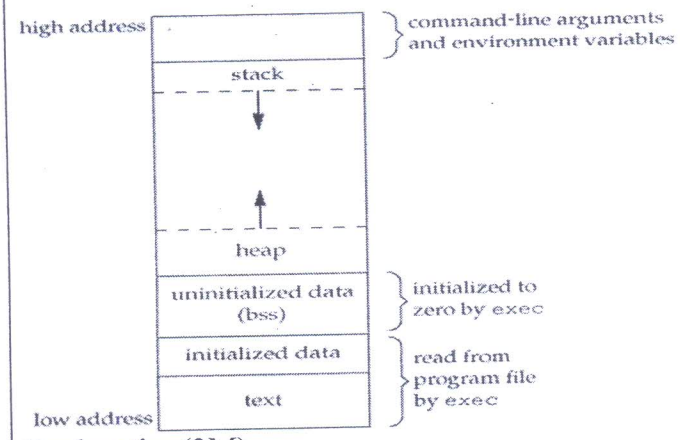
#include<unistd.h> #include<sys/types.h>

#include<sys/stat.h>

int*symlink*(const char *org_link, const char *sym_link);

Returns: If successful, the link function returns 0.If unsuccessful, link returns –1

(b) **MEMORY LAYOUT OF A C PROGRAM** 4+2=6M



Explanation(3M)

Diagram(2M)

(c) **fork Function (2M)** 2+2+2=6M

An existing process can create a new one by calling the fork function.

#include <unistd.h>

pid_t fork(void);

Returns: 0 in child, process ID of child in parent, 1 on error.

vfork*Function *(2M)

- ✓ The function vforkhas the same calling sequence and same return values as fork.
- ✓ The vfork function is intended to create a new process when the purpose of the new process is to exec a newprogram. The vfork function creates the new process, without copying the address space of the parent into the child, as the child won't reference that address space; the child simply calls exec (or exit) right after thevfork.

Wait(2M)

- Block, if all of its children are stillrunning
- ✓ Return immediately with the termination status of a child, if a child has terminated and is waiting for its termination status to befetched.

	#include <sys/wait.h> pid_t wait(int *statloc); return: process ID if OK, 0 (see later), or 1 on error.	
3(a)	**while: looping** The general syntax of this command is as follows:(1M) while condition is true do commands done Example Program(2M) **for: looping with a list** The general syntax of for loop is as follows (1M) for variable in list do commands done Example Program(2M)	1+2+1+2=6 M
(b)	#include <stdlib.h> int system(const char *cmdstring); Explaination:3M Example Program:3M	3+3= 6M
4(a)	**grep – searching for a pattern** It scans the file / input for a pattern and displays lines containing the pattern, the line numbers or filenames where the pattern occurs. It's a command from a special family in UNIX for handling search requirements. **grep*options pattern filename(s)*** **grep options** grep is one of the most important UNIX commands, and we must know the options that POSIX requires grep to support. Linux supports all of these options. -i ignores case for matching -v doesn't display lines matching expression -n displays line numbers along with lines -c displays count of number of occurrences -l displays list of filenames only -e exp specifies expression with this option -x matches pattern with entire line -f file takes patterns from file, one per line -E treats pattern as an extended RE -F matches multiple fixed strings	3+3=6M

(b)	**CHANGING USER IDs AND GROUP IDs** When our programs need additional privileges or need to gain access to resources that they currently aren't allowed to access, they need to change their user or group ID to an ID that has the appropriate privilege or access. Similarly, when our programs need to lower their privileges or prevent access to certain resources, they do so by changing either their user ID or group ID to an ID without the privilege or ability access to the resource #include <unistd.h>(2M) intsetuid(uid_tuid); intsetgid(gid_tgid); Both return: 0 if OK, 1 on error **setreuidand setregidFunctions (2M)** #include <unistd.h> intsetreuid(uid_truid, uid_teuid); intsetregid(gid_trgid, gid_tegid); Both return : 0 if OK, -1 on error **seteuid and setegid functions (2M)** #include <unistd.h> intseteuid(uid_tuid); intsetegid(gid_tgid); Both return : 0 if OK, 1 on error	2+2+2=6M

Signature of Course incharge **Signature of Module Coordinator** **Signature of HOD**

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

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

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 ODD SEMESTER

SET – B

USN										

Degree	:	**B.E**	**Semester**	:	**V A& B**
Branch	:	**Computer Science and Engineering**	**Course Code**	:	**18CS56**
Course Title	:	**UNIX PROGRAMMING**	**Date**	:	**19-11-2020**
Duration	:	**90 Minutes**	**Max Marks**	:	**30**

Note: Answer ONE full question from each part.

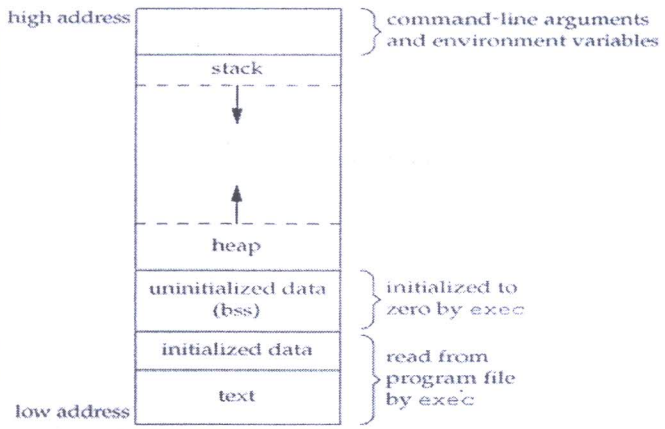
Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	Identify the attributes inherited by child process and attributes that are different from the parent	6	CO3	(Applying) K3
(b)	With a neat diagram, construct the memory layout of C program. In which segments are the automatic variables and dynamically created objects are stored?	6	CO3	(Applying) K3
(C)	Organize the following system calls in detail for usage in parent Child hierarchy : i)fork ii)vfork iii)wait()	6	CO3	(Applying) K3
	OR			
2(a)	Choose Symbolic Link and Hard link API and explain its usage	6	CO3	(Applying) K3
(b)	Choose exit, exit and atexit functions to develop their prototypes and identify usage.	6	CO3	(Applying) K3
(C)	Build the prototype for locking of files and records in detail.	6	CO3	(Applying) K3
	PART-B			
3(a)	Choose the purpose of grep command ? Explain all options	6	CO2	(Applying) K3
(b)	Organize the functions used for changing user ID and group ID	6	CO4	(Applying) K3
	OR			
4(a)	Identify shell features of while and for with syntax	6	CO2	(Applying) K3
(b)	Construct System function with an example program	6	CO4	(Applying) K3

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

II SESSIONAL TEST QUESTION PAPER 2020-21ODD SEMESTER

SET B **SCHEME AND SOLUTION**

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**CSE**	**Course Code**	:	**18CS56**
Course Titl	:	**UNIX Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
1(a)	Any 8 properties of the parent that are inherited by thechild(4M) Any 4 differences between the parent and child(2M)	4+2=6M
(b)	**MEMORY LAYOUT OF A C PROGRAM**  Explanation(4M) Diagram(2M)	4+2=6M
(c)	**fork Function (2M)** An existing process can create a new one by calling the fork function. #include <unistd.h> pid_t fork(void); Returns: 0 in child, process ID of child in parent, 1 on error. ***Vfork* Function *(2M)*** ✓ The function vforkhas the same calling sequence and same return values as fork. ✓ The vfork function is intended to create a new process when the purpose of the new process is to exec a newprogram. The vfork function creates the new process, without copying the address space of the parent into the	2+2+2=6M

	child, as the child won't reference that address space; the child simply calls exec (or exit) right after thevfork. **Wait**(2M) • Block, if all of its children are stillrunning ✓ Return immediately with the termination status of a child, if a child has terminated and is waiting for its termination status to befetched. #include <sys/wait.h> pid_t wait(int *statloc); return: process ID if OK, 0 (see later), or 1 on error.	
2(a)	**Hard Link(3M)** • The link function creates a new link for the existing file. • The prototype of the link function is `#include <unistd.h>` `int link(const char *cur_link, const char *new_link);` Returns: If successful, the link function returns 0.If unsuccessful, link returns –1. **Soft Link(3M)** • A symbolic link is created with the symlink. #include<unistd.h> #include<sys/types.h> #include<sys/stat.h> int*symlink*(const char *org_link, const char *sym_link); Returns: If successful, the link function returns 0.If unsuccessful, link returns –1	3+3=6
(b)	**Exit Functions (2 Marks each)** Three functions terminate a program normally: _exit and _Exit, which return to the kernel immediately, and exit, which performs certain cleanup processing and then returns to the kernel. #include <stdlib.h> void exit(int status); #include <unistd.h> void _exit(intstatus); All three exit functions expect a single integer argument, called the exit status. **atexitFunction** With ISO C, a process can register up to 32 functions that are automatically called by exit. These are called exit handlers and are registered by calling the atexitfunction.	2*3=6M

	#include <stdlib.h> intatexit(void (*func)(void)); Returns: 0 if OK, nonzero on error	
(c)	Multiple processes performs read and write operation on the same fileconcurrently. This provides a means for data sharing among processes, but it also renders difficulty for any process in determining when the other process can override data in afile. So, in order to overcome this drawback UNIX and POSIX standard support file lockingmechanism. ▪ File locking is applicable for regularfiles. ▪ Only a process can impose a write lock or read lock on either a portion of a file or on the entirefile. Prototype(2M) #include<fcntl.h> int***fcntl***(**int**fdesc,**int**cmd_flag,.....); The first argument specifies the filedescriptor. The second argument cmd_flag specifies what operation has to beperformed. If fcntl is used for file locking then it can valuesas F_SETLK sets a file lock, do not block if this cannot succeed immediately. F_SETLKW sets a file lock and blocks the process until the lock is acquired. F_GETLK queries as to which process locked a specified region offile. For file locking purpose, the third argument to fctnl is an address of a *struct flock* typevariable. This variable specifies a region of a file where lock is to be set, unset orqueried. **struct flock (2M)** **{** **short l_type; /* what lock to be set or to unlock file */** **short l_whence; /* Reference address for the next field */ off_t l_start;** **/*offset from the l_whence reference addr*/** **off_t l_len; /*how many bytes in the locked region*/** **pid_t l_pid; /*pid of a process which has locked the file*/** **};** ▪ The l_type field specifies the lock type to be set orunset. ▪ The possible values, which are defined in the <fcntl.h> header, and their uses are	2+2+1+1=6 M
3(a)	**grep – searching for a pattern** It scans the file / input for a pattern and displays lines containing the pattern, the line numbers or filenames where the pattern occurs. It's a command from a special family in UNIX for handling search requirements. **grep*options pattern filename(s)*** **grep options**	3+3=6M

	grep is one of the most important UNIX commands, and we must know the options that POSIX requires grep to support. Linux supports all of these options. -i ignores case for matching -v doesn't display lines matching expression -n displays line numbers along with lines -c displays count of number of occurrences -l displays list of filenames only -e exp specifies expression with this option -x matches pattern with entire line -f file takes patterns from file, one per line -E treats pattern as an extended RE -F matches multiple fixed strings	
(b)	**CHANGING USER IDs AND GROUP IDs** When our programs need additional privileges or need to gain access to resources that they currently aren't allowed to access, they need to change their user or group ID to an ID that has the appropriate privilege or access. Similarly, when our programs need to lower their privileges or prevent access to certain resources, they do so by changing either their user ID or group ID to an ID without the privilege or ability access to the resource #include <unistd.h>(2M) intsetuid(uid_tuid); intsetgid(gid_tgid); Both return: 0 if OK, 1 on error **setreuidand setregidFunctions (2M)** #include <unistd.h> intsetreuid(uid_truid, uid_teuid); intsetregid(gid_trgid, gid_tegid); Both return : 0 if OK, -1 on error **seteuid and setegid functions (2M)** #include <unistd.h> intseteuid(uid_tuid); intsetegid(gid_tgid); Both return : 0 if OK, 1 on error	2+2+2=6M
4(a)	**while: looping** The general syntax of this command is as follows:(1M) while condition is true do commands done Example Program(2M) **for: looping with a list** The general syntax of for loop is as follows (1M) for variable in list	1+2+1+2=6 M

	do commands done Example Program(2M)	
(b)	#include <stdlib.h> int system(const char *cmdstring); Explaination:3M Example Program:3M	

Signature of Course incharge

Signature of Module Coordinator

Signature of HOD

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

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 – 21ODDSEMESTER

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

SET – A

Degree : B.E **Semester: V**

Branch : Computer Science and Engineering **SubjectCode: 18CS56**

CourseTitle : UNIX PROGRAMMING **Date: 9-1-2021**

Duration : 90Minutes **MaxMarks: 30**

Note: Answer ONE full question from each part.

Q No.	Questions	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Utilize** FIFO and explain client server structure with a neat diagram	6	CO4	(Applying) K3
(b)	**Develop** IPC using: a. Streams pipe b. popen and pclose	6	CO4	(Applying) K3
(c)	**Build** the signal APIs with their prototypes and uses for sigprocmask, sigpending and sigaction.	6	CO5	(Applying) K3
	OR			
2(a)	**Construct** a code snippet that the parent sends "Hello world "message to child process through the pipe. Child on receiving this message should display it on output screen	6	CO4	(Applying) K3
(b)	**Make use of**relevant data structure and explain the message queue msgget and msgctlAPI's used for IPC.	6	CO4	(Applying) K3
(c)	**Build** the prototypes of all functions that are used to manipulate the signal sets	6	CO5	(Applying) K3
	PART-B			
3(a)	**Identify** the timer manipulation API's in POSIX.1b	6	CO5	(Applying) K3
(b)	**Identify** the ways in which the process can handle signals and also explain UNIX kernel support provided for handling signals.	6	CO5	(Applying) K3
	OR			
4(a)	**Identify** the characteristics of adaemon process.	6	CO5	(Applying) K3
(b)	**Interview** the daemon processes and explain with a neat diagram the error logging facility for a daemon process.	6	CO5	(Applying) K3

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
III SESSIONAL TEST QUESTION PAPER 2020-21ODDSEMESTER

SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**CSE**	**Course Code**	:	**18CS56**
Course Title	:	**UNIX Programming**	**Max Marks**	:	**30**

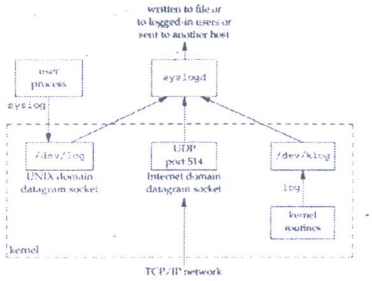
Q.NO.	POINTS	MARKS
1(a)	Diagram: 2M Explanation:4M ▪ FIFO's can be used to send data between a client and a server. If we have a server that is contacted by numerous clients, each client can write its request to a well-known FIFO that the server creates. Since there are multiple writers for the FIFO, the requests sent by the clients to the server need to be less than PIPE_BUF bytes insize. ▪ This prevents any interleaving of the client writes. The problem in using FIFOs for this type of client server communication is how to send replies back from the server to eachclient. ▪ A single FIFO can't be used, as the clients would never know when to read their response versus responses for other clients. One solution is for each client to send its process ID with the request. The server then creates a unique FIFO for each client, using a pathname based on the client's processID. ▪ For example, the server can create a FIFO with the name /vtu/ ser.XXXXX, where XXXXX is replaced with the client's process ID. This arrangement works, although it is impossible for the server to tell whether a client crashes. This causes the client-specific FIFOs to be left in the filesystem.	2+4=6M
(b)	**a.** Stream Pipes: (3M) **b.** Passing File Descriptors(3M)	3+3=6M
(c)	**a. Sigprocmask(2M)** A process initially inherits the parent's signal mask when it is created, but any pending signals for the parent process are not passed on. A process may query or set its signal mask via the sigprocmask API: `#include <signal.h>` `intsigprocmask(intcmd, constsigset_t *new_mask, sigset_t *old_mask);` Returns: 0 if OK, 1 on error **b. Sigpending(2M)** A process can query which signals are pending for it via the sigpending API: `#include<signal.h>` `intsigpending(sigset_t* sigmask);`	2+2+2=6

	Returns 0 if OK, -1 if fails. The sigpending API can be useful to find out whether one or more signals are pending for a process and to set up special signal handling methods for these signals before the process calls the sigprocmask API to unblock them. c. ***sigaction(2M)*** The sigaction API blocks the signal it is catching allowing a process to specify additional signals to be blocked when the API is handling a signal.Thesigaction API prototype is: `#include<signal.h>` `intsigaction(intsignal_num, structsigaction* action, structsigaction* old_action);`	
2(a)	#include <stdio.h> #include<fnctl.h> #include<unistd.h> int main(void) { intn, fd[2];pid_tpid; char line[MAXLINE]; if (pipe(fd) < 0) err_sys("pipeerror"); if ((pid = fork()) < 0) { err_sys("forkerror"); } else if (pid>0){ /* parent */ close(fd[0]); write(fd[1], "hello world\n", 12); }else{ /* child */ close(fd[1]); n = read(fd[0], line, MAXLINE); write(STDOUT_FILENO, line, n); } exit(0); }	3+3=6
(b)	Message Queue data structures (2M) msgget API(2M) msgctl(2M)	2+2+2=6M
(c)	`#include<signal.h>` `intsigemptyset(sigset_t* sigmask);` `intsigaddset(sigset_t* sigmask, constintsig_num);` `intsigdelset(sigset_t* sigmask, constintsig_num);` `intsigfillset(sigset_t* sigmask);` `intsigismember(constsigset_t* sigmask, constintsig_num);`	1mark each=6M
3(a)	#include<signal.h>	1mark each=6M

	#include<time.h> inttimer_create(clockid_t clock, structsigevent* spec, timer_t* timer_hdrp); inttimer_settime(timer_ttimer_hdr, int flag, structitimerspec* val, structitimerspec* old); inttimer_gettime(timer_ttimer_hdr, structitimerspec* old); inttimer_getoverrun(timer_ttimer_hdr); inttimer_delete(timer_ttimer_hdr);	
(b)	The ways in which the process can handle signals(3M) • Accept the **default action** of the signal, which for most signals will terminate the process. • **Ignore** the signal. The signal will be discarded and it has no effect whatsoever on the recipient process. • Invoke a **user-defined** function. The function is known as a signal handler routine and the signal is said to be ***caught*** when this function is called. **The unix kernel support of signals(3M)** • When a signal is generated for a process, the kernel will set the corresponding signal flag in the process table slot of the recipientprocess. • If the recipient process is asleep, the kernel will awaken the process by schedulingit. • When the recipient process runs, the kernel will check the process U-area that contains an array of signal handlingspecifications. • If array entry contains a zero value, the process will accept the default action of thesignal. • If array entry contains a 1 value, the process will ignore the signal and kernel will discardit. • If array entry contains any other value, it is used as the function pointer for a user-defined signal handler routine.	3+3= 6M
4(a)	**Characteristics of daemons are:** ▪ Daemons run inbackground. ▪ Daemons have super-userprivilege. ▪ Daemons don't have controllingterminal. ▪ Daemons are session and groupleaders.	3+3=6M
(b)	Daemon Process(1M) : Daemons are processes that live for a long time. They are often started when the system is bootstrapped and terminate only when the system is shut down. **Error logging** Diagram : 1M Explanation: 4M ▪ One problem a daemon has is how to handle error messages. It can't simply write to standard error,	1+1+4=6M

since it shouldn't have a controlling terminal. We don't want all the daemons writing to the console device, since on many workstations, the console device runs a windowing system. A central daemon error-logging facility is required.

Figure 13.2. The BSD syslog facility



There are three ways to generate log messages:

- Kernelroutinescancallthelogfunction.Thesemessagescanbereadbyanyuserprocessthatopensand reads the /dev/klogdevice.
- Most user processes (daemons) call the syslog(3) function to generate log messages. This causes the message to be sent to the UNIX domain datagram socket/dev/log.

  A user process on this host, or on some other host that is connected to this host by a TCP/IP network, can send log messages to UDP port 514. Note that the syslog function never generates these UDP datagrams: they require explicit network programming by the process generating the logmessage

Signature of Course incharge **Signature of Module Coordinator** **Signature of 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 – 21ODD SEMESTER

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

SET-B

Degree : B.E **Semester: V**
Branch : Computer Science and Engineering **SubjectCode: 18CS56**
CourseTitle : UNIX PROGRAMMING **Date: 4-2-2021**
Duration : 90Minutes **MaxMarks: 30**

Note: Answer ONE full question from each part.

Q No.	Questions	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Construct** a code snippet that the parent sends "Hello world "message to child process through the pipe. Child on receiving this message should display it on output screen	6	CO4	(Applying) K3
(b)	**Make use of**relevant data structure and explain the message queue msgget and msgctl API's used for IPC.	6	CO4	(Applying) K3
(c)	**Build** the API for shmget, shmctl, shmat and shmdt functions.	6	CO5	(Applying) K3
	OR			
2(a)	**Develop** IPC using: a. Co-Processes d. popen and pclose	6	CO4	(Applying) K3
(b)	**Utilize** FIFO and explain client server structure with a neat diagram	6	CO4	(Applying) K3
(c)	**Build** the signal APIs with their prototypes and uses for sigprocmask, sigpending and sigaction.	6	CO5	(Applying) K3
	PART-B			
3(a)	**Identify** the coding rules of a daemon process with an example	6	CO5	(Applying) K3
(b)	**Identify** the timer manipulation API's in POSIX.1b	6	CO5	(Applying) K3
	OR			
4(a)	**Interview** the daemon processes and explain with a neat diagram the error logging facility for a daemon process.	6	CO5	(Applying) K3
(b)	**Identify** the ways in which the process can handle signals and also explain UNIX kernel support provided for handling signals	6	CO5	(Applying) K3

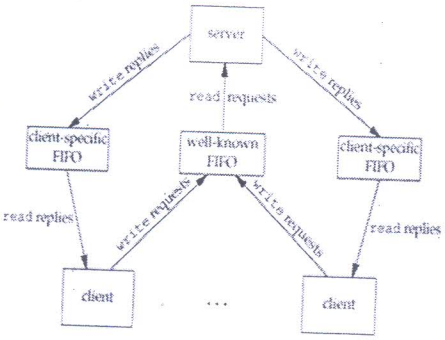
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
III SESSIONAL TEST QUESTION PAPER 2020-21ODDSEMESTER

SET B **SCHEME AND SOLUTION**

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**CSE**	**Course Code**	:	**18CS56**
Course Title	:	**UNIX Programming**	**Max Marks**	:	**30**

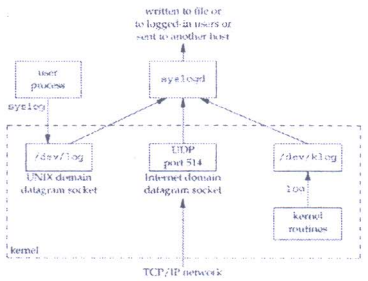
Q.NO.	POINTS	MARKS
1(a)	Program:6M #include <stdio.h> #include<fnctl.h> #include<unistd.h> int main(void) { intn, fd[2];pid_tpid; char line[MAXLINE]; if (pipe(fd) < 0) err_sys("pipeerror"); if ((pid = fork()) < 0) { err_sys("forkerror"); } else if (pid>0){ /* parent */ close(fd[0]); write(fd[1], "hello world\n", 12); }else{ /* child */ close(fd[1]); n = read(fd[0], line, MAXLINE); write(STDOUT_FILENO, line, n); } exit(0); }	6M
(b)	Message Queue data structures (2M) msgget API(2M) msgctl(2M)	2+2+2=6M
(c)	The API for shared memory shmget(2M) .#include <sys/shm.h> intshmget(key_t*key*, size_t*size*, int*flag*); Returns: shared memory ID if OK, 1 on error Shmctl(2M) #include <sys/shm.h> intshmctl(int*shmid*, int*cmd*, structshmid_ds **buf*);	2+2+1+1=6

