System Programming and Operating Systems(304185)

Teaching Scheme:
Lectures: 3 Hrs/ Week

Examination Scheme:
In Semester Assessment:
Phase I : 30
End Semester Examination:
Phase II: 70

Course Objectives:

- To understand fundamentals of system programming and operating systems.
- To study and understand how the system programming and operating system abstractions can be implemented.
- To develop comprehensive skills to design Assembler, Macro Processor, Compiler and Interpreters.
- To understand the importance of application of linkers, loaders and Software tools in system programming
- To Implement System Programming concepts and Operating systems components
- To analyze memory allocation methods, input output devices and file system w. r. t. various operating system.
- To study and implement various process scheduling techniques and dead lock avoidance schemes in operating system

Course Outcomes:

After successfully completing the course students will be able to

- Demonstrate the knowledge of Systems Programming and Operating Systems
- Formulate the Problem and develop the solution for same.
- Compare and analyze the different implementation approach of system programming and operating system abstractions.
- Interpret various OS functions used in Linux / Ubuntu

Unit I : Basics of System Programming 7L

Language processors: Language processing activities, Fundamentals of language processing, Fundamentals of language specification, Language processor development tools. Data structures for language processing: Search data structure, Allocation data structures. Scanning and parsing Assembler: Assembly language programming, simple assembly scheme, Pass structure of assembler, design of two pass assembler

Unit II: Macro Processor, Compilers and Interpreters 7L

Macro Processor: Macro definition and call, macro expansion, Machine Independent macro processor features, Nested macro calls, advanced macro facilities, Design of macro pre processor. Compilers: Basic compilers function, Phases of compilation, memory allocation, compilation of expression, Compilation of expressions, compilation of control structures, Code of optimization Interpreters.

Unit III: Linkers and Loaders and Software Tool 6L

Linkers and Loaders: Basic loaders functions, central loaders scheme, absolute loaders, Subroutine linkers, relocation loader, Direct linking loader, Dynamic linking loader, Design of absolute loaders and direct linking loader, Software tools: Software tools for program development, editors, debug monitor, programming environment, user interfaces.

Unit IV : Introduction to OS, Process Management and Deadlocks 8L

Operating System: Evolution of OS, OS Functions, Various OS, OS structure, OS System Calls with example. Process Management: Processes, Inter process communication, Classical IPC problems, Threads, CPU Scheduling. Deadlocks: System Model, Deadlock Characterization, Deadlock Prevention, Deadlock Avoidance, Deadlock detection and recovery.

Unit V : Memory Management 6L

Basics of memory management, Swapping, Memory Allocation, Paging, Segmentation Virtual memory, Demand Paging, Page replacement, Page replacement algorithms – Optimal FIFO, LRU, LRU approximation, Allocation of frames

Unit VI : Input and Output, File system 6L

Input and Output: Review of computer hardware, principles of I/O hardware, and principles of I/O software, I/O software layers, disks, disk scheduling Algorithms. File System w.r.t. Linux: Files, directories, file system and implementation, File system layout, implementing files, implementing directories, shared files, disc space management

Text Books

1. D. M. Dhamdhare, “Systems Programming and Operating System”, TMH.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Second Edition, PHI.

Reference Books

1. J. J. Donovan, “Systems Programming”, McGraw Hill.
2. Siberschatz A; Galvin P.B; Gagne G, “Operating System Concepts”, John Wiley.
3. Leland L. Beck, “System Software,” Pearson Editions.