Course Code	Course Title				Category
18BTIS502	System Programming & Operating Systems				core
Contact Hours per Week			CA	FE	Credits
L	T	D/P			
3	0	0	40	60	3

Prerequisite:

- Good knowledge of C programming
- Basic Data Structure concepts

Course Objectives:

- To understand basics of System Programming.
- To learn and understand data structures used in design of system software.
- To understand functions of operating system.
- To learn and understand process, resource and memory management

INTRODUCTION TO SYSTEMS PROGRAMMING (09 hrs)

Need of Systems programming, Fundamentals of Language Processing, Language Processing Activities, Components of System Software, Introduction to Assemblers, Elements of Assembly Language Programming, Simple Assembler scheme, Structure of an Assembler, Design of Two Pass Assembler, Single pass assembler.

INTRODUCTION TO MACRO PROCESSORS (09 hrs)

Introduction to Macro Processors, Macro Definition and Call, Macro expansion, Nested Macro Calls and definition, Advanced Macro Facilities. Loaders: Definition of Loader. Different Loader schemes: General loader Scheme, Absolute loaders, subroutine linkages, relocating loaders, direct linking loaders, overlay structure. Linkers: relocation and linking concepts, Self-relocating programs, an introduction to static and dynamic linked libraries.

INTRODUCTION TO OPERATING SYSTEM & PROCESS MANAGEMENT (09 hrs)

Process: Concept of a Process, Process States, Process Description, Process Control (Process Creation, Waiting for the Process/Processes, Loading Programs into Processes and Process Termination), **Threads**: Processes and Threads, Concept of Multithreading, Types of Threads, Thread programming Using Pthreads. Scheduling: Uniprocessor **Scheduling** - Types of Scheduling, Scheduling Algorithms (Ex. FCFS, Round Robin Scheduling and SJF), Principles of Concurrency, Requirements for Mutual Exclusion,

Mutual Exclusion: Operating System Support (Semaphores and Mutex), Example: Readers/Writers Problem,

Deadlock and Starvation: Principles of Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, An Integrated Deadlock Strategy, Example: Dining Philosophers Problem. Case Study: Linux OS

MEMORY MANAGEMENT **(09 hrs)**

Memory Management: Memory Management Requirements, Memory Partitioning: Fixed Partitioning, Dynamic Partitioning, Buddy System, Relocation, Paging and Segmentation. Virtual Memory: Principle of Locality, Demand Paging, Page Replacement Algorithms (Ex. FIFO, Optimal, LRU and Clock), and Thrashing: Dealing with Thrashing. Case Study: Linux.

I/O AND FILE MANAGEMENT **(09 hrs)**

I/O Management and Disk Scheduling: I/O Devices, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling (Ex. FCFS, SSTF & SCAN), Disk Cache. File Management: Overview, File Organization and Access, File Directories, File Allocation Methods and Free Space Management. Case Study: Linux

Course Outcomes: (Students will be able to-)

- Analyze and synthesize system software
- Understand the working of Assemblers and Macroprocessors
- Understand the functions of operating system.
- Do the programming for process and thread management in OS

TEXT BOOKS

1. Systems Programming and Operating Systems, Dhamdhere D.M, TMGH
2. Modern Operating System – Andrew S. Tanenbaum, Pearson Education India
3. Operating System Concepts - Abraham Silberschatz, Peter B. Galvin & Grege Gagne, Wiley

REFERENCES

1. Operating System : Concepts and Design - Milan Milenkovic, TMGH
2. Understanding the Linux kernel - Daniel P Bovet and Marco Cesati, O'Reilly
3. Linux System Programming - Robert Love, Publisher - SPD, O' Reilly
4. Systems programming - John J. Donovan, TMGH