	Returns: 0 if OK, 1 on error shmat (1M) #include <sys/shm.h> void *shmat(int*shmid*, const void **addr*, int*flag*); Returns: pointer to shared memory segment if OK, 1 on error and shmdt (1M) #include <sys/shm.h> intshmdt(void **addr*); Returns: 0 if OK, 1 on error	
2(a)	***a. COPROCESSES (2M)*** A UNIX system filter is a program that reads from standard input and writes to standard output. Filters are normally connected linearly in shell pipelines. A filter becomes a coprocess when the same program generates the filter's input and reads the filter's output. A coprocess normally runs in the background from a shell, and its standard input and standard output are connected to another program using apipe. The process creates two pipes: one is the standard input of the coprocess, and the other is the standard output of the coprocess. **b. <u>popenAND pcloseFUNCTIONS(4M)</u>** Since a common operation is to create a pipe to another process, to either read its output or send it input, the standard I/O library has historically provided the popen and pclose functions. These two functions handle all the dirty work that we've been doing ourselves: creating a pipe, forking a child, closing the unused ends of the pipe, executing a shell to run the command, and waiting for the command to terminate. `#include <stdio.h>` `FILE *popen(const char *cmdstring, const char *type);` Returns: file pointer if OK, NULLon error `intpclose(FILE *fp);` Returns: termination status of cmdstring, or 1 on error The function popendoes a forkand execto execute the cmdstring, and returns a standard I/O file pointer. If type is "r", the file pointer is connected to the standard output of cmdstring	2+4=6

(b)	Diagram: 2M Explanation:4M ▪  ▪ FIFO's can be used to send data between a client and a server. If we have a server that is contacted by numerous clients, each client can write its request to a well-known FIFO that the server creates. Since there are multiple writers for the FIFO, the requests sent by the clients to the server need to be less than PIPE_BUF bytes insize. ▪ This prevents any interleaving of the client writes. The problem in using FIFOs for this type of client server communication is how to send replies back from the server to eachclient. ▪ A single FIFO can't be used, as the clients would never know when to read their response versus responses for other clients. One solution is for each client to send its process ID with the request. The server then creates a unique FIFO for each client, using a pathname based on the client's processID. ▪ For example, the server can create a FIFO with the name /vtu/ ser.XXXXX, where XXXXX is replaced with the client's process ID. This arrangement works, although it is impossible for the server to tell whether a client crashes. This causes the client-specific FIFOs to be left in the filesystem.	2+4=6M
(c)	**a. Sigprocmask(2M)** A process initially inherits the parent's signal mask when it is created, but any pending signals for the parent process are not passed on. A process may query or set its signal mask via the sigprocmask API: `#include <signal.h>` `intsigprocmask(intcmd, constsigset_t *new_mask, sigset_t *old_mask);` Returns: 0 if OK, 1 on error **b. Sigpending(2M)** A process can query which signals are pending for it via the sigpending API: `#include<signal.h>` `intsigpending(sigset_t* sigmask);` Returns 0 if OK, -1 if fails. The sigpending API can be useful to find out whether one or more signals are pending for a process and to set up special signal handling methods for these signals before the process calls the sigprocmask API to unblock them. ***c. sigaction(2M)*** The sigaction API blocks the signal it is catching allowing a process to specify additional signals to be blocked when the API is handling a signal.Thesigaction API prototype is: `#include<signal.h>`	2+2+2=6M

	`intsigaction(intsignal_num, structsigaction* action, structsigaction* old_action);`	
3(a)	***Coding rules*** Some basic rules to coding a daemon prevent unwanted interactions from happening. • Call umaskto set the file mode creation mask to 0. ▪ Call fork and have the parent exit. ▪ Call setsidto create a new session. ▪ Change the current working directory to the root directory. ▪ Unneeded file descriptors should be closed. ▪ Some daemons open file descriptors 0, 1, and 2 to /dev/null so that any library routines that try to read from standard input or write to standard output or standard error will have no effect.	1mark each=6M
(b)	#include<signal.h> #include<time.h> inttimer_create(clockid_t clock, structsigevent* spec, timer_t* timer_hdrp); inttimer_settime(timer_ttimer_hdr, int flag, structitimerspec* val, structitimerspec* old); inttimer_gettime(timer_ttimer_hdr, structitimerspec* old); inttimer_getoverrun(timer_ttimer_hdr); inttimer_delete(timer_ttimer_hdr);	1 Mark each= 6M
4(a)	Daemon Process(1M) : Daemons are processes that live for a long time. They are often started when the system is bootstrapped and terminate only when the system is shut down. **Error logging** Diagram : 1M Explanation: 4M ▪ One problem a daemon has is how to handle error messages. It can't simply write to standard error, since it shouldn't have a controlling terminal. We don't want all the daemons writing to the console device, since on many workstations, the console device runs a windowing system. A central daemon error-logging facility is required.	1+1+4=6M

Figure 13.2. The BSD syslog facility



There are three ways to generate log messages:

- Kernelroutinescancallthelogfunction.Thesemessagescanbereadbyanyuserprocessthatopensa nd reads the /dev/klogdevice.
- Most user processes (daemons) call the syslog(3) function to generate log messages. This causes the message to be sent to the UNIX domain datagram socket/dev/log.

A user process on this host, or on some other host that is connected to this host by a TCP/IP network, can send log messages to UDP port 514. Note that the syslog function never generates these UDP datagrams: they require explicit network programming by the process generating the logmessage

(b) The ways in which the process can handle signals(3M) — 3+3=6M

- Accept the **default action** of the signal, which for most signals will terminate the process.
- **Ignore** the signal. The signal will be discarded and it has no effect whatsoever on the recipient process.
- Invoke a **user-defined** function. The function is known as a signal handler routine and the signal is said to be ***caught*** when this function is called.

The unix kernel support of signals(3M)

- When a signal is generated for a process, the kernel will set the corresponding signal flag in the process table slot of the recipientprocess.
- If the recipient process is asleep, the kernel will awaken the process by schedulingit.
- When the recipient process runs, the kernel will check the process U-area that contains an array of signal handlingspecifications.
- If array entry contains a zero value, the process will accept the default action of thesignal.
- If array entry contains a 1 value, the process will ignore the signal and kernel will discardit.

  If array entry contains any other value, it is used as the function pointer for a user-defined signal handler routine

Signature of Course incharge **Signature of Module Coordinator** **Signature of 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 and Engineering

YEAR / SEMESTER	III /V B
COURSE TITLE	UNIX PROGRAMMIN
COURSE CODE	18CS56
ACADEMIC YEAR	2020-2021

SI. NO.	USN	NAME	IA1 (30)	IA2 (30)	IA3 (30)/ IMP	ASSN 1 (10)	ASSN 2 (10)	ASSN 3 (10)	AVG IA (30)	ASSN AVG (10)	TOTAL (40)	SIGN
1	1KS18CS063	NITISH KUMAR.M.R	23	24	8	9	10	10	19.0	10.0	29	
2	1KS18CS064	NOOR SUMAIYA	28	30	19	8	10	10	26.0	10.0	36	
3	1KS18CS065	P SAI RAM	28	30	27	7	10	7	29.0	8.0	37	
4	1KS18CS066	PAVAN P	29	30	25	9	10	10	28.0	10.0	38	
5	1KS18CS067	POOJASHREE K	30	30	30	9	10	10	30.0	10.0	40	
6	1KS18CS069	PRANAV M S	24	28	15	9	7	10	23.0	9.0	32	
7	1KS18CS070	PRATEEK HAVALE	28	30	29	10	10	10	29.0	10.0	39	
8	1KS18CS071	PRAVEEN KUMAR K	29	30	30	9	10	10	30.0	10.0	40	
9	1KS18CS072	PREETHI K	29	30	12	7	7	7	24.0	7.0	31	
10	1KS18CS073	PUJARI VISHNU PRIYA	29	30	30	10	10	10	30.0	10.0	40	
11	1KS18CS074	PULLUR PAVAN KUMAR	29	30	28	9	10	10	29.0	10.0	39	
12	1KS18CS075	R DEKSHITHA	30	30	30	9	10	10	30.0	10.0	40	
13	1KS18CS076	R PRATIKSHA	30	30	30	9	10	10	30.0	10.0	40	
14	1KS18CS077	RAMYA R	29	30	29	8	10	7	30.0	9.0	39	
15	1KS18CS078	RAHUL.P	28	30	16	4	7	7	25.0	6.0	31	
16	1KS18CS079	RAIPALLE SHREYAA	30	28	20	6	10	10	26.0	9.0	35	
17	1KS18CS080	RAKSHA S	30	30	16	5	7	10	26.0	8.0	34	
18	1KS18CS081	RAKSHITH KUMAR.N	30	30	22	9	10	10	28.0	10.0	38	
19	1KS18CS082	REKHA N C	30	30	29	9	10	10	30.0	10.0	40	
20	1KS18CS083	RITHANA.N.RAJ	24	30	14	4	10	10	23.0	8.0	31	
21	1KS18CS084	RUBA ABDUL RAHMAN	30	30	13	5	7	7	25.0	7.0	32	
22	1KS18CS086	SAMHITHA	30	30	19	9	10	10	27.0	10.0	37	
23	1KS18CS087	SANDEEP KUMAR	29	30	26	5	10	7	29.0	8.0	37	
24	1KS18CS088	SAURAV KUMAR	29	30	0	9	10	0	20.0	7.0	27	
25	1KS18CS089	SAURAV S MAKAM	30	30	20	3	7	7	27.0	6.0	33	
26	1KS18CS090	SHALINI S	30	30	30	9	10	10	30.0	10.0	40	
27	1KS18CS091	SHASHANK G	29	30	30	10	10	10	30.0	10.0	40	
28	1KS18CS092	SHASHANK MISHRA	26	30	18	8	10	10	25.0	10.0	35	
29	1KS18CS093	SHIVANGI SRIVASTAVA	30	30	27	8	7	7	29.0	8.0	37	
30	1KS18CS094	SHIVA PRAKASH T	24	30	24	10	10	10	26.0	10.0	36	
31	1KS18CS095	SHUBHASHINI.R	27	30	30	9	10	7	29.0	9.0	38	
32	1KS18CS096	SINDU A S	30	30	27	10	10	10	29.0	10.0	39	

SI. NO.	USN	NAME	IA1 (30)	IA2 (30)	IA3 (30)/ IMP	ASSN 1 (10)	ASSN 2 (10)	ASSN 3 (10)	AVG IA (30)	ASSN AVG (10)	TOTAL (40)	SIGN
33	1KS18CS097	SOURABH SANTOSH KAMBLE	28	24	29	9	10	10	27.0	10.0	37	
34	1KS18CS098	SRI CHANDANA P	30	30	24	10	10	10	28.0	10.0	38	
35	1KS18CS099	SRIVIDYA H R	30	24	25	10	10	10	27.0	10.0	37	
36	1KS18CS100	SUBRAMANYA N	30	28	18	5	7	7	26.0	7.0	33	
37	1KS18CS101	SUDHAKAR YASWANTH	29	30	28	9	10	10	29.0	10.0	39	
38	1KS18CS102	SUDHANSHU JOSHI	29	30	30	9	10	10	30.0	10.0	40	
39	1KS18CS103	SUJAY G S	30	30	30	9	10	10	30.0	10.0	40	
40	1KS18CS104	SUNAINA NAYAK	29	30	30	10	10	10	30.0	10.0	40	
41	1KS18CS105	SURAJ C JAWOOR	29	30	9	5	7	7	23.0	7.0	30	
42	1KS18CS106	SUSHMITHA S	28	30	16	9	10	10	25.0	10.0	35	
43	1KS18CS107	SWETHA BIJANAPALLI	28	30	23	7	10	10	27.0	9.0	36	
44	1KS18CS108	THAMMINENI HEMANTH CHOWDARY	29	30	28	9	10	10	29.0	10.0	39	
45	1KS18CS109	THIRUMALAI SHAKTIVEL C	30	30	30	10	10	10	30.0	10.0	40	
46	1KS18CS110	VAISHAK P	27	30	30	10	10	10	29.0	10.0	39	
47	1KS18CS111	VARIDHI MADHURANATH	28	30	30	9	10	10	30.0	10.0	40	
48	1KS18CS112	VEDAVEDYA B H	28	30	9	8	10	10	23.0	10.0	33	
49	1KS18CS113	VEERA SREENIDHI.R	29	27	23	7	7	10	27.0	8.0	35	
50	1KS18CS114	VENKATESH M N	28	30	9	4	10	10	23.0	8.0	31	
51	1KS18CS115	VIJAY.N.S	28	28	19	4	10	10	25.0	8.0	33	
52	1KS18CS116	VIJAYASHREE.N.R	30	30	30	5	10	10	30.0	9.0	39	
53	1KS18CS117	VIJETHA	30	30	18	8	10	10	26.0	10.0	36	
54	1KS18CS118	VIINOD H MALALI	30	30	30	10	10	10	30.0	10.0	40	
55	1KS18CS119	VISHNUPRIYA D	30	30	23	7	10	10	28.0	9.0	37	
56	1KS18CS120	VYJAYANTHI K S	29	30	30	9	7	7	30.0	8.0	38	
57	1KS18CS121	YASHWANTH.K	24	30	28	9	10	10	28.0	10.0	38	
58	1KS18CS122	YOGITA RAIKAR	30	30	29	10	10	10	30.0	10.0	40	
59	1KS18CS123	ZAINA KHAN	30	30	23	9	10	10	28.0	10.0	38	
60	1KS18CS124	SHEWANI CHIB	29	30	29	6	10	10	30.0	9.0	39	
61	1KS17CS015	B R GAGAN	30	28	30	9	10	7	30.0	9.0	39	
62	1KS18CS125	R SOUMYA	30	30	30	10	10	10	30.0	10.0	40	
63	1KS18CS126	RAYYAAN MOHIADDIN	23	30	0	7	0	0	18.0	3.0	21	
64	1KS18CS127	ARVIND PATHAK	29	28	27	8	10	10	28.0	10.0	38	
65	1KS18CS128	BHAGYASHREE.V	30	30	24	7	10	10	28.0	9.0	37	
66	1KS18CS129	BI BI AYESHA	30	30	30	3	10	10	30.0	8.0	38	
67	1KS18CS130	LIKITHA.S	30	24	29	7	10	10	28.0	9.0	37	
68	1KS18CS131	SHALINI.K.P	30	30	30	7	10	10	30.0	9.0	39	
69	1KS19CS401	ARPITHA.G	30	28	16	4	10	10	25.0	8.0	33	
70	1KS19CS404	BHAVYASHREE.R	30	28	15	4	10	7	25.0	7.0	32	
71	1KS19CS413	SAHANA.V	30	30	24	7	10	7	28.0	8.0	36	
72	1KS19CS414	Y.MRUDULA JAIN	30	30	18	7	10	7	26.0	8.0	34	

15/2/2021

(Sruthi i/c)

15/2/2021

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 and Engineering

YEAR / SEMESTER	III /V B
COURSE TITLE	UNIX PROGRAMMING
COURSE CODE	18CS56
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	12 Marks	18 Marks	
1	1KS18CS063	NITISH KUMAR.M.R				6	6	6	5				23	18	5	
2	1KS18CS064	NOOR SUMAIYA				6	5	5	6	6			28	16	12	
3	1KS18CS065	P SAI RAM				6	6	5	5	6			28	17	11	
4	1KS18CS066	PAVAN P				6	6	5	6	6			29	17	12	ABS
5	1KS18CS067	POOJASHREE K				6	6	6	6	6			30	18	12	
6	1KS18CS069	PRANAV M S	5	6	5				4	4			24	16	8	
7	1KS18CS070	PRATEEK HAVALE				6	6	6	4	**6**			28	18	10	
8	1KS18CS071	PRAVEEN KUMAR K				6	6	6	5	6			29	18	11	
9	1KS18CS072	PREETHI K				6	5	6	6	**6**			29	17	12	
10	1KS18CS073	PUJARI VISHNU PRIYA				6	6	5	6	6			29	17	12	
11	1KS18CS074	PULLUR PAVAN KUMAR	6	6	5				6	6			29	17	12	
12	1KS18CS075	R DEKSHITHA				6	6	6	6	6			30	18	12	
13	1KS18CS076	R PRATIKSHA				6	6	6	6	6			30	18	12	
14	1KS18CS077	RAMYA R				6	6	6	5	6			29	18	11	
15	1KS18CS078	RAHUL.P				4	6	6	6	6			28	16	12	
16	1KS18CS079	RAIPALLE SHREYAA				6	6	6	6	6			30	18	12	
17	1KS18CS080	RAKSHA S				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	12 Marks	18 Marks
18	1KS18CS081	RAKSHITH KUMAR.N				6	6	6	6	6			30	18	12
19	1KS18CS082	REKHA N C				6	6	6	6	6			30	18	12
20	1KS18CS083	RITHANA.N.RAJ				6	6	6	6				24	18	6
21	1KS18CS084	RUBA ABDUL RAHMAN				6	6	6	6	6			30	18	12
22	1KS18CS086	SAMHITHA				6	6	6	6	6			30	18	12
23	1KS18CS087	SANDEEP KUMAR				6	6	6	5	6			29	18	11
24	1KS18CS088	SAURAV KUMAR				6	6	6			6	5	29	18	11
25	1KS18CS089	SAURAV S MAKAM				6	6	6	6	6			30	18	12
26	1KS18CS090	SHALINI S				6	6	6	6	6			30	18	12
27	1KS18CS091	SHASHANK G	6	6	6				6	5			29	18	11
28	1KS18CS092	SHASHANK MISHRA				6	4	4	6	6			26	14	12
29	1KS18CS093	SHIVANGI SRIVASTAVA				6	6	6	6	6			30	18	12
30	1KS18CS094	SHIVA PRAKASH T	1	6	6						5	6	24	13	11
31	1KS18CS095	SHUBHASHINI.R				6	5	6	4	6			27	17	10
32	1KS18CS096	SINDU A S				6	6	6	6	6			30	18	12
33	1KS18CS097	SOURABH SANTOSH KAMBLE				5	6	6			6	5	28	17	11
34	1KS18CS098	SRI CHANDANA P				6	6	6	6	6			30	18	12
35	1KS18CS099	SRIVIDYA H R				6	6	6	6	6			30	18	12
36	1KS18CS100	SUBRAMANYA N				6	6	6	6	6			30	18	12
37	1KS18CS101	SUDHAKAR YASWANTH				6	6	6	6	5			29	18	11
38	1KS18CS102	SUDHANSHU JOSHI	6	6	6						6	5	29	18	11
39	1KS18CS103	SUJAY G S				6	6	6	6	6			30	18	12
40	1KS18CS104	SUNAINA NAYAK				6	6	6	5	6			29	18	11
41	1KS18CS105	SURAJ C JAWOOR				6	6	6	5	6			29	18	11
42	1KS18CS106	SUSHMITHA S				6	6	6	4	6			28	18	10
43	1KS18CS107	SWETHA BIJANAPALLI				6	6	6	5	5			28	18	10
44	1KS18CS108	THAMMINENI HEMANTH CHOWDARY				6	6	6	6	5			29	18	11
45	1KS18CS109	THIRUMALAI SHAKTIVEL C	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	12 Marks	18 Marks
46	1KS18CS110	VAISHAK P	5	5	5						6	6	27	15	12
47	1KS18CS111	VARIDHI MADHURANATH				6	6	6	6	4			28	18	10
48	1KS18CS112	VEDAVEDYA B H				6	6	6	4	6			28	18	10
49	1KS18CS113	VEERA SREENIDHI.R				6	6	6	5	6			29	18	11
50	1KS18CS114	VENKATESH M N	6	6	6				5	5			28	18	10
51	1KS18CS115	VIJAY.N.S				6	6	6	5	5			28	18	10
52	1KS18CS116	VIJAYASHREE.N.R				6	6	6	6	6			30	18	12
53	1KS18CS117	VIJETHA				6	6	6	6	6			30	18	12
54	1KS18CS118	VIINOD H MALALI				6	6	6	6	6			30	18	12
55	1KS18CS119	VISHNUPRIYA D				6	6	6	6	6			30	18	12
56	1KS18CS120	VYJAYANTHI K S				6	6	6	5	6			29	18	11
57	1KS18CS121	YASHWANTH.K				6	6	6	6			6	24	18	6
58	1KS18CS122	YOGITA RAIKAR				6	6	6	6	6			30	18	12
59	1KS18CS123	ZAINA KHAN				6	6	6	6	6			30	18	12
60	1KS18CS124	SHEWANI CHIB				6	6	6	5	6			29	18	11
61	1KS17CS015	B R GAGAN	6	6	6						6	6	30	18	12
62	1KS18CS125	R SOUMYA				6	6	6	6	6			30	18	12
63	1KS18CS126	RAYYAAN MOHIADDIN		4		6	6		5	6			23	12	11
64	1KS18CS127	ARVIND PATHAK				6	6	6	6	5			29	18	11
65	1KS18CS128	BHAGYASHREE.V				6	6	6			6	6	30	18	12
66	1KS18CS129	BI BI AYESHA				6	6	6			6	6	30	18	12
67	1KS18CS130	LIKITHA.S				6	6	6			6	6	30	18	12
68	1KS18CS131	SHALINI.K.P				6	6	6			6	6	30	18	12
69	1KS19CS401	ARPITHA.G				6	6	6	6	6			30	18	12
70	1KS19CS404	BHAVYASHREE.R				6	6	6			6	6	30	18	12
71	1KS19CS413	SAHANA.V				6	6	6	6	6			30	18	12
72	1KS19CS414	Y.MRUDULA JAIN				6	6	6	6	6			30	18	12

