GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING (SOFTWARE ENGINEERING) (02)

ADVANCE OPERATING SYSTEM

SUBJECT CODE: 2720213

SEMESTER: II

Type of course: Regular

Prerequisite:

Data Structures and Algorithms,
Advanced Programming (or good working knowledge of C), and
Fundamentals of Computer Systems
Fundamentals of Operating System

Rationale:

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks		Practical Marks				
				ESE (E)	PA (M)	PA (V)		PA (I)		
						ESE	OEP	PA	RP	
3	2#	0	4	70	30	30	0	10	10	150

Content:

Sr. No.	Content	Total Hrs	% Weightage
1	**Introduction to UNIX/Linux kernel** System Structure, Architecture of UNIX operating system	1	3
2	**Files & Directories IO** Function : open- creat- close- lseek-read- write, file sharing, atomic operations, dup and dup2 function , fcntl function , ioctl function, /dev/fd stat, fstat & lstat function, file types, Set-User-ID and Set-Group-ID, file access permissions, ownership of new files and directories, access function, umask function, chmod and fchmod function, sticky bit, chown, fchown, and lchown function, file size, file truncation, file systems, link, unlink, remove, and rename functions, symbolic links, symlink and readlink functions, file times, utime, mkdir and rmdir, reading directories, chdir- fchdir and getcwd funciton, device special files. **Advanced File IO** Scatter/Gather I/O, The Event Poll Interface, Mapping Files into Memory, Advice for Normal File I/O, Synchronized, Synchronous, and Asynchronous Operations, I/O Schedulers and I/O Performance	4	12
3	**Process Environment:**	5	15

	Process termination, environment list, memory layout of a C program, shared libraries, memory allocation, environment variables, setjmp and longjmp, getrlimit and setrlimit **Process Control:** process identifiers, fork, vfork, exit, wait and waitpid, waitid, wait3 and wait4, race conditions, exec, changing user IDs and group IDs, interpreter files, system function, process accounting, user identification, process times **Process Relationship:** Terminal logins, network logins, process groups, sessions, controlling terminal, tcgetpgrp, tcsetpgrp, and tcgetsid functions, job control, shell execution of programs, orphaned process groups		
4	**Memory Management** The Process Address Space, Allocating Dynamic Memory, Managing Data Segment, Anonymous Memory Mappings, Advanced Memory Allocation, Debugging Memory Allocations, Stack-Based Allocations, Choosing a Memory Allocation Mechanism, Manipulating Memory, Locking Memory, Opportunistic Allocation	4	11
5	**Signal Handling** Signal Concepts, Basic Signal Management, Sending a Signal, Re-entrancy, Signal Sets, Blocking Signals, Advanced Signal Management, Sending a Signal with a Payload	4	11
6	**Elementary TCP Sockets** Socket introduction:socket address structure ,value –result argument ,Byte Manipulation Functions,inet_aton,inet_addr and inet_ntoa Functions ,sock_ntop and Related Functions ,readn,writen and readline Functions TCP Socket: socket Function,connect Function,bind Function,listen Function,accept Function,fork and exec Function, Concurrent Servers,close Function, getsockname and getpername Functions	3	8
7	**TCP Client/server Example : .** TCP Echo Server:main Function,str_echo function TCP Echo Client:main function, str_cli function, Normal Startup & termination,termination of server Process ,Crashing of server host, Crashing and Rebooting of Server Host, shut down of server host	3	8
8	**Socket options** getsockopt and setsocket Function,checking if an Option is supported and obtaining the Default,Socket States, Generic Socket Options,IPV4 Socket options , ICMPV6 Socket Option, IPV6 Socket Options, TCP Socket options,SCTP Socket Options,fcntl Functions	3	8
9	**Elementary UDP Sockets** recvfrom and sendto Functions,UDP Echo server: main function, dg_echo Function, Echo client : main function, dg_cli function Lost Datagrams, connect function with UDP, lack of Flow Control with UDP, Determining outgoing interface with UDP, TCP & Echo server using select	3	8

