BCA (HONS) 3rd SEMESTER
DISCIPLINE SPECIFIC COURSE (CORE)

BCA320C2: OPERATING SYSTEMS

CREDITS: THEORY: 4; PRACTICAL: 2
MAX. MARKS: THEORY: 60; PRACTICAL: 30
MIN. MARKS: THEORY: 24; PRACTICAL: 12

UNIT-I

1. Introduction **(10 Lectures)**

Basic OS functions, resource abstraction, types of operating systems-multiprogramming systems, batch systems, time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

2. Operating System Organization **(5 Lectures)**

Processor and user modes, kernels, system calls and system programs.

UNIT-II

3. Process Management **(15Lectures)**

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter-process communication; deadlocks.

UNIT-III

4. Memory Management **(15 Lectures)**

Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory

UNIT-IV

5. File and I/O Management **(10 Lectures)**

Directory structure, file operations, file allocation methods, device management.

6. Protection and Security **(5 Lectures)**

Policy mechanism, Authentication, Internal access Authorization.

Recommended Books:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8thEdition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modem Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modem Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles, 5th Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

LAB: OPERATING SYSTEMS **CREDITS: 2 LAB; 60 LECTURES** **C/ C++ PROGRAMS**

1. WRITE A PROGRAM (using *fork()* and/or *exec()* commands) where parent and child execute:
   a) same program, same code.
   b) same program, different code.
   c) before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behaviour of Linux kernel including information on configured memory, amount of free and used memory, (memory information)
4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
5. WRITE A PROGRAM to copy files using system calls.
6. Write program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.
11. Write program to implement SRJF scheduling algorithm.
12. Write program to calculate sum of n numbers using *thread* library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.