ABS

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 and Engineering

YEAR / SEMESTER	III /V B
COURSE TITLE	UNIX PROGRAMMING
COURSE CODE	18CS56
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 Marks	6 Marks	6 Marks
1	1KS18CS063	NITISH KUMAR.M.R	6	6							6	6	24	12	6	6
2	1KS18CS064	NOOR SUMAIYA	6	6	6				6	6			30	18	6	6
3	1KS18CS065	P SAI RAM	6	6	6				6	6			30	18	6	6
4	1KS18CS066	PAVAN P	6	6	6				6	6			30	18	6	6
5	1KS18CS067	POOJASHREE K	6	6	6				6	6			30	18	6	6
6	1KS18CS069	PRANAV M S				6	4	6			6	6	28	16	6	6
7	1KS18CS070	PRATEEK HAVALE	6	6	6				6	6			30	18	6	6
8	1KS18CS071	PRAVEEN KUMAR K	6	6	6				6	6			30	18	6	6
9	1KS18CS072	PREETHI K	6	6	6				6	6			30	18	6	6
10	1KS18CS073	PUJARI VISHNU PRIYA	6	6	6						6	6	30	18	6	6
11	1KS18CS074	PULLUR PAVAN KUMAR	6	6	6				6	6			30	18	6	6
12	1KS18CS075	R DEKSHITHA	6	6	6				6	6			30	18	6	6
13	1KS18CS076	R PRATIKSHA	6	6	6				6	6			30	18	6	6
14	1KS18CS077	RAMYA R	6	6	6				6	6			30	18	6	6
15	1KS18CS078	RAHUL.P	6	6	6				6	6			30	18	6	6
16	1KS18CS079	RAIPALLE SHREYAA	4	6	6				6	6			28	16	6	6
17	1KS18CS080	RAKSHA S	6	6	6				6	6			30	18	6	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			
			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 Marks	6 Marks	6 Marks	
18	1KS18CS081	RAKSHITH KUMAR.N	6	6	6				6	6			30	18	6	6	
19	1KS18CS082	REKHA N C	6	6	6				6	6			30	18	6	6	
20	1KS18CS083	RITHANA.N.RAJ	6	6	6						6	6	30	18	6	6	
21	1KS18CS084	RUBA ABDUL RAHMAN	6	6	6				6	6			30	18	6	6	
22	1KS18CS086	SAMHITHA				6	6	6			6	6	30	18	6	6	
23	1KS18CS087	SANDEEP KUMAR	6	6	6						6	6	30	18	6	6	
24	1KS18CS088	SAURAV KUMAR	6	6	6				6	6			30	18	6	6	
25	1KS18CS089	SAURAV S MAKAM	6	6	6				6	6			30	18	6	6	
26	1KS18CS090	SHALINI S	6	6	6				6	6			30	18	6	6	
27	1KS18CS091	SHASHANK G	6	6	6				6	6			30	18	6	6	
28	1KS18CS092	SHASHANK MISHRA	6	6	6				6	6			30	18	6	6	
29	1KS18CS093	SHIVANGI SRIVASTAVA				6	6	6	6	6			30	18	6	6	
30	1KS18CS094	SHIVA PRAKASH T	6	6	6				6	6			30	18	6	6	
31	1KS18CS095	SHUBHASHINI.R	6	6	6				6	6			30	18	6	6	
32	1KS18CS096	SINDU A S				6	6	6	6	6			30	18	6	6	
33	1KS18CS097	SOURABH SANTOSH KAMBLE		6	6				6	6			24	12	6	6	
34	1KS18CS098	SRI CHANDANA P	6	6	6				6	6			30	18	6	6	
35	1KS18CS099	SRIVIDYA H R	6		6		6				6	6	24	12	6	6	
36	1KS18CS100	SUBRAMANYA N	4	6	6				6	6			28	16	6	6	
37	1KS18CS101	SUDHAKAR YASWANTH											0	0	6	6	AB
38	1KS18CS102	SUDHANSHU JOSHI	6	6	6				6	6			30	18	6	6	
39	1KS18CS103	SUJAY G S	6	6	6				6	6			30	18	6	6	
40	1KS18CS104	SUNAINA NAYAK	6	6	6				6	6			30	18	6	6	
41	1KS18CS105	SURAJ C JAWOOR	6	6	6				6	6			30	18	6	6	
42	1KS18CS106	SUSHMITHA S	6	6	6				6	6			30	18	6	6	
43	1KS18CS107	SWETHA BIJANAPALLI				6	6	6			6	6	30	18	6	6	
44	1KS18CS108	THAMMINENI HEMANTH CHOWDARY	6	6	6				6	6			30	18	6	6	
45	1KS18CS109	THIRUMALAI SHAKTIVEL C	6	6	6				6	6			30	18	6	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		
			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 Marks	6 Marks	6 Marks
46	1KS18CS110	VAISHAK P	6	6	6				6	6			30	18	6	6
47	1KS18CS111	VARIDHI MADHURANATH				6	6	6			6	6	30	18	6	6
48	1KS18CS112	VEDAVEDYA B H	6	6	6				6	6			30	18	6	6
49	1KS18CS113	VEERA SREENIDHI.R	3	6	6				6	6			27	15	6	6
50	1KS18CS114	VENKATESH M N	6	6	6				6	6			30	18	6	6
51	1KS18CS115	VIJAY.N.S	4	6	6				6	6			28	16	6	6
52	1KS18CS116	VIJAYASHREE.N.R				6	6	6			6	6	30	18	6	6
53	1KS18CS117	VIJETHA	6	6	6				6	6			30	18	6	6
54	1KS18CS118	VIINOD H MALALI	6	6	6				6	6			30	18	6	6
55	1KS18CS119	VISHNUPRIYA D	6	6	6				6	6			30	18	6	6
56	1KS18CS120	VYJAYANTHI K S	6	6	6				6	6			30	18	6	6
57	1KS18CS121	YASHWANTH.K	6	6	6				6	6			30	18	6	6
58	1KS18CS122	YOGITA RAIKAR	6	6	6				6	6			30	18	6	6
59	1KS18CS123	ZAINA KHAN	6	6	6				6	6			30	18	6	6
60	1KS18CS124	SHEWANI CHIB	6	6	6				6	6			30	18	6	6
61	1KS17CS015	B R GAGAN	4	6	6				6	6			28	16	6	6
62	1KS18CS125	R SOUMYA	6	6	6				6	6			30	18	6	6
63	1KS18CS126	RAYYAAN MOHIADDIN	6	6	6				6	6			30	18	6	6
64	1KS18CS127	ARVIND PATHAK	4	6	6				6	6			28	16	6	6
65	1KS18CS128	BHAGYASHREE.V	6	6	6				6	6			30	18	6	6
66	1KS18CS129	BI BI AYESHA	6	6	6				6	6			30	18	6	6
67	1KS18CS130	LIKITHA.S	6	6	6				6	0			24	18	6	
68	1KS18CS131	SHALINI.K.P	6	6	6				6	6			30	18	6	6
69	1KS19CS401	ARPITHA.G	4	6	6				6	6			28	16	6	6
70	1KS19CS404	BHAVYASHREE.R	4	6	6				6	6			28	16	6	6
71	1KS19CS413	SAHANA.V				6	6	6	6	6			30	18	6	6
72	1KS19CS414	Y.MRUDULA JAIN				6	6	6	6	6			30	18	6	6

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

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

Department of Computer Science and Engineering

YEAR / SEMESTER	III /V B
COURSE TITLE	UNIX PROGRAMMING
COURSE CODE	18CS56
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		
			CO4	CO4	CO5	CO4	CO4	CO5	CO5	CO5	CO5	CO5		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 Marks	18 Marks	
1	1KS18CS063	NITISH KUMAR.M.R	5	2							1		8	7	1	
2	1KS18CS064	NOOR SUMAIYA											0	0	0	AB
3	1KS18CS065	P SAI RAM	5	5	5						6	6	27	10	17	
4	1KS18CS066	PAVAN P	6	5	2						6	6	25	11	14	
5	1KS18CS067	POOJASHREE K	6	6	6						6	6	30	12	18	
6	1KS18CS069	PRANAV M S	6								6	3	15	6	9	
7	1KS18CS070	PRATEEK HAVALE											0	0	0	AB
8	1KS18CS071	PRAVEEN KUMAR K	6	6	6				6	6			30	12	18	
9	1KS18CS072	PREETHI K				6					6	0	12	6	6	
10	1KS18CS073	PUJARI VISHNU PRIYA	6	6	6				6	6			30	12	18	
11	1KS18CS074	PULLUR PAVAN KUMAR	5	5	6				6	6	6		28	10	18	
12	1KS18CS075	R DEKSHITHA	6	6	6				6	6	3		30	12	18	
13	1KS18CS076	R PRATIKSHA	6	6	6				6	6			30	12	18	
14	1KS18CS077	RAMYA R	5	6	6				6	6			29	11	18	
15	1KS18CS078	RAHUL.P			6	6					6	4	16	6	10	
16	1KS18CS079	RAIPALLE SHREYAA	6	5							3	6	20	11	9	
17	1KS18CS080	RAKSHA S	5	6							4	1	16	11	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		
			CO4	CO4	CO5	CO4	CO4	CO5	CO5	CO5	CO5	CO5		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 Marks	18 Marks	
18	1KS18CS081	RAKSHITH KUMAR.N	3	6	6				6	1	5	1	22	9	13	
19	1KS18CS082	REKHA N C	5	6	6				6	6			29	11	18	
20	1KS18CS083	RITHANA.N.RAJ	6	3	1						3	1	14	9	5	
21	1KS18CS084	RUBA ABDUL RAHMAN			6						1	6	13	0	13	
22	1KS18CS086	SAMHITHA	6	2							5	6	19	8	11	
23	1KS18CS087	SANDEEP KUMAR											0	0	0	AB
24	1KS18CS088	SAURAV KUMAR											0	0	0	AB
25	1KS18CS089	SAURAV S MAKAM	6	4							6	4	20	10	10	
26	1KS18CS090	SHALINI S	6	6	6				6	6	6	6	30	12	18	
27	1KS18CS091	SHASHANK G	6	6	6						6	6	30	12	18	
28	1KS18CS092	SHASHANK MISHRA	6	6	4						2		18	12	6	
29	1KS18CS093	SHIVANGI SRIVASTAVA											0	0	0	AB
30	1KS18CS094	SHIVA PRAKASH T	6		6				6		6	6	24	6	18	
31	1KS18CS095	SHUBHASHINI.R	6	6	6						6	6	30	12	18	
32	1KS18CS096	SINDU A S											0	0	0	AB
33	1KS18CS097	SOURABH SANTOSH KAMBLE	6	6	6						6	5	29	12	17	
34	1KS18CS098	SRI CHANDANA P	6	6							6	6	24	12	12	
35	1KS18CS099	SRIVIDYA H R	6	6	6				1	6			25	12	13	
36	1KS18CS100	SUBRAMANYA N				6			6	6			18	6	12	
37	1KS18CS101	SUDHAKAR YASWANTH	6	6	6				6	4			28	12	16	
38	1KS18CS102	SUDHANSHU JOSHI	6	6	6						6	6	30	12	18	
39	1KS18CS103	SUJAY G S	6	6	6				6	6			30	12	18	
40	1KS18CS104	SUNAINA NAYAK	6	6	6				6	6			30	12	18	
41	1KS18CS105	SURAJ C JAWOOR	3								6		9	3	6	
42	1KS18CS106	SUSHMITHA S				6				6	4	6	16	6	10	
43	1KS18CS107	SWETHA BIJANAPALLI	6	6	6				5	0			23	12	11	
44	1KS18CS108	THAMMINENI HEMANTH CHOWDARY	6	6	6						4	6	28	12	16	
45	1KS18CS109	THIRUMALAI SHAKTIVEL C	6	6	6				6	6			30	12	18	

Sl. 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		
			CO4	CO4	CO5	CO4	CO4	CO5	CO5	CO5	CO5	CO5		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 Marks	18 Marks	
46	1KS18CS110	VAISHAK P	6	6	6						6	6	30	12	18	
47	1KS18CS111	VARIDHI MADHURANATH	6	6	6						6	6	30	12	18	
48	1KS18CS112	VEDAVEDYA B H	6	1						2	2		9	7	2	
49	1KS18CS113	VEERA SREENIDHI.R											0	0	0	AB
50	1KS18CS114	VENKATESH M N	6			6				3			9	6	3	
51	1KS18CS115	VIJAY.N.S	6	5	6						2		19	11	8	
52	1KS18CS116	VIJAYASHREE.N.R	6	6	6				6	6	6	6	30	12	18	
53	1KS18CS117	VIJETHA				6			6	6			18	6	12	
54	1KS18CS118	VIINOD H MALALI	6	6	6				6	6			30	12	18	
55	1KS18CS119	VISHNUPRIYA D	6		6				5	6			23	6	17	
56	1KS18CS120	VYJAYANTHI K S	6	6	6				6	6			30	12	18	
57	1KS18CS121	YASHWANTH.K	6	6	4				6	6			28	12	16	
58	1KS18CS122	YOGITA RAIKAR	6	6	6				6	5			29	12	17	
59	1KS18CS123	ZAINA KHAN	6		6				5	6			23	6	17	
60	1KS18CS124	SHEWANI CHIB	5	6	6						6	6	29	11	18	
61	1KS17CS015	B R GAGAN	6	6	6						6	6	30	12	18	
62	1KS18CS125	R SOUMYA	6	6	6				6	6			30	12	18	
63	1KS18CS126	RAYYAAN MOHIADDIN											0	0	0	AB
64	1KS18CS127	ARVIND PATHAK	5	6	6						4	6	27	11	16	
65	1KS18CS128	BHAGYASHREE.V	5	4	4						6	5	24	9	15	
66	1KS18CS129	BI BI AYESHA	6	6	6				6	6			30	12	18	
67	1KS18CS130	LIKITHA.S	6	6	5				6	6			29	12	17	
68	1KS18CS131	SHALINI.K.P	6	6	6				6	6			30	12	18	
69	1KS19CS401	ARPITHA.G	4								6	6	16	4	12	
70	1KS19CS404	BHAVYASHREE.R											0	0	0	AB
71	1KS19CS413	SAHANA.V	6		6						6	6	24	6	18	
72	1KS19CS414	Y.MRUDULA JAIN			6						6	6	18	0	18	

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 and Engineering

YEAR / SEMESTER	III /V B
COURSE TITLE	UNIX PROGRAMMING
COURSE CODE	18CS56
ACADEMIC YEAR	2020-2021
INTERNALS	IMPROVEMENT

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		
			CO4	CO4	CO5	CO4	CO4	CO5	CO5	CO5	CO5	CO5		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 Marks	18 Marks	
1	1KS18CS064	NOOR SUMAIYA				2	5	6	6			5	19	7	12	
2	1KS18CS070	PRATEEK HAVALE				6	6	5	6	6			29	12	17	
3	1KS18CS087	SANDEEP KUMAR				5	6	6			6	3	26	11	15	
4	1KS18CS088	SAURAV KUMAR											0	0	0	AB
5	1KS18CS093	SHIVANGI SRIVASTAVA				6	6	4			5	6	27	12	15	
6	1KS18CS096	SINDU A S				6	5	5	6	5			27	11	16	
7	1KS18CS113	VEERA SREENIDHI.R					6	5			6	6	23	6	17	
8	1KS18CS126	RAYYAAN MOHIADDIN											0	0	0	AB
9	1KS19CS401	ARPITHA.G					2		6	6			14	2	12	
10	1KS19CS404	BHAVYASHREE.R	5				4		6	4			15	5	10	

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 : 5

Sl NO.	USN	18CIV59	18CS51	18CS52	18CS53	18CS54	18CS55	18CS56	18CSL57	18CSL58	STUDENT SIGNATURE
1	1KS17CS015	32	29	25	40	40	39	39	25	36	
2	1KS18CS001	32	25	19	33	37	37	34	30	33	
3	1KS18CS002	35	25	20	40	37	33	31	34	39	
4	1KS18CS003	34	24	19	32	34	37	36	30	33	
5	1KS18CS005	30	25	24	33	35	31	35	36	35	
6	1KS18CS006	39	38	37	40	39	40	37	40	40	
7	1KS18CS007	35	29	40	40	39	40	36	36	37	
8	1KS18CS008	32	29	33	40	36	40	38	36	40	
9	1KS18CS009	33	24	27	36	37	37	36	31	38	
10	1KS18CS010	39	22	23	38	31	37	33	30	30	
11	1KS18CS012	27	30	22	38	39	39	36	34	35	
12	1KS18CS013	34	26	29	40	38	36	38	30	36	
13	1KS18CS014	32	21	19	26	33	35	32	30	30	
14	1KS18CS015	40	31	22	40	40	39	39	31	36	
15	1KS18CS016	35	26	23	38	39	36	32	40	37	
16	1KS18CS017	40	31	24	38	40	40	37	31	37	
17	1KS18CS018	32	27	31	40	40	40	37	30	39	
18	1KS18CS019	40	26	31	36	37	39	37	34	40	
19	1KS18CS020	36	24	33	38	40	38	36	31	39	
20	1KS18CS022	33	24	24	38	38	39	37	31	34	
21	1KS18CS023	34	20	20	39	40	39	34	31	38	
22	1KS18CS024	33	34	27	36	39	37	35	31	39	
23	1KS18CS025	27	20	20	37	26	33	24	31	37	
24	1KS18CS026	27	21	20	29	33	33	26	28	35	
25	1KS18CS027	34	31	27	38	35	38	35	27	40	
26	1KS18CS028	36	27	27	33	33	35	32	31	38	
27	1KS18CS029	31	26	26	37	38	38	37	33	36	
28	1KS18CS030	39	25	17	40	38	38	31	31	37	
29	1KS18CS031	37	25	20	37	38	39	36	31	36	
30	1KS18CS032	39	30	27	38	40	40	38	34	37	
31	1KS18CS033	31	28	24	38	37	37	34	31	39	
32	1KS18CS034	33	27	24	38	40	40	35	36	40	
33	1KS18CS035	35	29	36	36	38	37	33	31	37	
34	1KS18CS036	33	20	25	38	33	34	31	31	34	
35	1KS18CS037	34	32	32	38	40	39	38	31	39	
36	1KS18CS038	38	32	22	39	40	40	36	39	40	
37	1KS18CS039	33	26	19	37	37	34	31	31	37	
38	1KS18CS040	36	22	25	35	35	38	36	31	35	
39	1KS18CS041	40	25	19	38	38	37	34	31	37	

O.	USN	18CIV59	18CS51	18CS52	18CS53	18CS54	18CS55	18CS56	18CSL57	18CSL58	STUDENT SIGNATURE
41	1KS18CS043	27	18	19	35	30	35	28	25	33	
42	1KS18CS044	33	21	18	35	39	35	31	30	37	
43	1KS18CS045	38	27	33	37	39	38	33	34	39	
44	1KS18CS046	33	24	26	33	30	31	32	34	32	
45	1KS18CS047	34	28	30	35	37	40	32	33	38	
46	1KS18CS049	39	27	17	33	39	39	36	34	34	
47	1KS18CS050	39	25	19	32	38	37	31	27	35	
48	1KS18CS051	39	37	38	40	40	40	39	40	40	
49	1KS18CS052	40	30	23	36	34	40	35	33	39	
50	1KS18CS053	31	31	24	36	39	38	39	33	38	
51	1KS18CS054	37	27	17	38	32	33	33	30	35	
52	1KS18CS055	27	22	16	32	30	30	27	25	35	
53	1KS18CS056	32	25	29	40	39	38	34	36	40	
54	1KS18CS057	40	27	32	40	39	40	39	34	40	
55	1KS18CS058	33	21	17	29	25	28	24	31	35	
56	1KS18CS059	35	20	17	32	35	33	30	25	36	
57	1KS18CS060	37	26	20	33	36	37	30	25	37	
58	1KS18CS061	38	24	26	39	36	37	36	31	40	
59	1KS18CS062	33	34	40	40	40	39	40	37	40	
60	1KS18CS063	34	20	22	38	30	31	29	31	40	
61	1KS18CS064	37	22	27	37	39	37	36	23	35	
62	1KS18CS065	32	30	23	39	35	39	37	30	32	
63	1KS18CS066	33	25	23	40	39	38	38	23	33	Pavan. P
64	1KS18CS067	31	29	36	40	39	38	40	39	39	
65	1KS18CS069	31	21	22	23	29	30	32	26	40	
66	1KS18CS070	40	21	23	37	36	37	39	30	35	
67	1KS18CS071	33	29	32	38	39	39	40	24	38	
68	1KS18CS072	34	28	22	38	34	39	31	24	35	Preethik
69	1KS18CS073	39	30	36	39	39	39	40	33	39	
70	1KS18CS074	33	29	36	36	38	37	39	36	39	
71	1KS18CS075	39	30	39	40	38	37	40	24	40	
72	1KS18CS076	40	33	40	40	40	40	40	40	40	
73	1KS18CS077	39	26	23	38	38	39	39	20	32	Ramya R
74	1KS18CS078	33	22	16	30	30	34	31	21	30	
75	1KS18CS079	31	25	19	35	33	31	35	24	30	
76	1KS18CS080	27	24	20	36	32	34	34	28	37	
77	1KS18CS081	32	27	30	40	38	36	38	24	37	
78	1KS18CS082	33	30	29	40	40	40	40	35	35	
79	1KS18CS083	38	27	21	34	33	32	31	21	35	
80	1KS18CS084	33	24	18	38	35	34	32	24	35	
81	1KS18CS086	33	20	20	36	33	33	37	22	35	
82	1KS18CS087	33	22	18	39	34	36	37	22	30	
83	1KS18CS088	27	17	16	30	27	37	27	19	21	
84	1KS18CS089	32	20	21	37	35	36	33	21	30	
85	1KS18CS090	39	31	38	39	40	40	40	34	40	Shalini. S
86	1KS18CS091	34	28	30	39	38	39	40	39	40	
87	1KS18CS092	31	21	21	35	35	31	35	23	40	Shashank

	USN	18CIV59	18CS51	18CS52	18CS53	18CS54	18CS55	18CS56	18CSL57	18CSL58	STUDENT SIGNATURE
89	1KS18CS094	34	21	25	35	35	39	36	28	39	
90	1KS18CS095	37	25	27	36	38	37	38	24	39	
91	1KS18CS096	39	34	39	40	38	39	39	33	39	
92	1KS18CS097	39	33	30	40	36	40	37	32	40	
93	1KS18CS098	32	25	29	38	37	35	38	30	35	
94	1KS18CS099	34	25	32	37	35	36	37	31	36	
95	1KS18CS100	34	21	18	34	31	35	33	22	36	
96	1KS18CS101	32	20	21	36	38	37	39	24	35	
97	1KS18CS102	35	31	32	36	34	38	40	31	36	
98	1KS18CS103	39	34	40	40	40	40	40	40	40	
99	1KS18CS104	37	29	28	38	40	39	40	32	40	
100	1KS18CS105	32	23	16	33	31	33	30	23	30	
101	1KS18CS106	33	25	21	39	38	38	35	30	37	
102	1KS18CS107	35	21	25	37	38	32	36	24	30	
103	1KS18CS108	34	21	23	34	37	38	39	30	32	
104	1KS18CS109	34	30	32	36	38	40	40	37	40	
105	1KS18CS110	34	23	25	33	37	32	39	24	32	
106	1KS18CS111	34	31	29	38	38	38	40	36	40	
107	1KS18CS112	33	23	19	30	36	33	33	23	35	
108	1KS18CS113	40	26	20	36	33	32	35	21	32	
109	1KS18CS114	33	23	16	33	30	33	31	22	35	
110	1KS18CS115	30	25	20	36	38	35	33	23	36	
111	1KS18CS116	40	30	19	38	34	40	39	28	33	
112	1KS18CS117	34	27	22	38	38	38	36	24	40	
113	1KS18CS118	34	20	28	38	39	38	40	23	40	
114	1KS18CS119	38	28	26	38	40	37	37	25	40	
115	1KS18CS120	35	27	25	38	37	40	38	32	35	
116	1KS18CS121	31	21	33	38	37	37	38	32	39	
117	1KS18CS122	40	31	40	40	40	39	40	29	40	
118	1KS18CS123	38	29	26	40	40	39	38	27	40	
119	1KS18CS124	31	27	25	38	38	37	39	22	38	
120	1KS18CS125	37	31	28	40	40	40	40	39	36	
121	1KS18CS126	27	16	4	19	20	27	21	22	26	
122	1KS18CS127	36	33	20	40	36	39	38	22	30	
123	1KS18CS128	31	31	19	40	38	36	37	26	35	
124	1KS18CS129	34	28	23	38	40	40	38	23	30	
125	1KS18CS130	32	29	20	40	40	39	37	24	30	
126	1KS18CS131	33	26	16	40	38	37	39	24	30	
127	1KS19CS400	33	22	19	36	36	39	34	31	36	
128	1KS19CS401	38	20	16	36	30	32	33	26	30	
129	1KS19CS402	40	23	16	32	38	38	31	31	35	
130	1KS19CS403	39	27	16	31	36	36	32	31	34	
131	1KS19CS404	39	22	17	37	32	33	32	22	32	
132	1KS19CS405	39	34	18	40	36	37	31	36	39	
133	1KS19CS406	33	23	16	36	31	37	30	31	35	
134	1KS19CS407	27	20	16	30	28	31	23	31	33	
135	1KS19CS408	36	26	18	31	32	33	33	28	36	

SI NO.	USN	18CIV59	18CS51	18CS52	18CS53	18CS54	18CS55	18CS56	18CSL57	18CSL58	STUDENT SIGNATURE
137	1KS19CS410	39	20	19	37	26	28	31	28	35	
138	1KS19CS411	38	27	19	40	38	38	31	30	38	
139	1KS19CS412	39	21	17	32	37	32	32	28	36	
140	1KS19CS413	32	24	16	40	35	40	36	39	40	
141	1KS19CS414	33	25	16	38	33	40	34	24	38	
--x--	Faculty Signature										----xxxxxxx----

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

Nirarapur 26/2/2021

HOD

Seal and Signature

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

26/2/2021

PRINCIPAL

Seal and Signature

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

K S INSTITUTE OF TECHNOLOGY, BANGALORE-109.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBJECT NAME & CODE: UNIX PROGRAMMING - 18CS56

V SEM -COMPUTER SCIENCE AND ENGINEERING

IA - INTERNAL ASSESMENT

EX - VTU END SEM EXAMINATION

Sl.No	USN	Name of the Student	18CS56		
			IA	EX	TOT
		CREDIT POINT	3		
1	1KS18CS063	NITISH KUMAR.M.R	29	29	58
2	1KS18CS064	NOOR SUMAIYA	36	29	65
3	1KS18CS065	P SAI RAM	37	42	79
4	1KS18CS066	PAVAN P	38	35	73
5	1KS18CS067	POOJASHREE K	40	42	82
6	1KS18CS069	PRANAV M S	32	25	57
7	1KS18CS070	PRATEEK HAVALE	39	39	78
8	1KS18CS071	PRAVEEN KUMAR K	40	42	82
9	1KS18CS072	PREETHI K	31	31	62
10	1KS18CS073	PUJARI VISHNU PRIYA	40	35	75
11	1KS18CS074	PULLUR PAVAN KUMAR	39	34	73
12	1KS18CS075	R DEKSHITHA	40	29	69
13	1KS18CS076	R PRATIKSHA	40	46	86
14	1KS18CS077	RAMYA R	39	23	62
15	1KS18CS078	RAHUL.P	31	22	53
16	1KS18CS079	RAIPALLE SHREYAA	35	34	69
17	1KS18CS080	RAKSHA S	34	37	71
18	1KS18CS081	RAKSHITH KUMAR.N	38	41	79
19	1KS18CS082	REKHA N C	40	42	82
20	1KS18CS083	RITHANA.N.RAJ	31	21	52
21	1KS18CS084	RUBA ABDUL RAHMAN	32	23	55
22	1KS18CS086	SAMHITHA	37	21	58
23	1KS18CS087	SANDEEP KUMAR	37	26	63
24	1KS18CS088	SAURAV KUMAR	27	AB	27
25	1KS18CS089	SAURAV S MAKAM	33	21	54
26	1KS18CS090	SHALINI S	40	44	84

