19-202-0610 OPERATING SYSTEM LABORATORY

Course Outcomes:

On completion of this course the student will be able to:

1. *Develop shell scripts.*
2. *Implement scheduling algorithms.*
3. *Write programs using system calls.*
4. *Write programs to implement inter process communication.*
5. *Write system level programs.*

Cycle-I

1. Study of different system calls.
2. Programs using the system calls of linux operating system- fork,exec,getpid,exit,wait,close,stat,opendir,readdir.
3. Programs using the I/O system calls of Linux operating system.
4. Programs to simulate Linux commands like ls,grep etc

Cycle- II

1. Programs to study and analyse various scheduling policies.
2. Programs to study uses of semaphore.
3. Programs to implement page replacement algorithms.

Cycle-III

1. Programs to implement IPC using shared memory, pipes, and message queue.
2. Linux shell programming.
3. Kernel programming--Linux Kernel configuration, compilation and rebooting from the newly compiled kernel.
4. Kernel space programming: Implement and add a loadable kernel module to Linuxkernel, demonstrate using insmod, lsmod and rmmod commands.
5. Developing device drivers.
6. Creating Linux distributions from debian source.

References:

1. Richard Stevens, W., UNIX Network Programming: Interprocess communications, Volume 2,Second Edition, Prentice Hall, ISBN: 9780130810816.
2. Peter Jay Salzman, Michael Burian, Ori Pomerantz, The Linux Kernel Module Programming Guide. http://www.tldp.org/LDP/lkmpg/2.6/lkmpg.pdf.
3. Robert Love, Linux Kernel Development, 3rd Edition Addison-Wesley Professional, ISBN: 978- 0672329463.
4. Mark G. Sobell, Practical Guide to Linux Commands, Editors, and Shell Programming, 3rd Edition, Prentice Hall, ISBN-13: 978-033085044.