10	**Name and Address Conversions** Domain Name system, gethostbyname Function, gethosybyaddr Function, getservbyname ad getservbyport Function, getaddrinfor Function,getadrinfo function, gai_strerror Function, freeaddrinfo Function getadrinfo function: IPV6,getaddrinfo function,host_serv Function, tcp_connect function, tcp_listen function, udp_client Function, udp_connect function, udp_server function,getname info function,re-entrant function , gethostbyname_r and gethostbyaddr_rfunctions, Obsolete IPV6 Address Lookup Functions	6	16

Reference Books:

1. Advanced Programming in the UNIX Environment, Addison-Wesley, by Richard Stevens.
2. Linux System Programming, O'Reilly, by Robert Love.
3. Unix Network Programming, Volume 1: The Sockets Networking API (3rd Edition)
4. The Design of the UNIX Operating System, PHI, by Maurice J. Bach.
5. The Definitive Guide to Linux Network Programming by Kathryn Davis Apress publication

Course Outcome:

After learning the course the students should be able to:

- Discuss the various synchronization, scheduling and memory management issues.
- Do advanced operating system programming.
- Manage files, directories, and processes, carefully and laying the groundwork for more advanced techniques, such as signal handling and terminal I/O.
- Do socket programming in linux/unix environment.

List of tutorials:

1. Write your own dup2 function that performs the same service as the dup2 system call without calling the fcntl function. Be sure to handle errors correctly.
2. Write a utility like cp(1) that copies a file containing holes, without writing the bytes of 0 to the output file.
3. Write a C program that creates a zombie, and then call systemto execute the ps(1) command to verify that the process is a zombie.
4. Implement your own sig2str function.
5. Write a C program that creates a file and writes the integer 0 to the file. The process then creates a child, and the Parent and Child alternate incrementing the counter in the file. Each time the counter is incremented, print which process (Parent or Child) is doing the increment.
6. Write a C program that calls fork and has the child create a new session. Verify that the child becomes a process group leader and that the child no longerhas a controlling terminal.
7. Write a C function which handles all possible signals. The function should consist of a single loop that iterates once for every signal in the current signal mask (not once for every possible signal).
8. Write a C program that calls sleep (60) in an infinite loop. Every five times through the loop (every 5 minutes), fetch the current time of day and print the tm_sec field.
9. Write a C program that calls fwrite with a large buffer (a few hundred megabytes). Before

calling fwrite, call alarm to schedule a signal in 1 second. In your signal handler, print that the signal was caught and return.

10. Write Unix domain program to use the sockets interface to establish a connection and transmit data between client-server using TCP.
11. Write Unix domain program to use the sockets interface to establish a connection and transmit data between client server using UDP.
12. The problem is to implement a client - server user-level application using sockets API in C/C++. The Server application has to support at least five clients simultaneously. Server accepts strings from clients (even multiple strings from each client) and replies with reverse strings. For example, when client sends "IITHYD", Server replies with "DYHTII". Both server and client(s) have to output both sending & receiving strings on the terminal. The server and client processes should be run on different machines

Major Equipment:

i Linux based Host machines (Free & Open Source Software or Open source)
ii Computers with latest hardware configuration

List of Open Source Software/learning website:

- Operating System concepts: http://nptel.iitm.ac.in/
- Linux basics: www.freeos.com/guides/lsst
- Linux basics: www.linuxcommand.org/writing_asell_scripts.
- Linux basics: www.distro.ibiblio.org/damnsmall/current/dsl-4.4.10- embedded.zip

Review Presentation (RP): The concerned faculty member shall provide the list of peer reviewed Journals and Tier-I and Tier-II Conferences relating to the subject (or relating to the area of thesis for seminar) to the students in the beginning of the semester. The same list will be uploaded on GTU website during the first two weeks of the start of the semester. Every student or a group of students shall critically study 2 papers, integrate the details and make presentation in the last two weeks of the semester. The GTU marks entry portal will allow entry of marks only after uploading of the best 3 presentations. A unique id number will be generated only after uploading the presentations. Thereafter the entry of marks will be allowed. The best 3 presentations of each college will be uploaded on GTU website.