27	1KS18CS091	SHASHANK G	40	36	**76**
28	1KS18CS092	SHASHANK MISHRA	35	15	**50**
29	1KS18CS093	SHIVANGI SRIVASTAVA	37	41	**78**
30	1KS18CS094	SHIVA PRAKASH T	36	32	**68**
31	1KS18CS095	SHUBHASHINI.R	38	38	**76**
32	1KS18CS096	SINDU A S	39	40	**79**
33	1KS18CS097	SOURABH SANTOSH KAMBLE	37	24	**61**
34	1KS18CS098	SRI CHANDANA P	38	35	**73**
35	1KS18CS099	SRIVIDYA H R	37	29	**66**
36	1KS18CS100	SUBRAMANYA N	33	32	**65**
37	1KS18CS101	SUDHAKAR YASWANTH	39	35	**74**
38	1KS18CS102	SUDHANSHU JOSHI	40	46	**86**
39	1KS18CS103	SUJAY G S	40	35	**75**
40	1KS18CS104	SUNAINA NAYAK	40	52	**92**
41	1KS18CS105	SURAJ C JAWOOR	30	31	**61**
42	1KS18CS106	SUSHMITHA S	35	33	**68**
43	1KS18CS107	SWETHA BIJANAPALLI	36	35	**71**
44	1KS18CS108	THAMMINENI HEMANTH CHOWDARY	39	43	**82**
45	1KS18CS109	THIRUMALAI SHAKTIVEL C	40	42	**82**
46	1KS18CS110	VAISHAK P	39	32	**71**
47	1KS18CS111	VARIDHI MADHURANATH	40	36	**76**
48	1KS18CS112	VEDAVEDYA B H	33	35	**68**
49	1KS18CS113	VEERA SREENIDHI.R	35	23	**58**
50	1KS18CS114	VENKATESH M N	31	24	**55**
51	1KS18CS115	VIJAY.N.S	33	27	**60**
52	1KS18CS116	VIJAYASHREE.N.R	39	28	**67**
53	1KS18CS117	VIJETHA	36	37	**73**
54	1KS18CS118	VIINOD H MALALI	40	42	**82**
55	1KS18CS119	VISHNUPRIYA D	37	48	**85**
56	1KS18CS120	VYJAYANTHI K S	38	23	**61**
57	1KS18CS121	YASHWANTH.K	38	25	**63**
58	1KS18CS122	YOGITA RAIKAR	40	39	**79**
59	1KS18CS123	ZAINA KHAN	38	44	**82**
60	1KS18CS124	SHEWANI CHIB	39	30	**69**
61	1KS18CS125	R SOUMYA	40	51	**91**

62	1KS18CS126	RAYYAAN MOHIADDIN	21	AB	**21**
63	1KS18CS127	ARVIND PATHAK	38	39	**77**
64	1KS18CS128	BHAGYASHREE.V	37	29	**66**
65	1KS18CS129	BI BI AYESHA	38	32	**70**
66	1KS18CS130	LIKITHA.S	37	39	**76**
67	1KS18CS131	SHALINI.K.P	39	21	**60**
68	1KS19CS401	ARPITHA.G	33	21	**54**
69	1KS19CS404	BHAVYASHREE.R	32	21	**53**
70	1KS19CS413	SAHANA.V	36	30	**66**
71	1KS19CS414	Y.MRUDULA JAIN	34	36	**70**
72	1KS17CS015	B R GAGAN (2018 SCHEME)	39	36	**75**
		NUMBER OF PASS		**18**	**72**
		AVERAGE MARKS	**36**	**17**	**69**
		FAILURES IN SUBJECTS		**01**	
		ABSENTS IN SUBJECTS		**02**	

Head of the Department
Dept. of Computer Science & Engg.
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	**Dr Rekha B Venkatapur**
Course Name /Code	**Unix Programming / 18CS56**
Semester/Section	**5th B**
Activity Name	**Quiz:**
Topic Covered	**Module 1 and 2:**
Date	**14th September 2020 and 14th October 2020**
No. of Participants	**58, 13 (Total 71/72)**
Objectives/Goals	**To understand the topics precisely**
ICT Used	**Google form & Zoom Meeting app**

Appropriate Method/Instructional materials/Exam Questions

Students are asked answer 10 quiz questions with multiple choice options. The quiz is conducted for 10 minutes. The Google forms are used to randomize the options. In the class room quiz is conducted for the students.

In this discussion, key questions to be examined are

1. How well the students understood the topics taught in the class?
2. Whether they are indulge in experiential learning by practicing the commands?
3. Are they able to answer all questions within the given time?

Relevant PO's:	**PO: 1,2,5 and 9**
Significance of Results/Outcomes	Students able to know the importance of experiential learning and understanding the Unix commands, options usage and their alternate methods in different versions of UNIX.
Reflective Critique	The main goal of this quiz is to know how well students will be able to answer, explore the commands and apply the knowledge effectively.

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

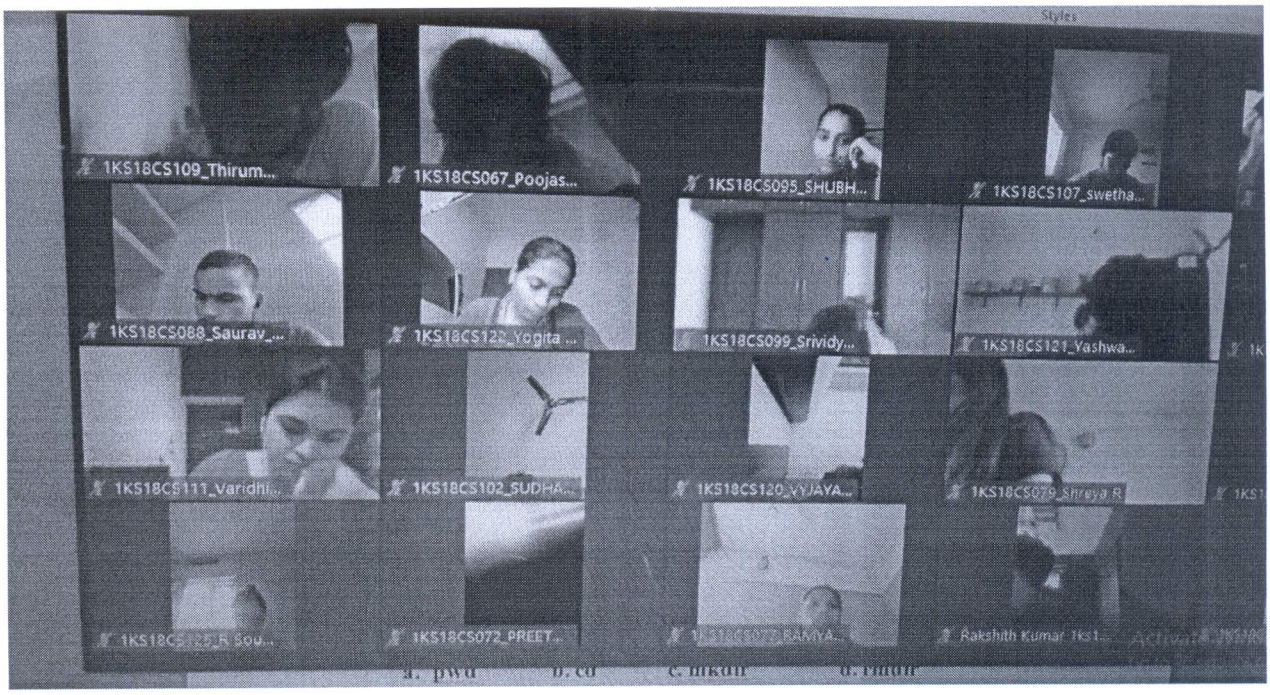


Fig. Photograph of online quiz conducted through google forms.

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

UNIX Programming (18CS56)

Pedagogy Activity - 1

5th Semester B Section

Quiz Questions Date : 14-9-2020 10 Marks

1. What does the command ls do?
   a. Shows a calendar **b. Display of files and folders, present in the folder where you are** c. Opening a file d. Display of the contents of a file

2. How do you find out what's your shell?
   a. whomami b. pwd **c. echo $SHELL** d. $sh

3. Cal command is used for
   a. Calculator **b. Calendar** c. concatenate d. cancel

4. Choose the output of the following command $type echo
   a. echo **b.echo is a shell builtin** c. "echo" d. None of the these

5. Bash interpret the escape sequence only when echo is used with ----- option
   a. –c b. – n c. –t **d. –e**

6. Command $ls – r
   a. **sorts filesname in reverse order** b. Displays recursive files c. displays removed file name d. displays root directory

7. Which option is required to get date in mm/dd/yy format
   a. **D** b.d c. y d. all of these

8. Find the valid file name in Unix
   a. 5-B b. 5. c. .5-B **d. All of these**

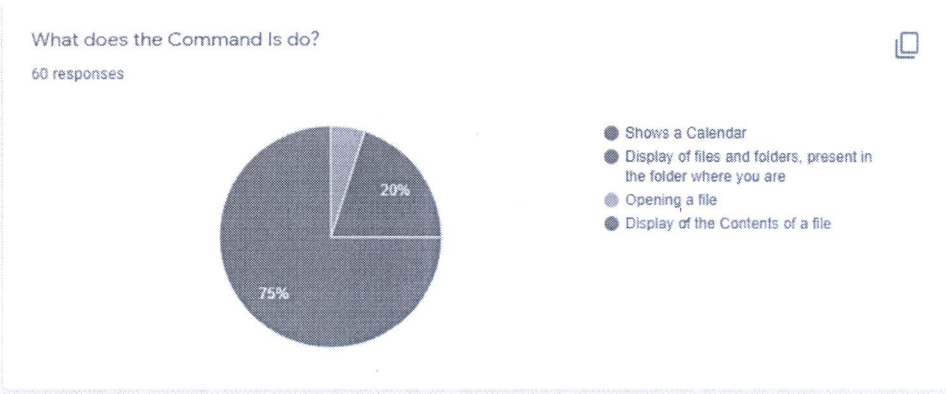
9. _____Interacts with user
   a. Kernel **b. Shell** c. Both Kernel and Shell d. hardware

10. _____ is not a shell
    a. bsh b.ksh c. BASH **d. bat**

Sl.No	USN	Name	Quiz Marks Obtained (Max 10)
1	1KS17CS015	B R GAGAN	9
2	1KS18CS063	Nitish kumar M R	9
3	1KS18CS064	Noor sumaiya	4
4	1KS18CS065	P SAI RAM	6
5	1KS18CS066	PAVAN P	8
6	1KS18CS067	Poojashree.K	7
7	1KS18CS069	Pranav M S	8
8	1KS18CS070	Prateek havale	8
9	1KS18CS071	Praveen Kumar K	8
10	1KS18CS072	PREETHI K	8
11	1KS18CS073	PUJARI VISHNUPRIYA	10
12	1KS18CS074	PULLUR PAVAN KUMAR	8
13	1KS18CS075	R. DEKSHITHA	7
14	1KS18CS076	R Pratiksha	8
15	1KS18CS077	R Ramya	6
16	1KS18CS081	Rakshith Kumar N	9
17	1KS18CS082	Rekha N.C	8
18	1KS18CS086	SAMHITHA	9
19	1KS18CS088	Saurav Kumar	8
20	1KS18CS090	SHALINI.S	9
21	1KS18CS091	SHASHANK G	8
22	1KS18CS092	SHASHANK MISHRA	8
23	1KS18CS093	Shivangi Srivastava	8
24	1KS18CS094	SHIVAPRAKASH T	9
25	1KS18CS095	Shubhashini R	7
26	1KS18CS096	SINDU A S	9
27	1KS18CS097	SOURABH SANTOSH KAMBLE	7
28	1KS18CS098	Sri Chandana P	9
29	1KS18CS099	Srividya H.R.	9
30	1KS18CS100	Subramanya N	9
31	1KS18CS101	Sudhakar Yaswanth	7
32	1KS18CS102	Sudhanshu Joshi	7
33	1KS18CS103	Sujay GS	8
34	1KS18CS104	SUNAINA NAYAK	10
35	1KS18CS106	Sushmitha S	7

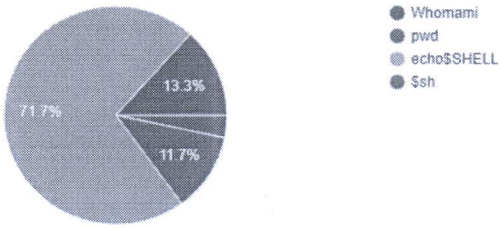
36	1KS18CS107	Swetha bijanapalli	6
37	1KS18CS108	T HEMANTH CHOWDARY	7
38	1KS18CS109	Thirumalai Shaktivel	9
39	1KS18CS110	Vaishak P	9
40	1KS18CS111	Varidhi	8
41	1KS18CS112	Vedavedya B H	6
42	1KS18CS113	VEERA SREENIDHI R	4
43	1KS18CS115	Vijay	7
44	1KS18CS116	Vijayashree.N.R	6
45	1KS18CS117	Vijetha	6
46	1KS18CS118	VINOD H MALALI	8
47	1KS18CS119	Vishnupriya D	7
48	1KS18CS120	Vyjayanthi K S	8
49	1KS18CS121	YASHWANTH K	8
50	1KS18CS122	Yogita Raikar	10
51	1KS18CS123	Zaina khan	8
52	1KS18CS124	Shewani Chib	6
53	1KS18CS125	R Soumya	9
54	1KS18CS128	BHAGYASHREE V	3
55	1KS18CS129	Bi Bi Ayesha	6
56	1KS18CS130	Likitha S	7
57	1KS18CS131	SHALINI K P	5
58	1KS19CS414	Mrudula Jain	5

Analysis of Quiz



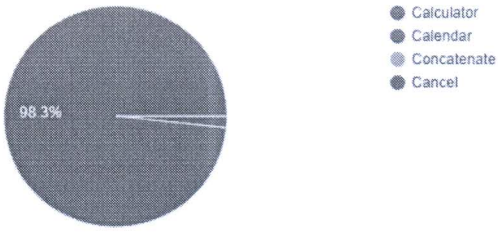
How do you find out what's your shell?

60 responses



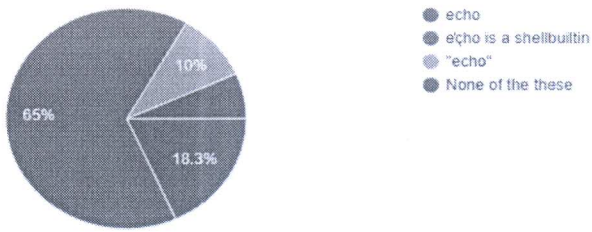
Cal command is used for

60 responses



Choose the output of the following command $type echo

60 responses



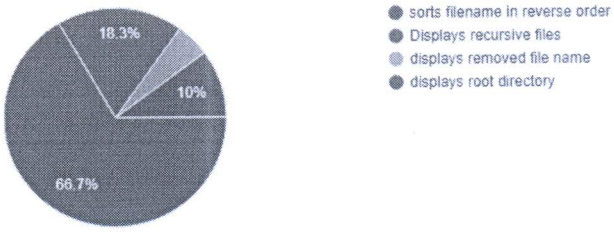
Bash interpret the escape sequence only when echo is used with ____ option

60 responses

Activate Wi
Go to PC settin

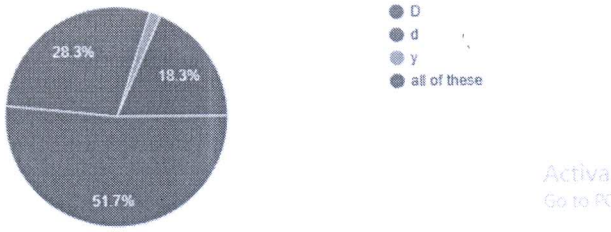
Command $ls -r

60 responses



Which option is required to get date in mm/dd/yy format

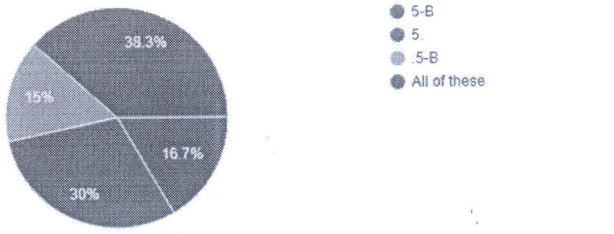
60 responses



Activate Windows
Go to PC settings to act

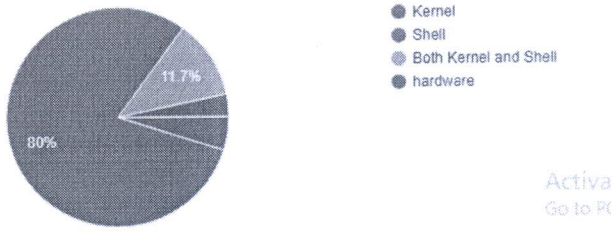
Find the valid file name in Unix

60 responses

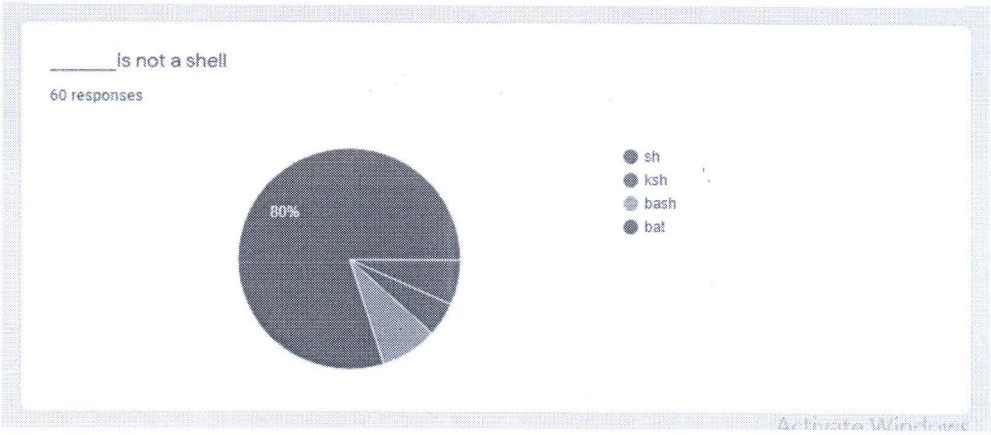


_____Interacts with user

60 responses



Activate Windows
Go to PC settings to act



UNIX Programming (18CS56)

Pedagogy Activity – 1(For Absentees)

5th Semester B Section

Quiz Questions **Date : 8-10-2020** **10 Marks**

UNIX Programming (18CS56)

Pedagogy Activity – 1(For Absentees)

5th Semester B Section

Quiz Questions **Date : 10-9-2020** **10 Marks**

1. The command used to display listing of directories
   a. **ls –d** b. ls -ld c. ls –dir d. ls – dl

2. Th operation used to change permission in relative manner is
   a. **+ or – or =** b. rwx c. user, group owner and others d. chmod

3. The ls -l file1 command results in
   a. Long listing of file 1 b. 7 attributes of file 1 **c. both** d. None

4. Octal 100 indicates ----- in relative permission
   a. **Read** b. write c. execute d. None of the above

5. What is the output of the commands on ordinary file startx
   $chmod a -rwx stratx; ls – l startx
   a. –rwxrwxrwx b. d-------- **c. ----------** d.000000000

6. What is the result of this command on ordinary file named startx
   $chmod 777 startx
   a. **–rwxrwxrwx** b. --------- c. drwxr-xr-x d. None of the above

7. The first column of the command $ls – l startx indicate
   a. Owner of file **b. ordinary file** c. type and permission of file d. directory file

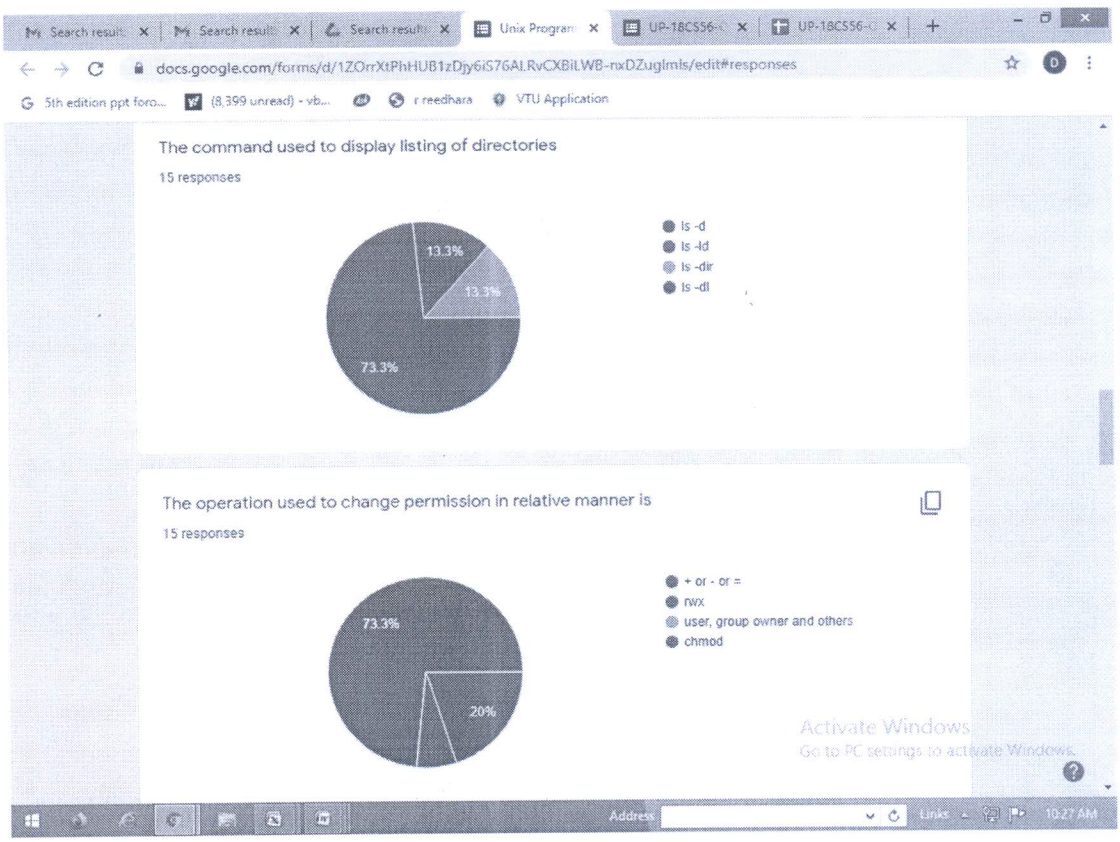
8. Which of the following is used to assign all permissions to files f1 and f2
   a. ls –l file1,file2 b. chmod 777 f1,f2 **c. chmod 777 f1; chmod 777 f2**
   b. None of the above

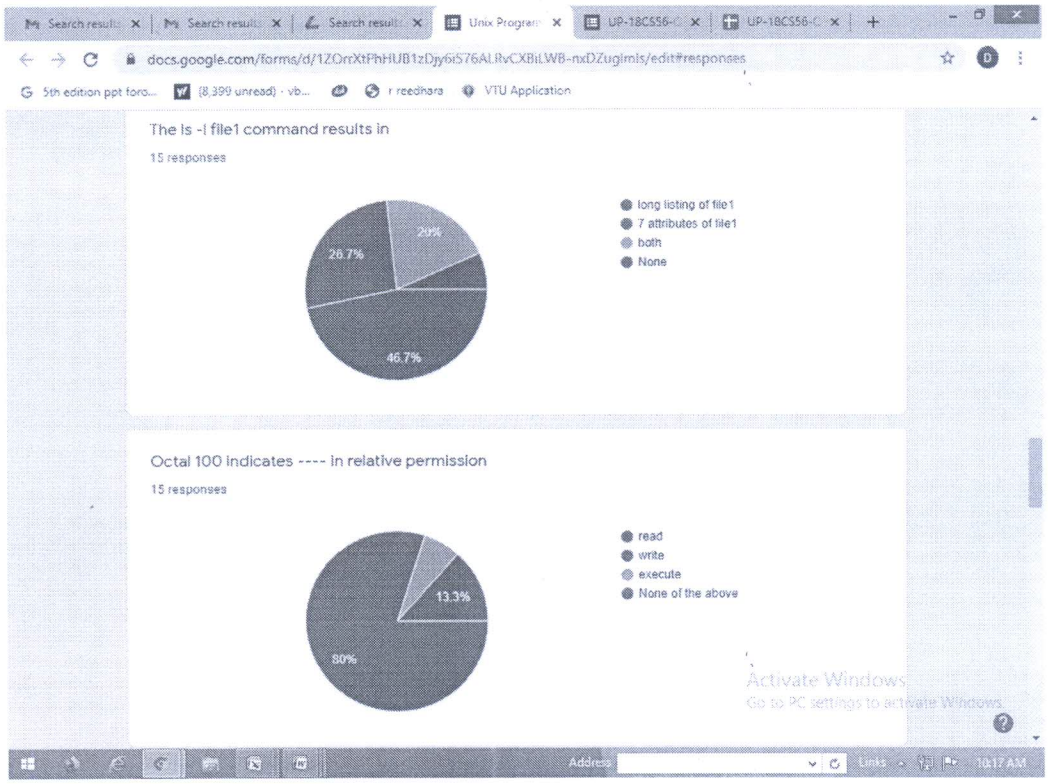
9. umask command is used
   a. to check existing permissions b. to display UID,GID **c. to check and change existing permissions** d. for all the above

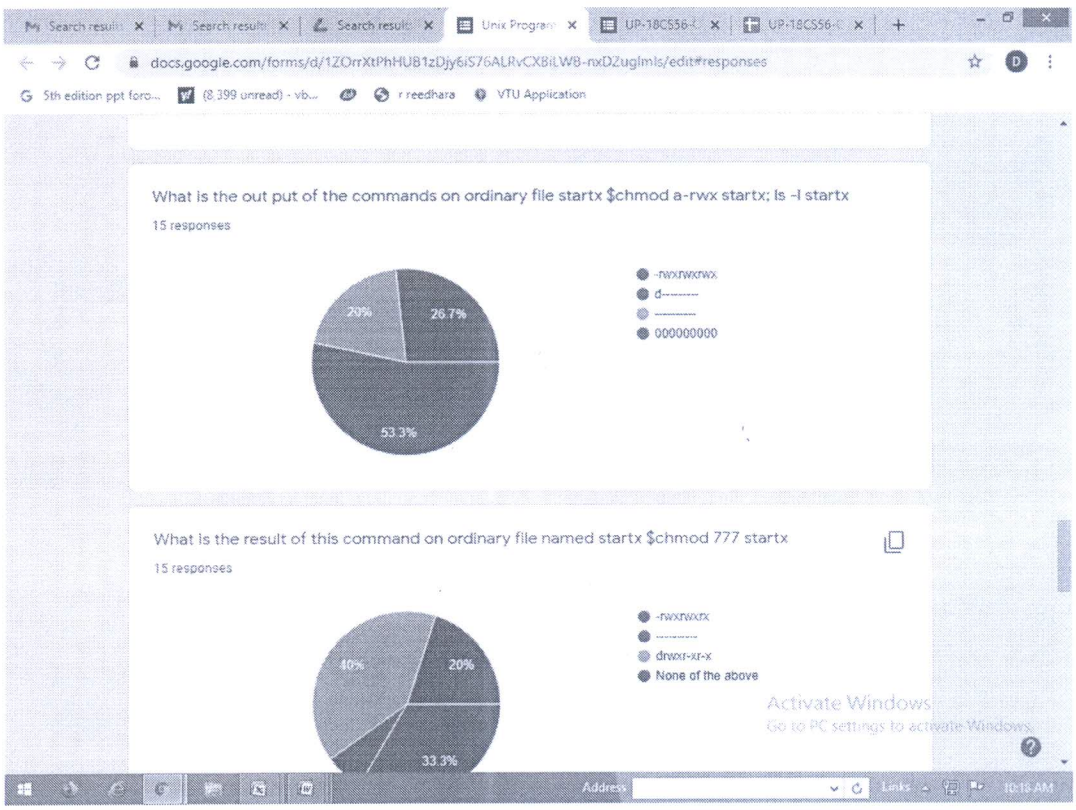
10. The date and time attribute of a file indicates
    a. Last modification date and Time b. Last accessed date and Time c. Date time of creation of file d. Date and time of file deleted

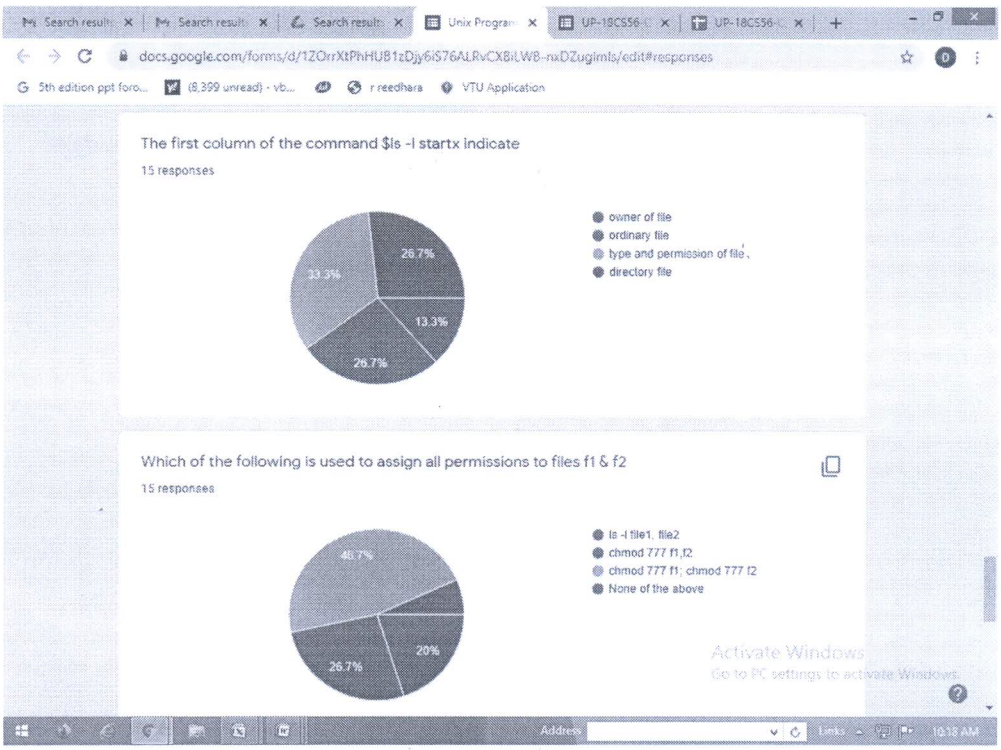
Sl.No	USN	Name	Quiz Marks Obtained (Max 10)
1	1KS18CS079	Shreya Raipalle	1
2	1KS18CS080	Raksha S	2
3	1KS18CS083	Rithana N Raj	1
4	1KS18CS084	Ruba Abdul Rahaman	1
5	1KS18CS087	Sandeep Kumar	3
6	1KS18CS105	Suraj C Jawoor	2
7	1KS18CS114	Venkatesh M.N	1
8	1KS18CS126	Rayyaan Mohiaddin	3
9	1KS18CS127	Aravind Pathak	3
10	1KS19CS401	Arpitha.G	1
11	1KS19CS404	Bhavyashree. R	2
12	1KS19CS413	Sahana. V	2

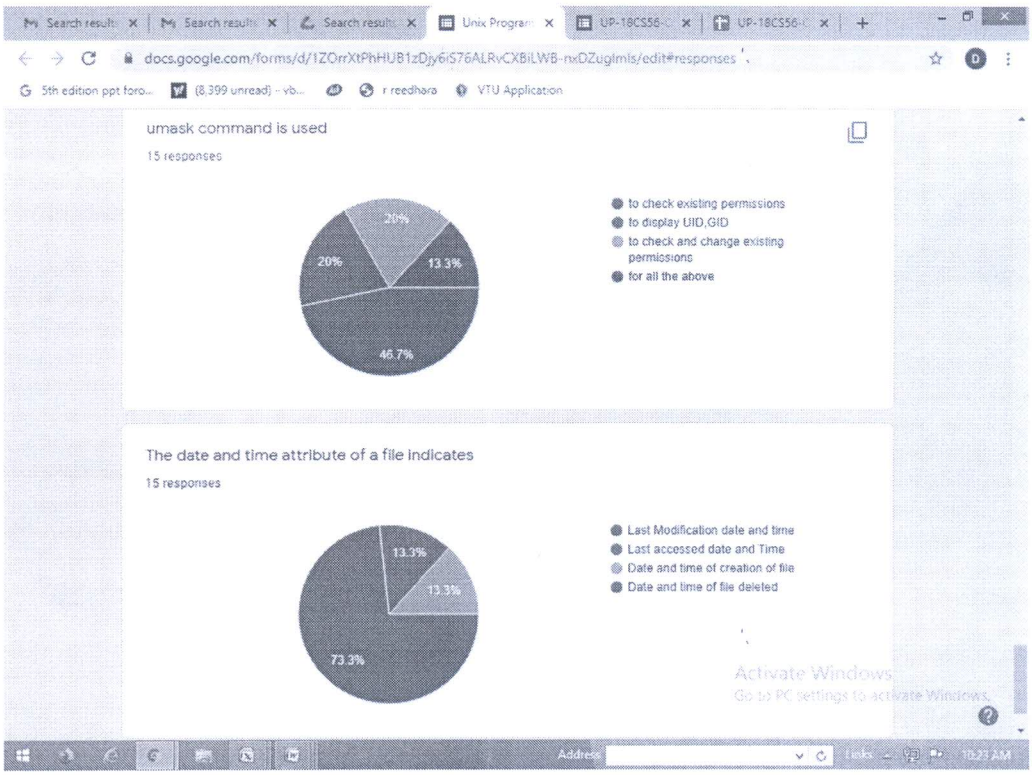
Analysis of Quiz




docs.google.com/forms/d/1ZOrrXtPhHUB1zDjy6iS76ALRvCXBiLWB-nxDZuglmls/edit#responses
The ls -l file1 command results in
15 responses
20%
26.7%
46.7%
long listing of file1
7 attributes of file1
both
None
Octal 100 indicates ---- in relative permission
15 responses
13.3%
80%
read
write
execute
None of the above
Activate Windows
Go to PC settings to activate Windows.


docs.google.com/forms/d/1ZOrrXtPhHUB1zDjy6iS76ALRvCXBiLWB-nxDZuglmls/edit#responses
What is the out put of the commands on ordinary file startx $chmod a-rwx startx; ls -l startx
15 responses
20%
26.7%
53.3%
-rwxrwxrwx
d---------
000000000
What is the result of this command on ordinary file named startx $chmod 777 startx
15 responses
40%
20%
33.3%
-rwxrwxrx
drwxr-xr-x
None of the above
Activate Windows
Go to PC settings to activate Windows.


docs.google.com/forms/d/1ZOrrXtPhHUB1zDjy6iS76ALRvCXBiLWB-nxDZuglmls/edit#responses
The first column of the command $ls -l startx indicate
15 responses
26.7%
33.3%
13.3%
26.7%
owner of file
ordinary file
type and permission of file
directory file
Which of the following is used to assign all permissions to files f1 & f2
15 responses
46.7%
20%
26.7%
ls -l file1, file2
chmod 777 f1,f2
chmod 777 f1; chmod 777 f2
None of the above
Activate Windows
Go to PC settings to activate Windows.


docs.google.com/forms/d/1ZOrrXtPhHUB1zDjy6iS76ALRvCXBiLWB-nxDZuglmls/edit#responses
umask command is used
15 responses
20%
20%
13.3%
46.7%
to check existing permissions
to display UID,GID
to check and change existing permissions
for all the above
The date and time attribute of a file indicates
15 responses
13.3%
13.3%
73.3%
Last Modification date and time
Last accessed date and Time
Date and time of creation of file
Date and time of file deleted
Activate Windows
Go to PC settings to activate Windows.

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	**Dr Rekha B Venkatapur**
Course Name /Code	**Unix Programming / 18CS56**
Semester/Section	**5th B**
Activity Name	**Demonstration of Programs (team Activity)**
Topic Covered	**Module 3 (File APIs)**
Date	**25-11-2020**
No. of Participants	**54 /72**
Objectives/Goals	**Experiential learning**
ICT Used	**Linux platform for demo and Online tool Microsoft team for presentation**

Appropriate Method/Instructional materials/Exam Questions

In the class of 72 students 3 teams are formed as Team B1, B2 and B3.
The team leaders are identified by team mates.

Team B1: Total Members 25
Leader: Sujay G S(1KS18CS103)

Team B2: Total Members 24
Leader: Pujari Vishnupriya (1KS18CS073)

Team B3: Total Members 22
Leader : Zaina Khan (1KS18CS119)

Students are asked to give a demonstration on

1. Write a program to find the type of a file – Whether it is a Regular File/Directory file/ Character Special/Block special/FIFO or pipe/ symbolic file etc.

2. C program to implement a FIFO file named as fifo5 with all access permission to everyone.

3. Write a C/C++ program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.

In this discussion, key questions to be examined are

1. Whether they are indulge in experiential learning by practicing the commands?
2. Are they able to present the work assigned – demonstration and answer the queries related to the Program.
3. Are they able to work effectively as an individual and team?

Relevant PO's:	**PO: 1,2,5 and 9**
Significance of Results/Outcomes	Students able to know the importance of experiential learning and understanding the file attributes and permissions.
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 commands and apply the knowledge effectively.

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

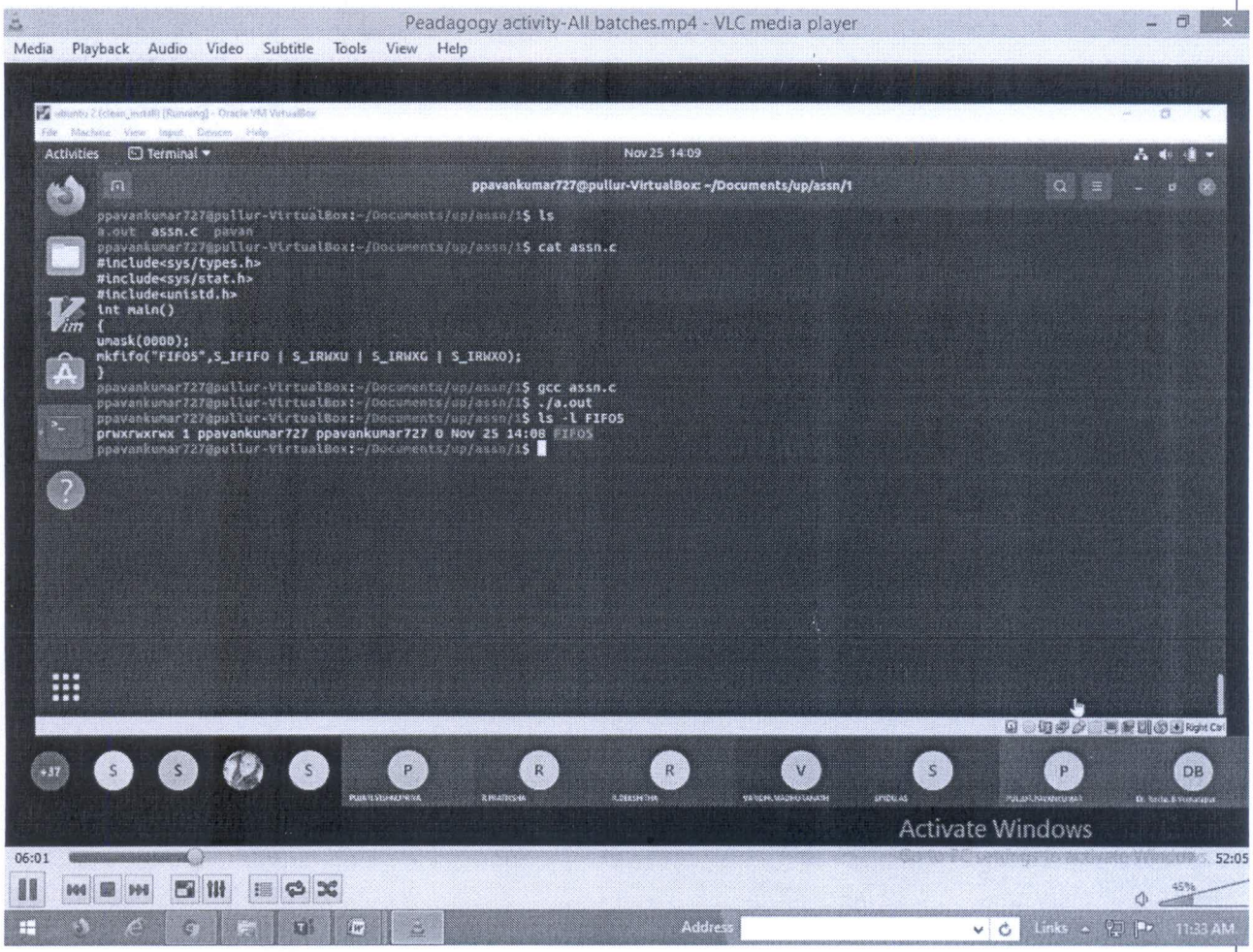


Fig. Photograph during the demonstration of FIFO pgm by Pullar Pavan kumar of Batch B2.

Video Link
https://drive.google.com/file/d/1XPgXtJeS0etEho4yvda9h1HfZeWHo8_3/view?usp=sharing

25/11/2020

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

UNIX Programming (18CS56)

Pedagogy Activity - 2

5th Semester B Section

Program Demonstration – Team Activity

Date : 25-11-2020 **10 Marks**

In the class of 72 students 3 teams are formed as Team B1 , B2 and B3.
The team leaders are identified by team mates.

Team B1: Total Members 25
Leader: Sujay G S

Team B2: Total Members 24
Leader: Pujari Vishnupriya

Team B3: Total Members 22
Leader : Zaina Khan

Programs given for demonstration:

1. Write a program to find the type of a file – Whether it is Regular File/Directory file/ Character Special/Block special/FIFO or pipe/ symbolic file etc.

2. C program to implement a FIFO file named as fifo5 with all access permission to everyone.

3. Write a C/C++ program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.

Demonstration by Batches

Video Link
https://drive.google.com/file/d/1XPgXtJeS0etEho4yvda9h1HfZeWHo8_3/view?usp=sharing

25/11/2020
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

Team B1

NAME	USN
SUJAY GS (L)	1KS18CS103
SOURABH SANTOSH KAMBLE	1KS18CS097
BI BI AYESHA	1KS18CS129
BHAGYASHREE V	1KS18CS128
LIKITHA S	1KS18CS130
SHALINI KP	1KS18CS131
NITISH KUMAR MR	1KS18CS063
SHASHANK G	1KS18CS091
SAURAV KUMAR	1KS18CS088
SUDHANSHU JOSHI	1KS18CS102
SHIVAPRAKASH T	1KS18CS094
BR GAGAN	1KS17CS015
THIRUMALAI SHAKTIVEL C	1KS18CS109
VAISHAK P	1KS18CS110
PRAVEEN KUMAR K	1KS18CS071
RAKSHITH KUMAR N	1KS18CS081
P SAIRAM	1KS18CS065
PAVAN P	1KS18CS066
YASHWANTH K	1KS18CS121
VINOD H MALALI	1KS18CS118
VEDAVEDYA BH	1KS18CS112
THAMMINENI HEMANTH CHOWDARY	1KS18CS108
SUDHAKAR YASHWANTH	1KS18CS101

PRATEEK HAVALE	1KS18CS070
RAYYAAN MOHIADDIN	1KS18CS126

Team B2

NAME	USN
PUJARI VISHNUPRIYA (L)	1KS18CS073
R. PRATIKSHA	1KS18CS076
PULLUR PAVAN KUMAR	1KS18CS074
SHEWANI CHIB	1KS18CS124
YOGITHA RAIKAR	1KS18CS122
ARVIND PATHAK	1KS18CS127
SHIVANGHI SRIVASTAVA	1KS18CS093
VIJETHA	1KS18CS117
VYJAYANTHI K S	1KS18CS120
SHALINI S	1KS18CS090
PREETHI K	1KS18CS072
VARIDHI M	1KS18CS111
SWETHA BIJANAPALLI	1KS18CS107
R DEKSHITHA	1KS18CS075
SUSHMITHA S	1KS18CS106
RAMYA R	1KS18CS077
SINDU A S	1KS18CS096
SHUBHASHINI R	1KS18CS095
SRI CHANDANA P	1KS18CS098
SURAJ C JAWOOR	1KS18CS105
SUBRAMANYA N	1KS18CS100
SAURAV S MAKAM	1KS18CS089
RAHUL P	1KS18CS078
VENKATESH M N	1KS18CS114

Team B3

NAME	USN
VISHNUPRIYA D	1KS18CS119
RITHANA N RAJ	1KS18CS083
RAKSHA S	1KS18CS080
SHASHANK MISHRA	1KS18CS092
REKHA N.C	1KS18CS082
NOOR SUMAIYA	1KS18CS064
RUBA ABDUL RAHMAN	1KS18CS084
SUNAINA NAYAK	1KS18CS104
SANDEEP KUMAR	1KS18CS087
R SOUMYA	1KS18CS125
PRANAV M S	1KS18CS069
BHAVYASHREE.R	1KS19CS404
SAMHITHA	1KS18CS086
VIJAY S	1KS18CS115
MRUDULA JAIN	1KS19CS414
SHREYA R	1KS18CS079
SRIVIDYA H R	1KS18CS099

SAHANA V	1KS19CS413
VIJAYASHREE	1KS18CS116
POOJASHREE K	1KS18CS067
ARPITHA.G	1KS19CS401
ZAINA KHAN (L)	1KS18CS123

K.S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

18CS56–UNIX PROGRAMMING

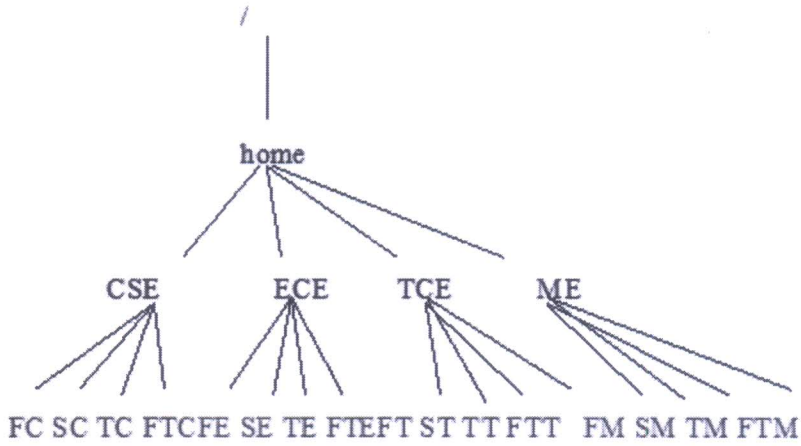
EXHAUSTIVE QUESTION BANK

MODULE-I

1. Explain the architecture of UNIX operating system with a neat diagram(8)
2. Illustrate with a neat diagram the typical UNIX file system and explain different types of files supported in UNIX.(8)
3. **Explain/compare internal and External commands with examples(4)**
4. Explain salient features of UNIX operating system(7)
5. Explain parent child relationship of UNIX file system with a diagram.(6)
6. **Explain with example Absolute pathname and relative pathname.(6)**
7. **Describe command arguments and options with suitable examples** (4)
8. Define a file.With examples explain three categories of files supported by UNIX.(8)
9. **Briefly describe**: (6)
   a. System call b. **PATH** **c.HOME** **d. date** **e. who** **f. ls**
   e. **printf**
10. How an ordinary user can become a superuser and vise versa? Explain with example.
11. **Briefly describe commands with example** (8)
    a. pwd **b. cd** **c. mkdir** **d. rmdir**
12. Name the command used for creating, deleting and changing directory. Explain with suitable syntax and example.

13. Write the command sequence for creating tree structure. Create a file by name country.txt in /home/CSE/FC, write the command to copy file country.txt from /home/CSE/FC to SE, TT and FTM directory.

1.



14. Draw the tree structure of the file system created by the following commands(Assume you are in the directory /usr/office). Why is it not possible to issue the command rmdir /usr/office/right (8)

$mkdir left

$mkdir middle

$mkdir right

$ cd left

$mkdir left middle right

$cd ../middle

$mkdir dir1 dir2 /usr/office/right/dir3

15. Assume you are in /home/Karthik, which of the following commands will work when executed in sequence? Explain the proper reason.

mkdir a/b/c →mkdir a a/b

mkdir a a/b a/b/c →rmdir a/b/c →rmdir a a/b→mkdir a/p a/q a/p/r

Draw the final tree for directory a. (7)

16. Explain the significance of dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names.

17. Write the command line to perform the following

    a. Change current directory to home directory

    b. Change to parent of parent directory. (2)

18. Explain the following commands with suitable example for each.

    a. cat b. rm c. mv d. cp

K.S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

18CS56–UNIX PROGRAMMING

EXHAUSTIVE QUESTION BANK

MODULE-2

1. Briefly explain the significance of the seven fields of 'ls –l' command.
2. Explain absolute and relative methods of assigning permissions to file using examples.(or) Explain different ways of setting file permissions.
3. Current file permission of a regular file "Attendance .txt" are **rw- -w-r-x** write the chmod expression required to change it to following:

   a) rwxrw-r-x b) –xrw-rwx c) rwxrwxrwx

   Using both relative and absolute methods of assigning permissions
4. Explain shell features of while, if and for with syntax
5. What is the purpose of grepcommand ? explain all options
6. What are wildcard characters? Explain each with example.
7. Explain test command for handling string
8. Write a shell script using case to perform all arithmetic operations.
9. Explain for loop with all possible sources for argument list.
10. Explain with example set and shift commands in unix to manipulate positional arguments
11. With examples explain logical operators in shell programming
12. Write Shell programs:

    a) List of files b) processes of user c) Todays date d)Users of the system
13. Differentiate between hard link and soft link
14. Explain hard link and soft link with examples.
15. Write a menu driven shell script to do the following:

    1. List of files b) Date c) Users of the system d) Process of user

    14. Explain here document (<<) with example.

Dr. Rekha B venkatapur, Professor & Head, Dept of CSE, KSIT.

15. Briefly explain Basic Regular Expression(BRE) and Extended Regular Expression(ERE) metacharacters

16. Write regular expression to match the following:

    a. A decimal number which is non-negative and floating point number

    b. A valid C variable.

17. What would be the effect of the following commands:

    (a) grep"^[A - Z]" file1

    (b) egrep "UNIX|Unix|unix" file1

    (c) grep "UNIX$" file1

    (d) grep "UNIX. UNIX" file1

    (e) grep ".*" file1 > file2

18. Refer to the following employee database and

2233 |a.|jaisharma |Director |Production| |12/03/50 | 7000

5678 |Ramesh Babu |D.G.M |Marketing |19/04/43 | 7800

2365 |barunsengupta|Director |Personnel |11/04/47 | 5400

1265 |S.N. Dasgupta |Manager |Sales |12/09/63 | 5600

2467 |anilaggarwal |Manager |Sales |01/10/78 |3000

3245|Sudhir Agarwal |Executive |Personnel |12/6/89 |7500

3245|Sudhir Agrawal|Manage |Personnel|12/6/89|7500

a) Frame the regular expression using grep command to search the details

b) Search the employees in Sales department

c) Search the employees who are Directors.

d) Search the employees having name as agarwal/aggarwal/agrawal ignore case

e) Search the employees who are manger and show the line numbers.

f) Count the number of employee in production department

g) List the employee in sales, marketing and perconnel department

h) List the employee who's employee id starts with 3

i) List the employee who's salary is above 7000

K.S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

18CS56–UNIX PROGRAMMING

EXHAUSTIVE QUESTION BANK

MODULE-3

1. Explain how fcntl API is used for file and record locking
2. Explain file and record locking?
3. Explain directory file and device file APIs?
4. Explain symbolic link API?
5. Explain Directory link API
6. Explain Device File API
7. Explain FIFO File API
8. Write an explanatory note on environment variables
9. Describe the UNIX Kernel support for process. Show the related data structures
10. Bring out the importance of locking files. Explain in brief the types of lock with API.
11. What are the different ways in which a process can terminate? With a neat block schematic,explain how a process is launched and terminates clearly indicating the role of C- startup routine and the exit handlers.
12. Explain _exit, exit and atexit functions with their prototypes.
13. With a neat diagram, explain the memory layout of c program. In which segments are the automatic variables and dynamically created objects are stored?
14. Write a short note on command-line arguments?
15. Explain the three functions for memory allocation and alternate memory allocators?
16. Explain setjmp and longjmp functions?
17. Explain getrlimit and setrlimit functions?
18. Explain the following system calls: i)fork ii)vfork iii)exit iv)wait.
19. Explain attributes inherited by child process and attributes that are **different between the parent and child processes:**
20. Explain the following:i)wait ii)waitpid
21. Explain the following:i)waited ii)wait3 iii)wait4
22. What is race condition? Write a program in C/C++ to illustrate a race condition
23. Giving the prototype explain different variant of exec system call.

Dr. Rekha B Venkatapur, Professor & Head, Dept of CSE, KSIT.

K.S INSTITUTE OF TECHNOLOGY, BENGALURU-560109
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
18CS56–UNIX PROGRAMMING
EXHAUSTIVE QUESTION BANK

Module 4

Question Bank

1. Discuss the applications of FIFO
2. Explain popen and pclose functions.
3. Explain different API's used for message queues
4. Explain shmget, shmctl, shmat and shmdt functions.
5. Write short notes on client server properties.
6. Explain three different methods in which client and server can get access to same IPC structure? Explain different API's used for these structure(10 Marks)
7. Explain client server structure using FIFO with a neat diagram(10M)
8. What is semaphore? Explain the API along with relevant data structure involved in implementing semaphore.
9. Write short notes on:

a. Socket

b. Shared Memory

c. Stream Pipes

10. Define FIFO? What is difference between FIFO and pipe? What are two uses of FIFO? Explain each with example. (12 M)
11. Explain the following statement:

msqid=msget(15,IPC-CREAT|0644);

Also explain the following: i) msgctl ii) msgrcv (8 M)

12. Define IPC. List the IPC types supported by UNIX system (5M)
13. How to create pipe in UNIX programming? List the limitations of pipe. (4M)
14. Develop a code snippet that the parent sends "Hello world "message to child process through the pipe. Child on receiving this message should display it on output screen. (7M)
15. Define message queue. Discuss how it is used in inter-process communication in detail. (8M)
16. Write a C/C++ program to create a shared memory segment of 100,000bytes, print first and last memory address in which the segment is attached and finally remove the shared memory segment from memory. Use the relevant shared memory segment functions. (8M)

17. Explain passing file descriptors over UNIX domain sockets with relevant structures and macros. (12M)

18. Write short notes on:

a. Streams pipe

b. Passing file descriptors

c. Co-Processes

19. Explain with diagrams setting up connld for unique connections.(10M)

20. Explain shared memory in detail maintained by kernel. (10M) 21.Which is the fastest form of IPC? Explain(10M)

21. Explain Streams based pipes. Write a C function that is used by a server to wait for a client's connecting request to arrive. (10M)

K.S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

18CS56–UNIX PROGRAMMING

EXHAUSTIVE QUESTION BANK

Module 5

1. What is signal? Explain different conditions or an event generates the signal; explain any five POSIX defined signals.
2. What is signal? Explain how to setup signal handlers?
3. Explain UNIX kernel support for handling signals.
4. What is signalmask? Explain with an example the use of signal mask.
5. Explain following signal APIs with their prototypes and uses, sigprocmask, Sigpending and sigaction.
6. Write the prototypes of all functions that are used to manipulate the signal sets.
7. Explain with an example 1) kill and raise functions 2) alarm functions.
8. Write a program to setup a real time clock interval timer using the alarm API.
9. Write a program to setup a real time clock interval timer using the settimer API.
10. What is alarm API? Give the prototype of alarm API. How can the alarm API be used to implement sleep API
11. List the timer manipulation API's in POSIX.1b
12. Explain three ways to generate log messages.
13. Explain SIGCHLD signal and waitpid API with an example. Write a program to avoid zombie using signal.
14. Explain with an example the sigsetjmp and siglongjmp functions.
15. Discuss Daemon characteristics and coding rules with an example.
16. Write functions that can be called from a program that wants to initialize itself as a Daemon.
17. Write a note on error logging.
18. What is a Signal?Write a program to setup a signal handler for the SIGINT signal using sigaction API.
19. What is daemon process?
20. Explain Kill API with programming example. Also explain kill command with an example.

Dr. Rekha B Venkatapur, Professor & Head, Dept of CSE, KSIT.

21. Categorize the ways in which the process can handle signals.
22. Write a C program that checks weather SIGINT signal is present in process signal mask and adds it to the mask if it is not there. It should clear SIGSEGV signal from signal mask.
23. Write a program to transform normal user process to deamon process. Explain every step in it

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

K S INSTITUTE OF TECHNOLOGY

Kanakapura Main Road, Raghuvanahalli, Bengaluru-560109

Department of Computer Science and Engineering

Sub: Unix Programming Sem: 3rd

Sub Code: 18CS56 Section: A & B

Challenging Questions for advanced Learners

Module 1 to Module 3

1. Explain the significance of interrupt and eof characters. With which keys are they associated on your system?

2. What are system calls and what role do they play in the system? How is C Programming so different and powerful in the Unix environment compared to Windows

3. How do you direct man page to use a specific pager?

4. Explain the significance of interrupt and eof characters. With which keys are they associated on your system?

5. What are system calls and what role do they play in the system? How is C Programming so different and powerful in the Unix environment compared to Windows

6. How do you direct man page to use a specific pager?

7.How to copy three or more words or entire file using simple key strokes?

8. Is it possible to move multiple sections of text from one file to another in a single switch?.

9. How do you compile C program without leaving the editor?

10. Use chmod –w and then try to create and remove a file in the current directory. Can you do that? Is the command the same as chmod –a-w foo

11. A file contains 1026 bytes How many bytes of disk space will it occupy on a system whether the size of a disk block is 1024 bytes.

Source:

Text Book : Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill

Model Question Paper-1 with effect from 2019-20 (CBCS Scheme)

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

Fifth Semester B.E. Degree Examination
UNIX PROGRAMMING

TIME: 03 Hours **Max. Marks: 100**

Note: 01. Answer any **FIVE** full questions, choosing at least **ONE** question from each **MODULE**.

		Module – 1	Marks
Q.1	(a)	Illustrate unix architecture with neat diagram.	08
	(b)	Discuss the silent features of UNIX operating system.	08
	(c)	What are internal and external commands in UNIX? Explain them with suitable example.	04
		OR	
Q.2	(a)	Illustrate command structure usage and behavior with respect to absolute and relative pathnames of following commands with suitable examples. i). mkdir ii). rmdir	10
	(b)	Discuss different file types available in UNIX operating system with neat diagram.	8
	(c)	Explain parent-child relationship in UNIX file system.	2
		Module – 2	**Marks**
Q.3	(a)	Which command is used for listing file attributes? Briefly describe the significance of each field of the output	08
	(b)	Current file permission of a regular file "unix" are **rw--w---x.** Illustrate both relative and absolute methods required to change permission to the following: i). –wxrwxr-x ii). r-------x iii). –w-r-x-w- iv). --xrw-r--	08
	(c)	Explain wild cards with examples and its various types.	04
		OR	
Q.4	(a)	Define is shell programming? Write a shell program to create a simple calculator which can perform basic arithmetic operations?	10
	(b)	Explain grep command with all options.	06
	(c)	Write the output for following command. i) grep ^[^3] abcd ii) grep -v "please delete" filename.txt | wc iii)ls | wc-l >fcount iv)cat *.c | wc –c	04
		Module – 3	**Marks**
Q.5	(a)	Describe general unix file API's with syntax and explain the each field in detail	10
	(b)	Explain file and record locking in detail.	06
	(c)	List the number of ways a process ...	

Save trees use e-Question Paper
Go green
DOWNLOAD THIS FREE AT www.vturesource.com

CBCS Scheme

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

15CS35

Third Semester B.E. Degree Examination, Dec.2017/Jan.2018
UNIX and Shell Programming

Time: 3 hrs. Max. Marks: 80

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

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. List and explain features of UNIX operating system. **(07 Marks)**

b. Discuss internal and external commands, with suitable examples. **(06 Marks)**

Write the outputs of the following commands :

i) cal 8 1947

ii) echo 'Todays date is `date`'

iii) date + "Date is : %a/%h/%Y". **(03 Marks)**

OR

2 a. Explain "man" documentation, and its internal commands. **(08 Marks)**

b. Describe command arguments and options with suitable examples. **(04 Marks)**

c. How an ordinary user can become a super user and vise versa? Explain with suitable commands. **(04 Marks)**

Module-2

3 a. What is a file system? Explain Unix file system with neat diagram, also explain parent and child relationships with suitable examples. **(08 Marks)**

b. What is pathname? List and explain types of path-names with an examples. **(06 Marks)**

c. Write the command line to perform the followings :

i) Change current directory to home directory

ii) Change to parent of parent directory. **(02 Marks)**

OR

4 a. What are file permissions? Describe different ways of changing the file permissions. **(07 Marks)**

b. Explain CP and Od commands with options. **(06 Marks)**

c. Write the output for the following command lines.

i) mv filenamea dir_name

ii) ls | wc -w

iii) who | wc -l. **(03 Marks)**

www.vturesource.com
ALL BRANCHES | ALL SEMESTERS | NOTES | QUESTON PAPERS | LAB MANUALS
A Vturesource Go Green initiative

Save trees use e-Question Paper Go green
DOWNLOAD THIS FREE AT www.vturesource.com

Module-3

5 a. List and explain the different modes of Vi editor, also explain different ways of quitting Vi editor. (08 Marks)

b. Discuss the following commands with respect to Vi editor.
i) b ii) w iii) | iv) G v) :1, 5w ab.txt vi) h vii) J viii) abbr. (08 Marks)

OR

6 a. What are wild cards characters? Explain each of them with suitable examples. (08 Marks)

b. What is the purpose of grep? Explain grep with all options. (06 Marks)

c. Explain tee command with an example. (02 Marks)

Module-4

7 a. Explain test command for handling strings. (04 Marks)

b. Write a shell script using case to perform all arithmetic operations. (06 Marks)

c. Explain for loop, also possible sources of argument list. (06 Marks)

OR

8 a. Explain cut command with all options with examples. (05 Marks)

b. What are links? How to create different types of links? And list their differences. (06 Marks)

c. Discuss umask and default file permissions. (05 Marks)

Module-5

9 a. Discuss how to execute commands periodically with suitable example. (05 Marks)

b. Explain find command in detail. (06 Marks)

c. What is process? Explain different mechanisms of process creation. (05 Marks)

OR

10 a. Explain string handling functions in PERL. (07 Marks)

b. Write a PERL programs check the given year is leap year or not. (07 Marks)

c. Explain split function in PERL briefly. (02 Marks)

* * * * *

www.vturesource.com
ALL BRANCHES | ALL SEMESTERS | NOTES | QUESTON PAPERS | LAB MANUALS
A Vturesource Go Green initiative

Save trees use e-Question Paper Go green
DOWNLOAD THIS FREE AT www.vturesource.com

CBCS Scheme

USN | | | | | | | | | | 15CS35

Third Semester B.E. Degree Examination, Dec.2016/Jan.2017
UNIX and Shell Programming

Time: 3 hrs. Max. Marks: 80

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

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. Discuss the salient features of UNIX Operating system. **(06 Marks)**

b. Explain the following commands with examples :
i) echo ii) ls iii) who iv) date. **(04 Marks)**

c. Write a note on man documentation and explain the keyword option and whatis option? **(06 Marks)**

OR

2 a. Explain how to display and set the terminal characteristics of a UNIX OS. **(06 Marks)**

b. Explain the contents of /etc/passwd and /etc/shadow file with respect to UNIX OS. **(06 Marks)**

c. Explain the commands to add and delete a user. **(04 Marks)**

Module-2

3 a. Explain the different file types available in UNIX. **(06 Marks)**

b. With the help of a neat diagram, explain the parent child relationship with respect to UNIX file system. **(05 Marks)**

c. Explain the following commands with example :
i) HOME ii) cd iii) pwd iv) mkdir v) rmdir. **(05 Marks)**

OR

4 a. Explain the following commands with example :
i) cat ii) mv iii) rm iv) cp v) wc. **(05 Marks)**

b. Explain the seven field output of ls - l command. **(05 Marks)**

c. What are different ways of setting file permissions? **(06 Marks)**

Module-3

5 a. Explain the different modes of vi editor. **(04 Marks)**

b. Explain how the text is entered and replaced in input mode of vi editor. **(06 Marks)**

c. Discuss the navigation commands in vi editor with example. **(06 Marks)**

OR

6 a. Explain Shell's interpretive life cycle. **(04 Marks)**

b. Discuss the three standard files supported by UNIX. Also give details about the special files used for output redirection in UNIX. **(06 Marks)**

c. With the help of example, explain grep command and list its options with their significance. **(06 Marks)**

Module-4

7 a. Explain the shell features of "while" and "for" with syntax. **(08 Marks)**

b. Explain with example set and shift commands in UNIX to manipulate positional parameters. **(08 Marks)**

www.vturesource.com
ALL BRANCHES | ALL SEMESTERS | NOTES | QUESTON PAPERS | LAB MANUALS
A Vturesource Go Green initiative

Save trees use e-Question Paper Go green
DOWNLOAD THIS FREE AT www.vturesource.com

OR

8 a. Differentiate between hard link and soft link. (04 Marks)
b. Explain the following with example :
i) head ii) tail iii) cut iv) paste. (08 Marks)
c. Discuss briefly sort command with its options. (04 Marks)

Module-5

9 a. Explain mechanism of process creation. (04 Marks)
b. Explain the following command :
i) at ii) cron iii) nice iv) nohup. (08 Marks)
c. Explain find command with its options. (04 Marks)

OR

10 a. Explain the following string handling functions of PERL with examples :
i) length ii) index iii) substr iv) reverse. (08 Marks)
b. With suitable examples, explain split and join functions in Perl. (04 Marks)
c. Explain file handling in Perl. (04 Marks)

* * * * *

www.vturesource.com
ALL BRANCHES | ALL SEMESTERS | NOTES | QUESTON PAPERS | LAB MANUALS
A Vturesource Go Green initiative

K. S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

V B STUDENTS UNIX PROGRAMMING (18CS56) COURSE END SURVEY

FACULTY NAME : Dr. REKHA B VENKATAPUR

Q1. Rate your understanding about UNIX features, architecture, structure and organization of file system

Q2. Grade your knowledge about constructing regular expression for grep command and implementing shell programs

Q3. To what extent you are able to use various categories of UNIX API's

Q4. Rate your understanding about Inter process communication using various techniques

Q5. Grade your knowledge about Signal API's and daemon processes

SL. NO.	USN	Email Address	Q1	Q2	Q3	Q4	Q5
1	1KS15CS105	jawoorsuraj@gmail.com	Very Good	Very Good	Very Good	Very Good	Very Good
2	1KS18CS117	vijetha.k.byndoor@gmail.com	Good	Good	Good	Good	Good
3	1KS18CS116	123vijayashree@gmail.com	Very Good	Very Good	Very Good	Very Good	Very Good
4	1KS18CS114	venkateshmnvenki@gmail.com	Fair	Fair	Fair	Fair	Fair
5	1KS18CS110	vaishak.puttaswamy@gmail.com	Fair	Fair	Fair	Fair	Fair
6	1KS18CS081	rakshithkumar66666@gmail.com	Good	Good	Good	Very Good	Excellent
7	1KS18CS102	sudhanshujoshi019@gmail.com	Fair	Fair	Fair	Fair	Fair
8	1KS18CS101	sudhakaryaswanth001@gmail.com	Good	Excellent	Very Good	Very Good	Very Good
9	1KS18CS090	shalinis24102000@gmail.com	Good	Good	Good	Good	Good
10	1KS18CS127	arvindpathak96445@gmail.com	Good	Good	Good	Good	Good
11	1KS18CS128	bhagyamohan3@gmail.com	Good	Good	Good	Good	Good
12	1KS18CS066	pavan2p2000@gmail.com	Good	Very Good	Excellent	Very Good	Excellent
13	1KS18CS107	swethabijanapalli@gmail.com	Good	Good	Good	Good	Good
14	1KS18CS104	nayaksunaina88@gmail.com	Good	Good	Good	Very Good	Very Good
15	1KS18CS115	vijaysingh13091999@gmail.com	Good	Good	Good	Good	Good
16	1KS18CS094	shivaprakash832000@gmail.com	Good	Good	Good	Fair	Fair
17	1KS18CS126	rayyaan.m786@gmail.com	Very Good	Good	Good	Very Good	Very Good
18	1KS18CS130	likitha6065@gmail.com	Good	Good	Good	Good	Good
19		veerasreenidhi@gmail.com	Good	Good	Good	Good	Good
20	1KS19CS413	smilesweaty6655@gmail.com	Very Good	Excellent	Excellent	Excellent	Excellent
21	1KS18CS099	vidyavidu2000@gmail.com	Excellent	Excellent	Very Good	Excellent	Very Good
22	1KS18CS076	r.pratiksha1713@gmail.com	Good	Good	Good	Good	Good
23		sujay.suresh28@gmail.com	Excellent	Excellent	Very Good	Very Good	Good
24	1KS18CS100	subramanyagowda8123@gmail.com	Good	Good	Good	Good	Good
25	1KS18CS084	rubaabdulrahman456@gmail.com	Good	Good	Good	Good	Good
26	1KS18CS079	shreyaraipalle@gmail.com	Very Good	Very Good	Very Good	Excellent	Very Good
27	1KS18CS095	shubhashinir05204@gmail.com	Good	Good	Good	Good	Good
28	1KS18CS064	nlight110@gmail.com	Good	Good	Good	Good	Good
29	1KS18CS082	rekhancsagar107@gmail.com	Good	Good	Very Good	Good	Very Good
30	1KS18CS077	ramyareddy9488@gmail.com	Good	Good	Good	Fair	Good
31	1KS18CS092	shashankmishra.0598@gmail.com	Very Good	Very Good	Very Good	Very Good	Very Good
32	1KS18CS124	shewanichib01@gmail.com	Good	Good	Good	Good	Good
33	1KS18CS120	vyju72000@gmail.com	Good	Good	Good	Good	Fair
34	1KS19CS401	arpithagopal505@gmail.com	Good	Good	Good	Good	Good
35	1KS18CS080	rakshakaranth.s@gmail.com	Fair	Fair	Fair	Fair	Fair
36	1KS18CS063	nitiesh.mr@gmail.com	Good	Good	Good	Good	Good
37	1KS18CS067	poojakshree@gmail.com	Very Good	Very Good	Very Good	Excellent	Very Good
38	1KS18CS131	shalinikp96@gmail.com	Very Good	Excellent	Very Good	Very Good	Very Good

SL. NO.	USN	Email Address	Q1	Q2	Q3	Q4	Q5
39	1KS18CS096	0908.sindu@gmail.com	Very Good	Very Good	Very Good	Very Good	Very Good
40	1KS18CS108	hemanthchowdary743@gmail.com	Good	Very Good	Very Good	Good	Excellent
41	1KS18CS075	dekshi17@gmail.com	Very Good	Very Good	Very Good	Very Good	Very Good
42	1KS18CS083	rithananraj@gmail.com	Fair	Fair	Fair	Fair	Fair
43	1KS18CS098	srichandana1020@gmail.com	Good	Good	Good	Good	Good
44	1KS19CS404	bhavyajanu77@gmail.com	Good	Very Good	Good	Good	Very Good
45	1KS18CS097	sourabhsk112@gmail.com	Good	Very Good	Good	Very Good	Good
46	1KS18CS087	sandeepkr2909@gmail.com	Good	Good	Good	Good	Good
47		dvishnupriya1112@gmail.com	Good	Good	Good	Very Good	Good
48	1KS18CS086	samhithav1999@gmail.com	Very Good	Good	Good	Good	Very Good
49	1KS18CS122	yogitaramesh11@gmail.com	Good	Good	Good	Good	Good
50	1KS18CS073	pujarivishnupriya2000@gmail.com	Very Good	Good	Fair	Good	Good
51	1KS18CS069	college@pranavms.ml	Very Good	Good	Very Good	Very Good	Very Good
52	1KS18CS121	pathakamuriyashwanth@gmail.com	Good	Good	Good	Good	Good
53	1KS18CS070	prateekhavale27@gmail.com	Good	Good	Good	Good	Fair
54	1KS18CS118	vinodmalali139@gmail.com	Good	Good	Good	Good	Good
55	1KS18CS111	varidhim08@gmail.com	Good	Good	Good	Good	Good
56	1KS18CS112	vedavedya007@gmail.com	Good	Good	Good	Good	Good
57	1KS18CS109	thirumalaishaktivel@gmail.com	Good	Good	Good	Good	Good
58	1KS18CS129	aishakulsum0404@gmail.com	Excellent	Excellent	Excellent	Excellent	Excellent
59	1KS18CS078	rgrahul@icloud.com	Fair	Good	Fair	Good	Fair
60	1KS18CS089	sauravsmakam30@gmail.com	Very Good	Very Good	Very Good	Very Good	Very Good
61	1KS18CS093	shivi.aol16@gmail.com	Very Good	Very Good	Excellent	Very Good	Very Good
62	1KS18CS065	psairam360@gmail.com	Fair	Fair	Fair	Fair	Fair
63	1KS18CS071	kujmarkp547@gmail.com	Fair	Fair	Fair	Fair	Fair
64	1KS18CS072	preethisuma9@gmail.com	Fair	Fair	Fair	Fair	Fair
65	1KS18CS074	ppavankumar727@gmail.com	Good	Good	Good	Good	Good
66	1KS18CS091	shashankganesh68@gmail.com	Good	Good	Good	Good	Good
67	1KS18CS103	sujay.suresh28@gmail.com	Good	Good	Good	Good	Good
68	1KS18CS105	jawoorsuraj@gmail.com	Good	Good	Good	Good	Good
69	1KS18CS106	sushmithas.bdvt@gmail.com	Good	Good	Good	Good	Good
70	1KS18CS123	khanzaina307@gmail.com	Good	Good	Good	Good	Good
71	1KS18CS113	veerasreenidhi@gmail.com	Good	Good	Good	Good	Good
72	1KS18CS125	soumyahegde984@gmail.com	Good	Good	Good	Good	Good
73	1KS17CS015	gagnravi1104@gmail.com	Good	Good	Good	Good	Good
		Verry Good	15	13	15	17	18
		Good	46	46	44	41	38
		Fair	9	8	10	10	12
		Excellent	3	6	4	5	5

Faculty Signature

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

INDIRECT ATTAINMENT COURSE END SURVEY

YEAR / SEMESTER	III/ V
COURSE TITLE	UNIX Programming
COURSE CODE	18CS56
ACADEMIC YEAR	2020-21

	Q1	Q2	Q3	Q4	Q5	
EXCELLENT	25	28	25	29	26	
VERY GOOD	38	34	35	34	35	
GOOD	62	62	60	57	59	91%
SATISFACTORY	9	10	14	14	14	
STUDENTS RESPONSE(GOOD & ABOVE)	93%	93%	90%	90%	90%	

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

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	III / V
COURSE TITLE	UNIX PROGRAMMING
COURSE CODE	18CS56
ACADEMIC YEAR	2020-2021

Sl. No	USN	Name of the Student	18CS62															
			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	1KS18CS001	ADARSH K	17	10	6	4	6	17	5	2	5	2	6	9	4	6	39	47
2	1KS18CS002	AMARAVATHI M	15	5	6	3	6	16	6	2	6	2	7	6	4	6	35	42
3	1KS18CS003	ANIKETH.H	15	10	6	4	6	18	5	2	5	2	6	18	4	6	21	26
4	1KS18CS005	ANUSHRUTI SINGH	15	12	6	4	6	18	5	2	6	2	7	11	4	6	21	26
5	1KS18CS006	ARUNA P	17	12	6	4	6	17	5	2	6	2	7	15	4	6	41	50
6	1KS18CS007	ASHWINI J	18	12	6	4	6	18	6	2	6	2	15	9	4	6	38	46
7	1KS18CS008	AVINASH PRASAD	16	11	6	4	6	18	6	2	6	2	11	14	4	6	30	36
8	1KS18CS009	B DEVA DEEKSHITH	17	11	6	4	6	17	6	2	5	2	11	10	4	6	35	42
9	1KS18CS010	BHAGWAT GOUTAM	14	10	6	4	5	16	6	2	5	2	5	12	4	6	24	29
10	1KS18CS012	BHOOMIKA H	15	10	6	4	6	17	6	2	6	2	6	17	4	6	40	48
11	1KS18CS013	BHUVANA CHANDRIKA GANTI	15	11	6	4	6	17	6	2	6	2	11	16	4	6	42	51
12	1KS18CS014	BRIJESH.S	14	11	6	3	6	15	6	2	4	2	3	11	4	6	21	26
13	1KS18CS015	CHAITHRA R	18	11	6	4	6	17	6	2	6	2	12	16	4	6	36	44
14	1KS18CS016	CHANDAN KUMAR	15	11	6	4	6	16	5	2	6	2	4	8	4	6	26	32
15	1KS18CS017	DANDU NIHARIKA	15	11	6	4	6	14	6	2	6	2	12	17	4	6	37	45
16	1KS18CS018	DHANANJAYA S	18	11	6	4	6	18	6	2	6	2	10	11	4	6	33	40
17	1KS18CS019	DHRUV JYOTI SHUKLA	18	12	6	4	5	17	6	2	6	2	11	12	4	6	27	33
18	1KS18CS020	FARIYA N	15	12	6	4	6	15	6	2	6	2	10	13	4	6	33	40
19	1KS18CS022	GAGANSURI M S	18	11	6	4	6	16	6	2	5	2	7	15	4	6	28	34
20	1KS18CS023	GANESH A	14	11	6	4	6	14	6	2	6	2	9	11	4	6	41	50
21	1KS18CS024	GOUTHAM M	15	11	6	4	6	17	6	2	6	2	10	10	4	6	47	57
22	1KS18CS025	GUNAL BINANI	15	11	6	4	6	12	NA	2	4	2	NA	NA	4	6	22	27
23	1KS18CS026	HARSHITH C PRASAD	14	11	6	4	6	14	6	2	4	2	NA	NA	4	6	29	35
24	1KS18CS027	KANDIMALLA KRISHNA PAVITHRA	14	12	6	4	5	17	6	2	6	2	6	13	4	6	27	33
25	1KS18CS028	KARTHIK K	15	11	6	4	6	17	6	2	6	2	7	4	4	6	21	26
26	1KS18CS029	KAVITA CHAUDHARY	16	11	6	4	6	17	6	2	6	2	10	14	4	6	35	42
27	1KS18CS030	KENCHAM ARUN	14	11	6	4	6	15	6	2	2	2	5	7	4	6	36	44
28	1KS18CS031	KILARI ASHWIK	17	11	6	4	6	18	6	2	6	2	5	16	4	6	21	26

29	1KS18CS032	KILARI JASWANTH	15	12	6	4	6	17	6	2	2	2	12	16	4	6	32	39
30	1KS18CS033	KIRAN VEERANNA DAMBAL	16	11	6	4	6	17	6	2	6	2	6	13	4	6	37	45
31	1KS18CS034	KRITHIKA.K.N	16	11	6	4	6	18	6	2	6	2	12	6	4	6	27	33
32	**1KS18CS035**	KRUTHIKA.S.VASISHT	15	11	6	4	5	18	5	2	6	2	9	6	4	6	33	40
33	1KS18CS036	LEKKALA SHARANDEEP CHOWDARY	16	11	6	4	4	18	6	2	4	2	8	2	4	6	39	47
34	1KS18CS037	LATHA V	15	11	6	4	6	18	6	2	6	2	10	17	4	6	46	56
35	1KS18CS038	LAVANYA.C.R	15	11	6	4	6	16	6	2	6	2	6	18	4	6	37	45
36	1KS18CS039	LIKHITHA.N	14	6	6	4	6	18	6	2	6	2	6	6	4	6	42	51
37	1KS18CS040	LOKESH R	14	11	6	4	6	17	6	1	6	2	10	14	4	6	42	51
38	1KS18CS041	MADDULA JITENDRA	16	11	6	4	6	16	6	2	6	2	8	7	4	6	26	32
39	1KS18CS042	MADHUSUDHAN.S.R	16	11	6	4	6	15	6	2	6	2	10	10	4	6	32	39
40	1KS18CS043	MAHARAJ S	14	11	6	4	6	18	5	2	5	2	NA	NA	4	6	16	20
41	1KS18CS044	MAHESH B V	17	11	5	3	6	17	6	2	6	2	7	1	4	6	28	34
42	1KS18CS045	MANIKONDA THARUN	16	11	6	4	6	12	6	2	5	2	8	10	4	6	37	45
43	1KS18CS046	MANVITH P	17	11	6	4	6	14	6	1	2	2	9	6	4	6	21	26
44	1KS18CS047	MD SUJAN	13	12	6	4	6	13	5	2	6	2	8	9	4	6	27	33
45	1KS18CS049	MEGHASHREE A	15	11	6	4	6	17	6	2	5	2	11	10	4	6	31	38
46	1KS18CS050	MIKKIN K M	15	10	6	4	6	17	6	2	6	2	8	7	NA	NA	26	32
47	1KS18CS051	MONICA S	15	12	6	4	6	18	6	2	6	2	12	18	4	6	45	54
48	1KS18CS052	MONIKA.K.C	18	11	6	4	6	18	6	2	6	2	8	8	4	6	38	46
49	1KS18CS053	MOPURI SREELAKSHMI	18	11	6	3	6	18	6	2	6	2	11	15	4	6	31	38
50	1KS18CS054	N SAI JAHANAVI	15	12	6	4	6	14	6	2	6	2	9	11	NA	NA	29	35
51	1KS18CS055	NAGARJUN N	16	11	5	3	6	15	6	2	4	2	NA	NA	4	6	23	28
52	1KS18CS056	NANDINI J K	15	10	6	4	6	15	6	2	6	2	9	10	4	6	43	52
53	1KS18CS057	NARASIMHA MAIYA G S	17	11	6	4	6	17	6	2	6	2	11	16	4	6	48	58
54	1KS18CS058	NARASIMHARAJU R	15	11	5	3	NA	NA	NA	2	2	2	3	15	4	6	AB	AB
55	1KS18CS059	NIKHIL.M	10	11	6	4	6	12	6	2	6	2	6	4	4	6	22	27
56	1KS18CS060	NIKHIL VASAN	13	11	6	4	6	8	6	1	6	2	5	7	4	6	35	42
57	1KS18CS061	NIKIL B S	16	11	6	4	6	17	6	2	6	2	11	11	4	6	40	48
58	1KS18CS062	NIKITHA M	18	12	6	4	6	18	6	2	6	2	12	17	4	6	49	59
59	1KS18CS063	NITISH KUMAR.M.R	18	5	4	5	6	12	6	2	6	2	7	1	4	6	29	35
60	1KS18CS064	NOOR SUMAIYA	16	12	5	3	6	18	6	2	6	2	6	13	4	6	29	35
61	1KS18CS065	P SAI RAM	17	11	4	3	6	18	6	2	6	2	10	17	NA	NA	42	51
62	1KS18CS066	PAVAN P	17	12	5	4	6	18	6	2	6	2	11	14	4	6	35	42
63	1KS18CS067	POOJASHREE K	18	12	5	4	6	18	6	2	6	2	12	18	4	6	42	51
64	1KS18CS069	PRANAV M S	16	8	5	4	6	16	6	NA	NA	NA	6	9	4	6	25	30
65	1KS18CS070	PRATEEK HAVALE	18	10	6	4	6	18	6	2	6	2	12	17	4	6	39	47
66	1KS18CS071	PRAVEEN KUMAR K	18	11	5	4	6	18	6	2	6	2	12	18	4	6	42	51
67	1KS18CS072	PREETHI K	17	12	5	2	6	18	6	NA	NA	NA	6	6	NA	NA	31	38
68	1KS18CS073	PUJARI VISHNU PRIYA	17	12	6	4	6	18	6	2	6	2	12	18	4	6	35	42
69	1KS18CS074	PULLUR PAVAN KUMAR	17	12	5	4	6	18	6	2	6	2	10	18	4	6	34	41
70	**1KS18CS075**	R DEKSHITHA	18	12	5	4	6	18	6	2	6	2	12	18	4	6	29	35
71	1KS18CS076	R PRATIKSHA	18	12	5	4	6	18	6	2	6	2	12	18	4	6	46	56
72	1KS18CS077	RAMYA R	18	11	5	3	6	18	6	2	6	2	11	18	NA	NA	23	28
73	1KS18CS078	RAHUL.P	16	12	NA	NA	6	18	6	NA	NA	NA	6	10	NA	NA	22	27

74	1KS18CS079	RAIPALLE SHREYAA	18	12	5	1	6	16	6	2	6	2	11	9	4	6	34	41
75	1KS18CS080	RAKSHA S	18	12	3	2	6	18	6	NA	NA	NA	11	5	4	6	37	45
76	1KS18CS081	RAKSHITH KUMAR.N	18	12	5	4	6	18	6	2	6	2	9	13	4	6	41	50
77	1KS18CS082	REKHA N C	18	12	5	4	6	18	6	2	6	2	11	18	4	6	42	51
78	1KS18CS083	RITHANA.N.RAJ	18	6	3	1	6	18	6	2	6	2	9	5	4	6	21	26
79	1KS18CS084	RUBA ABDUL RAHMAN	18	12	4	1	6	18	6	NA	NA	NA	NA	13	NA	NA	23	28
80	1KS18CS086	SAMHITHA	18	12	4	5	6	18	6	2	6	2	8	11	4	6	21	26
81	1KS18CS087	SANDEEP KUMAR	18	11	2	3	6	18	6	2	6	2	11	9	NA	NA	26	32
82	1KS18CS088	SAURAV KUMAR	18	11	5	4	6	18	6	2	6	2	NA	NA	NA	NA	AB	AB
83	1KS18CS089	SAURAV S MAKAM	18	12	3	NA	6	18	6	NA	NA	NA	10	10	NA	NA	21	26
84	1KS18CS090	SHALINI S	18	12	5	4	6	18	6	2	6	2	12	18	4	6	44	53
85	1KS18CS091	SHASHANK G	18	11	6	4	6	18	6	2	6	2	12	18	4	6	36	44
86	1KS18CS092	SHASHANK MISHRA	14	12	4	4	6	18	6	2	6	2	12	6	4	6	15	18
87	1KS18CS093	SHIVANGI SRIVASTAVA	18	12	4	4	6	18	6	NA	NA	NA	11	16	NA	NA	41	50
88	1KS18CS094	SHIVA PRAKASH T	13	11	6	4	6	18	6	2	6	2	6	18	4	6	32	39
89	1KS18CS095	SHUBHASHINI.R	17	10	5	4	6	18	6	2	6	2	12	18	NA	NA	38	46
90	1KS18CS096	SINDU A S	18	12	6	4	6	18	6	2	6	2	11	16	4	6	40	48
91	1KS18CS097	SOURABH SANTOSH KAMBLE	17	11	5	4	6	12	6	2	6	2	12	17	4	6	24	29
92	1KS18CS098	SRI CHANDANA P	18	12	6	4	6	18	6	2	6	2	12	12	4	6	35	42
93	1KS18CS099	SRIVIDYA H R	18	12	6	4	6	12	6	2	6	2	12	13	4	6	29	35
94	1KS18CS100	SUBRAMANYA N	18	12	NA	5	6	16	6	NA	NA	NA	6	12	NA	NA	32	39
95	1KS18CS101	SUDHAKAR YASWANTH	18	11	5	4	6	NA	6	2	6	2	12	16	4	6	35	42
96	1KS18CS102	SUDHANSHU JOSHI	18	11	5	4	6	18	6	2	6	2	12	18	4	6	46	56
97	1KS18CS103	SUJAY G S	18	12	5	4	6	18	6	2	6	2	12	18	4	6	35	42
98	1KS18CS104	SUNAINA NAYAK	18	11	6	4	6	18	6	2	6	2	12	18	4	6	52	63
99	1KS18CS105	SURAJ C JAWOOR	18	11	3	2	6	18	6	NA	NA	NA	3	6	NA	NA	31	38
100	1KS18CS106	SUSHMITHA S	18	10	5	4	6	18	6	2	6	2	6	10	4	6	33	40
101	1KS18CS107	SWETHA BIJANAPALLI	18	10	5	2	6	18	6	2	6	2	12	11	4	6	35	42
102	1KS18CS108	THAMMINENI HEMANTH CHOWDARY	18	11	5	4	6	18	6	2	6	2	12	16	4	6	43	52
103	1KS18CS109	THIRUMALAI SHAKTIVEL C	18	12	6	4	6	18	6	2	6	2	12	18	4	6	42	51
104	1KS18CS110	VAISHAK P	15	12	6	4	6	18	6	2	6	2	12	18	4	6	32	39
105	1KS18CS111	VARIDHI MADHURANATH	18	10	5	4	6	18	6	2	6	2	12	18	4	6	36	44
106	1KS18CS112	VEDAVEDYA B H	18	10	5	3	6	18	6	2	6	2	7	2	4	6	35	42
107	1KS18CS113	VEERA SREENIDHI.R	18	11	5	2	6	15	6	NA	NA	NA	11	12	4	6	23	28
108	1KS18CS114	VENKATESH M N	18	10	3	1	6	18	6	2	6	2	6	3	4	6	24	29
109	1KS18CS115	VIJAY.N.S	18	10	NA	4	6	16	6	2	6	2	11	8	4	6	27	33
110	1KS18CS116	VIJAYASHREE.N.R	18	12	2	3	6	18	6	2	6	2	12	18	4	6	28	34
111	1KS18CS117	VIJETHA	18	12	5	3	6	18	6	2	6	2	6	12	4	6	37	45
112	1KS18CS118	VIINOD H MALALI	18	12	6	4	6	18	6	2	6	2	12	18	4	6	42	51
113	1KS18CS119	VISHNUPRIYA D	18	12	NA	4	6	18	6	2	6	2	6	17	4	6	48	58
114	1KS18CS120	VYJAYANTHI K S	18	11	5	4	6	18	6	NA	NA	NA	12	18	NA	NA	23	28
115	1KS18CS121	YASHWANTH.K	18	6	5	4	6	18	6	2	6	2	12	16	4	6	25	30
116	1KS18CS122	YOGITA RAIKAR	18	12	6	4	6	18	6	2	6	2	12	17	4	6	39	47
117	1KS18CS123	ZAINA KHAN	18	12	5	4	6	18	6	2	6	2	6	17	4	6	44	53
118	1KS18CS124	SHEWANI CHIB	18	11	4	2	6	18	6	2	6	2	11	18	4	6	30	36

119	1KS18CS125	R SOUMYA	18	12	6	4	6	18	6	2	6	2	12	18	4	6	51	62
120	1KS18CS126	RAYYAAN MOHIADDIN	18	11	4	3	6	18	6	NA	NA	NA	11	16	NA	NA	AB	AB
121	1KS18CS127	ARVIND PATHAK	18	11	5	3	6	16	6	2	6	2	11	16	4	6	39	47
122	1KS18CS128	BHAGYASHREE.V	18	12	5	2	6	18	6	2	6	2	9	15	4	6	29	35
123	1KS18CS129	BI BI AYESHA	18	12	NA	3	6	18	6	2	6	2	12	18	4	6	32	39
124	1KS18CS130	LIKITHA.S	18	12	5	2	6	18	NA	2	6	2	12	17	4	6	39	47
125	1KS18CS131	SHALINI.K.P	18	12	5	2	6	18	6	2	6	2	12	18	4	6	21	26
126	1KS19CS400	AKSHAY.B.R	15	12	6	4	6	15	6	2	5	2	NA	NA	4	6	39	47
127	1KS19CS401	ARPITHA.G	18	12	3	1	6	16	6	2	6	2	4	12	4	6	21	26
128	1KS19CS402	B.K.SUSMITHA	16	9	6	4	6	13	6	2	6	2	6	6	4	6	42	51
129	1KS19CS403	BHAVANI.K.G	16	11	6	4	6	15	6	1	6	2	4	11	4	6	41	50
130	1KS19CS404	BHAVYASHREE.R	18	12	3	1	6	16	6	2	6	2	5	10	NA	NA	21	26
131	1KS19CS405	CHAITRA.K	16	10	6	4	NA	NA	NA	2	6	2	6	11	4	6	32	39
132	1KS19CS406	DHANALAKSHMI.B	12	6	6	3	6	7	6	2	5	2	NA	NA	4	6	22	27
133	1KS19CS407	GOLLA YASWANTH	9	11	6	4	6	8	6	1	4	2	NA	NA	4	6	28	34
134	1KS19CS408	KALPITHA.A.J	15	12	6	4	5	12	6	2	5	2	12	6	4	6	26	32
135	1KS19CS409	KARTHIK PRAKASH HUDEDAMANI	16	12	6	3	6	16	6	2	5	2	NA	NA	4	6	38	46
136	1KS19CS410	RAMYA.R	15	10	6	4	5	15	4	2	5	2	3	9	4	6	28	34
137	1KS19CS411	RANJITHA.H.D	11	12	6	3	6	14	6	2	5	2	NA	NA	4	6	39	47
138	1KS19CS412	RUSHI.C.S	15	11	6	4	6	13	6	3	4	2	4	9	4	6	32	39
139	1KS19CS413	SAHANA.V	18	12	5	2	6	18	6	2	6	2	6	18	NA	NA	30	36
140	1KS19CS414	Y.MRUDULA JAIN	18	12	5	2	6	18	6	2	6	2	6	12	NA	NA	36	44
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			139	135	126	121	138	134	136	123	124	128	85	86	121	121	87	87
Total number of students (Y)			140	140	135	138	138	137	136	128	128	128	129	130	121	121	137	137
CO Percentage			99.29	96.43	93.33	87.68	100.00	97.81	100.00	96.09	96.88	100.00	65.89	66.15	100.00	100.00	63.50	63.50
			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	96.31	63.50	79.91	3.00	91.00	3.00	3
CO2	95.05	63.50	79.28	3.00	91.00	3.00	3
CO3	97.34	63.50	80.42	3.00	91.00	3.00	3
CO4	91.47	63.50	77.49	3.00	91.00	3.00	3
CO5	83.08	63.50	73.29	3.00	91.00	3.00	3
AVERAGE							3.00

	IA1	A1	IA2	A2	IA3	A3	AVG
CO1	99.29	93.33					96.31
CO2	96.43	87.68	100.00	96.094			95.05
CO3			97.81	96.875			97.34
CO4			100.00	100	65.89	100	91.47
CO5					66.15	100	83.08

CO Attainment Level	Significance
Level 3	60% and above students should have scored >= 60% of Total marks
Level 2	55% to 59% of students should have scored >= 60% of Total marks
Level 1	50% to 54% of students should have scored >= 60% of Total marks

For Direct attainment , 50% of CIE and 50% of SEE marks are considered.

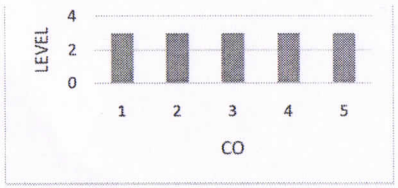
For indirect attainment, Course end survey is considered.

CO attainment is 90%of direct attainment + 10% of Indirect atttainment.

PO attainment = CO-PO mapping strength/3 * CO attainment .

CO ATTAINMENT

Co-Po Mapping Table



CO'S	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PSO2
CO1	3	1	1	-	2	-	-	-	2.00	-	-	-	2	
CO2	3	2	2	-	2	-	-	-	2.00	-	-	-	2	2
CO3	3	2	2	-	2	-	-	-	2.00	-	-	-	2	2
CO4	3	2	2	-	2	-	-	-	-	-	-	-	2	2
CO5	3	1	1	-	-	-	-	-	-	-	-	-	2	2
AVG	3.00	1.60	1.60	-	2.00	-	-	-	2.00	-	-	-	2.00	2.00

PO ATTAINMENT TABLE

CO'S	CO Attainment in %	CO RESULT	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PS02
CO1	3.00	Y	3.00	1.00	1.00	-	2.00	-	-	-	2	-	-	-	2.00	0.00
CO2	3.00	Y	3.00	2.00	2.00	-	2.00	-	-	-	2	-	-	-	2.00	2.00
CO3	3.00	Y	3.00	2.00	2.00	-	2.00	-	-	-	2	-	-	-	2.00	2.00
CO4	3.00	Y	3.00	2.00	2.00	-	2.00	-	-	-	-	-	-	-	2.00	2.00
CO5	3.00	Y	3.00	1.00	1.00	-	-	-	-	-	-	-	-	-	2.00	2.00
Average			3.00	1.60	1.60	-	2.00	-	-	-	2.00	-	-	-	2.00	2.00

Course Incharge

HOD

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

MODULE - 1

Syllabus :
Introduction: .Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. General features of Unix commands/ command structure. Command arguments andoptions. Basic Unix commands such as echo, printf, ls, who, date, passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The root login. Becoming the super user: su command.

Topics from chapter 2 , 3 and 15 of text book 1,chapter 1 from text book 2

Text Book 1: Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill
Text Book 2: Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. 2009.

Introduction

An operating system is a control program for a computer that performs the following operations:

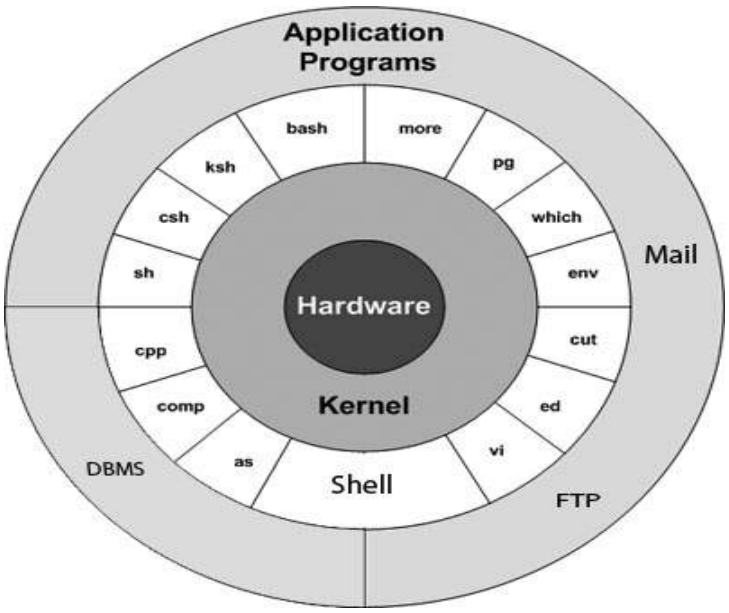
- allocates computer resources
- schedules routine tasks
- provides a platform to run application software for users to accomplish tasks
- provides an interface between the user & the computer

UNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops. UNIX systems also have a graphical user interface (GUI) similar to Microsoft Windows which provides an easy to use environment. However, knowledge of UNIX is required for operations which aren't covered by a graphical program, or for when there is no windows interface available, for example, in a telnet session. Unix was developed by Ken Thompson and Dennis Ritchie in AT & T Lab in 1969. Unix is basically Network Operating System generally used with Virtual Console (CLI). Everything is case sensitive in Unix including Usernames, commands, passwords and filenames.

Brief History

- The first versions of UNIX were written in "machine-dependent" program (such as PDP-7).
- Ken Thompson approach Dennis Ritchie developer of C program), and in 1973 they compiled UNIX in C programming language to make operating system "portable" to other computers systems.

Unix Components/UNIX Architecture



UNIX architecture comprises of two major components viz., the shell and the kernel. The kernel interacts with the machine's hardware and the shell with the user.

- The entire UNIX system is supported by a handful of essentially simple and abstract concepts.
- The success of UNIX, according to Thompson and Ritchie, "lies not so much in new inventions but rather in the full exploitation of a carefully selected fertile ideas, and especially in showing that they can be keys to the implementation of a small and yet powerful operating system".
- UNIX is no longer a small system, but it certainly is a powerful one.
- The UNIX architecture has three important agencies-
    - Division of labor: Kernel and shell
    - The file and process
    - The system calls

- ***Division of labor: Kernel and shell***
- The fertile ideas in the development of UNIX has two agencies – kernel and shell.
- The kernel interacts with the machine's hardware.
- The shell interacts with the user.

The Kernel

- The core of the operating system - a collection of routines mostly written in C.

- It is loaded into memory when the system is booted and communicates directly with the hardware.
- User programs (the applications) that need to access the hardware use the services of the kernel, which performs the job on the user's behalf.
- These programs access the kernel through a set of functions called system calls.
- Apart from providing support to user's program, kernel also does important housekeeping.
- It manages the system's memory, schedules processes, decides their priorities and so on.
- The kernel has to do a lot of this work even if no user program is running.
- The kernel is also called as the operating system - a programs gateway to the computer's resources.

The Shell

- Computers don't have any capability of translating commands into action.
- That requires a **command interpreter**, also called as the shell.
- Shell is actually interface between the user and the kernel.
- Most of the time, there's only one kernel running on the system, there could be several shells running – one for each user logged in.
- The shell accepts commands from user, if require rebuilds a user command, and finally communicates with the kernel to see that the command is executed.

Type of Shells

- Bourne shell (sh)
- C shell (csh)
- Korn shell (ksh)
- Bourne Again Shell (bash)

At one time only one shell runs.

Features of UNIX OS

UNIX is an operating system, so it has all the features an operating system is expected to have.

- A Multiuser System
- A Multitasking System
- The building-block approach
- The UNIX toolkit
- Pattern Matching
- Programming Facility
- Documentation
- Portability
- Organized file system
- Device Independence

- Utilities

A Multiuser System

- UNIX is a multiprogramming system, it permits multiple programs to run and compete for the attention of the CPU.
- This can happen in two ways:
- Multiple users can run separate jobs
- A single user can also run multiple jobs

A Multitasking System

- A single user can also run multiple tasks concurrently.
- UNIX is a multitasking system.
- It is usual for a user to edit a file, print another one on the printer, send email to a friend and browse www - all without leaving any of applications.
- The kernel is designed to handle a user's multiple needs.
- In a multitasking environment, a user sees one job running in the foreground; the rest run in the background.
- User can switch jobs between background and foreground, suspend, or even terminate them.

The Building-block Approach

- The designer never attempted to pack too many features into a few tools.
- Instead, they felt "**small is beautiful**", and developed a few hundred commands each of which performed one simple job.
- UNIX offers the | (filters) to combine various simple tools to carry out complex jobs.
- Example:
  - **$ cat note** #cat displays the file contents WELCOME TO HIT
  - **$ cat note | wc** #wc counts number of lines, words & characters in the file 1 3 15

The UNIX Toolkit

- Kernel itself doesn't do much useful task for users
- UNIX offers facility to add and remove many applications as and when remove many applications as and when required.
- Tools include general purpose tools, Text manipulation tools, Compilers, interpreters Networked applications and system administration tools.

Networking

While UNIX was developed to be an interactive, multiuser, multitasking system, networking is also incorporated into the heart of the operating system. Access to another

system uses a standard communications protocol known as Transmission Control Protocol/Internet Protocol (TCP/IP).

Utilities

UNIX provides a rich library of utilities that can be used to increase user productivity,often referred to as commands. Accomplish universal functions – editing, file maintenance, printing, sorting, programming support, online info.

Pattern Matching

- UNIX features very sophisticated pattern matching features.
- Example: The * (zero or more occurrences of characters) is a special character used by system to indicate that it can match a number of filenames.

Programming Facility

- The UNIX shell is also a programming language; it was designed for programmer, not for end user.
- It has all the necessary ingredients, like control structures, loops and variables, that establish powerful programming language.
- This features are used to design shell scripts – programs that can also invoke UNIX commands.
- Many of the system's functions can be controlled and automated by using these shell scripts.

Documentation

- The principal on-line help facility available is the man command, which remains the most important references for commands and their configuration files.
- Apart from the man documentation, there's a vast ocean of UNIX resources available on the Internet.

Portability

UNIX can be installed on many hardware platforms. Its widespread use can be traced to the decision to develop it using the C language.

Organized File System

UNIX has a very organized file and directory system that allows users to organize and maintain files.

Device Independence

UNIX treats input/output devices like ordinary files. The source or destination for file input and output is easily controlled through a UNIX design feature called redirection.

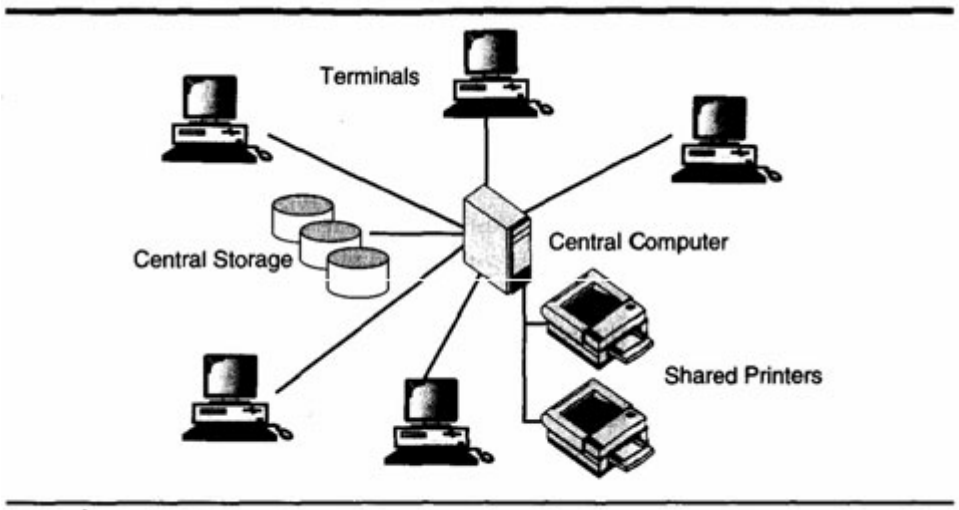
Unix Environment

Different Computing Environments
1. Stand alone Personal Environment
2. Time sharing Environment
3.Client/server system

Personal environment

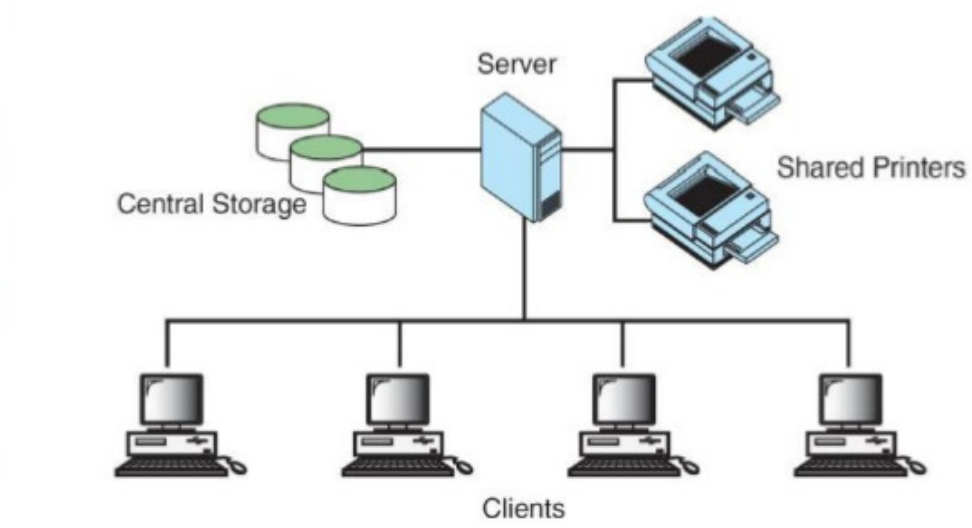
With availability of Linux, a free Unix system personal unix system trend is accelerated

Time sharing environment



Time-sharing Environment

Many Users connected to one or more computers.
All computing must be done by the central computer which has many devices.
Central computer has to control shared resources, data and printers.
Client/Server Environment



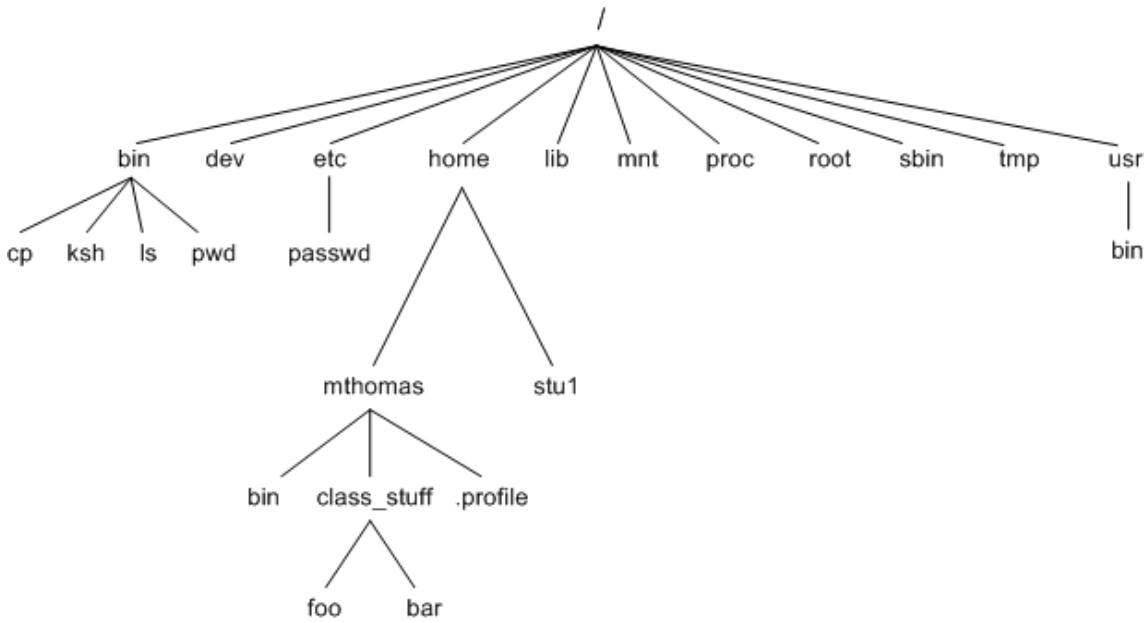
The computing function is split between a central computer(powerful mainframe) and users computers(Micro computers). Users use personal computers/workstations. The responsibility of central computer is moved from central computer and it is assigned to personal computers.

Central computer – server

Microcomputer - client

Advantage –Response time and monitor display are faster, users are more productive.

UNIX File Structure



The File and Process

"Files have places and processes have life".
All files are "flat": just a sequence of bytes File system is hierarchical.

A file is an array of bytes that stores information. It is also related to another file in the sense that both belong to a single hierarchical directory structure.

A process is the second abstraction UNIX provides. It can be treated as a time image of an executable file. Like files, processes also belong to a hierarchical structure. We will be discussing the processes in detain in a subsequent chapter.

Locating Files

All UNIX commands are single words like ls, cd, cat, etc. These names are in lowercase. These commands are essentially *files* containing programs, mainly written in C. Files are stored in directories, and so are the binaries associated with these commands. You can find the location of an executable program using type command:

```
$ type ls
ls is /bin/ls
```

This means that when you execute ls command, the shell locates this file in /bin directory and makes arrangements to execute it.

In bash shell ls is shellbuiltin

The Path

The sequence of directories that the shell searches to look for a command is specified in its own PATH variable. These directories are colon separated. When you issue a command, the shell searches this list in the sequence specified to locate and execute it.

POSIX and Single Unix Specification

POSIX:
Portable Operating System Interface for computing environments(POSIX) developed by IEEE. Two standards of POSIX family are
POSIX.1 : Deals with C application program interface – the system calls
POSIX.2 : Deals with shell and utilities

Single UNIX Specification:
Single Unix Specification version 3(SUSV3) is developed by X/Open and IEEE – Write once on any POSIX complaint UNIX system &adopt everywhere. Feature - Easy portability to any other POSIX complaint machine.

The Login Prompt

Login prompt indicates that the terminal is available to login (connect to machine) .Indiactes to user that he can enter with login name
login: kumar
Password:
System is now requesting for secrete code (Known only to you)

Login:kumar
Password:*******
The string entered by user as password is not displayed on screen. This is a security feature built into that hides password.
String entered in 1st part login prompt is – Login name or user-id or username
In 2nd part password prompt string entered by user is known as password.

Command Structure

UNIX commands take the following general form:

verb [options] [arguments]

where verb is the command name that can take a set of optional options and one or more optional arguments.
Commands, options and arguments have to be separated by spaces or tabs to enable the shell to interpret them as words. A contiguous string of spaces and tabs together is called a whitespace. The shell compresses multiple occurrences of whitespace into a single whitespace.
Unix arguments range from simple to complex. They consists of options, expressions, instructions, file names etc.

Options

An option is preceded by a minus sign (-) to distinguish it from filenames.
Example: $ ls –l
There must not be any whitespaces between – and l. Options are also arguments, but given a special name because they are predetermined. Options can be normally compined with only one – sign. i.e., instead of using

$ ls –l –a –t

we can as well use,

$ ls –lat

Because UNIX was developed by people who had their own ideas as to what options should look like, there will be variations in the options. Some commands use + as an option prefix instead of -.

Filename Arguments

Many UNIX commands use a filename as argument so that the command can take input from the file. If a command uses a filename as argument, it will usually be the last argument, after all options.
Example: cp file1 file2 file3 dest_dir
rm file1 file2 file3
The command with its options and argumens is known as the command line, which is considered as complete after *[Enter]* key is pressed, so that the entire line is fed to the shell as its input for interpretation and execution.

Exceptions

Some commands in UNIX like pwd do not take any options and arguments. Some commands like who may or may not be specified with arguments. The ls command can run without

arguments (ls), with only options (ls –l), with only filenames (ls f1 f2), or using a combination of both (ls –l f1 f2). Some commands compulsorily take options (cut). Some commands like grep, sed can take an expression as an argument, or a set of instructions as argument.

UNIX provides flexibility in using the commands.

Entering a Command before previous command has finished

You need not have to wait for the previous command to finish before you can enter the next command. Subsequent commands entered at the keyboard are stored in a buffer (a temporary storage in memory) that is maintained by the kernel for all keyboard input. The next command will be passed on to the shell for interpretation after the previous command has completed its execution.

Basic Commands

echo, printf, ls, who, date, passwd, cal

echo command:(Displaying a message)

- It is an internal command – when you type echo shell won't look in its PATH to locate it. It will execute it from its own set of built in. It can be checked as follows

  $type echo
  echo is a shell builtin

- This message is often used in shell scripts to display diagnostic message on the terminal or to issue prompts for taking user inputs

Ex: 1.
$echo hello
hello
$_
Ex 2.
$echo "enter filename: \c":
Enter filename: $_

- An escape sequence is generally a two character string beginning with a \(backslash)
- Bash interpret the escape sequence only when echo is used with **the –e** option

Escape sequences

- \a Bell
- \b backspace
- \c no new line
- \f formfeed
- \n newline
- \r carriage return

- \t tab
- \v vertical tab
- \\ backslash
-

printf command : An alternate to echo
It exists as an external command but only in bash shell it is built in
Usage
$printf " Good Morning\n"
Good Morning
$_
It also accepts escape sequence and formatted strings
$printf" My Shell is %s\n" $SHELL
My Shell is /usr/bin/bash

- $_

Formats
%s string
%30s print in a space 30 characters
%d decimal integer
%6d print in a space 6 characters
%o octal integer
%x hexadecimal integer
%f floating point number
Ex.

$ printf "the value of 255 is %o in octal and %x in hexadecimal \n" 255 255
the value of 255 is 377 in octal and FF in hexadecimal

- Format strings can convert data from one form to another
- The %o and %x format strings are also used by awk and perl to convert a decimal integer to octal and hex, respectively

ls command : listing directory contents

- Use to obtain a list of all filename in the current directory

Syntax:
ls [options] [argument]

- Ex: ls

ls options
-a Lists all files, including those beginning with a dot (.).
-d Lists only names of directories, not the files in the directory
-F Indicates type of entry with a trailing symbol: executables with *, directories with / and symbolic

links with @
-R Recursive list
- r sorts filesname in reverse order
-u Sorts filenames by last access time
-t Sorts filenames by last modification time
-i Displays inode number
-l Long listing: lists the mode, link information, owner,
size, last modification (time).
-x output in multiple columns
-d dirname

who command : who are the users?

- Unix maintains an account of all users who are logged on to the system

Who command displays an informative listing of there users

- $ who

root console **aug 1 07:51** (:0)
kumarpts/10 **aug 1 07:56** (pc123.heavens.com)
Sharmapts/6 **aug 1 02:10** (pc123.heavens.com)

- User name or userid's
- Device name of their there respectively terminals
- Logging time
- Machine name from where the user logged in
- Who
  -H display with header information
  -u prints coloumn header
- Who am i

kumarpts/10 **aug 1 07:56** (pc123.heavens.com)

date command: Displaying the system date

- The unix system maintains an internal clock. When the system is shut down, a battery backup keeps the clock ticking
- You can display the current date with the date command
- $date

Tue Aug 16 10:30:40 IST 2016

- $date +%m
  08
- $date +%h
  Aug
- $date +"%h%m"

Aug 08

Options

- d - day of the month
- y - last two digit of the year
- H, M, S - hour, min and sec
- D- date in mm/dd/yy format
- T- time in format hh:mm:ss

Note: When you use multiple format specifiers, you must enclose them within quotes and use a single + symbol before it

passwd command- changing your password

- $passwd

passwd: changing password for kumar
Enter login password:********
New password:*******
Re-enter New passowrd:*******
passwd(SYSTEM): Password successfully
changed for kumar

- Encryption is stored in a file named shadow in the /etc directory

cal command – The calendar

- cal can be used to see the calendar of any specific month or a complete year
- Syntax

cal [[month] year]
Every thing in rectangular bracket are optional
cal can be used without arguments
Ex 1.
$ cal
Aug 2016
montue…

Ex 2.

$ cal 03 2016
Ctrl-s to make it pause

Combining Commands

Instead of executing commands on separate lines, where each command is processed and executed before the next could be entered, UNIX allows you to specify more than one command in the single command line. Each command has to be separated from the other by a ; (semicolon).

wc sample.txt ; ls –l sample.txt

You can even group several commands together so that their combined output is redirected to a file.

(wc sample.txt ; ls –l sample.txt) >newfile

When a command line contains a semicolon, the shell understands that the command on each side of it needs to be processed separately. Here ; is known as a metacharacter.

Note: When a command overflows into the next line or needs to be split into multiple lines, just press enter, so that the secondary prompt (normally >) is displayed and you can enter the remaining part of the command on the next line.

Meaning of Internal and External Commands

Some commands are implemented as part of the shell itself rather than separate executable files. Such commands that are built-in are called internal commands. If a command exists both as an internal command of the shell as well as an external one (in /bin or /usr/bin), the shell will accord top priority to its own internal command with the same name. Some built-in commands are echo, pwd, etc.

type command:knowing the type of command and locating it

Ex1.

$type cat

cat is /bin/ls

Ex 2.

$type echo

echo is a shell builtin

Root login

The system administrator also known as superuser or root user.

The job of system administration involves the management of the entire system- ranging from maintaining user accounts, security and managing disk space to performing backups.

Root: The system administrator's login

The unix system provides a special login name for the exclusive use of the administrator; it is called root.

This account doesn't need to be separately created but comes with every system.

Its password is generally set at the time of installation of the system and has to be used on logging in:

login: root

Password: ********* [Enter]

-

The prompt of root is #.

Once you login as a root you are placed in root's home directory.

Depending on the system, this directory could be / or /root.

Administrative commands are resident in /sbin and /usr/sbin in modern systems and in older system it resides in /etc.

Roots PATH list includes detailed path

, for example: /sbin:/bin:/usr/sbin:/usr/bin:/usr/dt/bin

Becoming the super user,

su command. :

su: Acquiring superuser status Any user can acquire superuser status with the su command if they know the root password.

For example, the user abc becomes a superuser in this way.

$ su

Password: **********

#pwd /home/abc
Though the current directory doesn't change,
the # prompt indicates that abc now has powers of a superuser. To be in root's home directory on superuser login, use su –l.