FIFTH SEMESTER

CODE	COURSE TITLE	L	T	P	C
CS301	Systems Programming	3	0	0	3
CS303	Operating Systems	3	0	0	3
CS305	Microprocessors And Microcontrollers	3	0	0	3
CS307	Software Engineering	3	0	0	3
CS309	Graph Theory	3	1	0	4
CS311	Advanced Computer Architecture	3	1	0	4
CS315	Microprocessors And Microcontrollers Laboratory	0	0	3	2
CS313	Operating Systems Laboratory	0	0	3	2
TOTAL CREDITS 24					

FIFTH SEMESTER

CS301-SYSTEMS PROGRAMMING

Credits: 3

Objectives

- To introduce the major programming paradigms, data structures and principles involved in systems programming
- To acquire comprehensive knowledge about various system components and its functionalities as well as the interactions with hardware resources
- To provide basic insight about writing system programs for each system components
- To gain knowledge about developing interfaces for various system components and its issues

Unit-I Fundamentals of language processors

Language specification - Data structure for language processing - Scanning - Parsing.

Unit-II Assemblers

Elements of assembly language programming - Single pass and two pass assembler - Assembler for IBM PC.

Unit-III Macro Processors

Macro definition and call - macro expansion - conditional and nested macro calls - design of a macro processor.

Unit-IV Loaders

Relocation and linking concepts - Relocating programs - Design of a linker - Linking for overlays - A linker for MSDOS.

Unit-V Linkers

Software tools - Text editor - Debug monitors - Interpreters - Program generators - User interfaces - Recent trends and developments

Outcomes

- Obtain deep knowledge about basic systems programming paradigms

Knowhow on the importance and design principles of various system component

Text Books

1. Leland L.Beck,“An introduction to systems programming”, 4th edition, addision wesley 2001.
2. D.M. Dhamdhere,”Systems programming and operating systems”, 3rd edition, Tata Mcgraw Hill, 2002.
3. J.J. Donovan,”Systems programming”, Mcgraw Hill, 1984.

CS303 – OPERATING SYSTEMS

Credits: 3

Objectives

- To know the basics such as process and CPU scheduling algorithms
- To understand the critical regions and dead lock problem
- To understand virtual memory concept, thrashing problem and page replacement algorithms
- To understand the file tables, access algorithms, and spoofing

Unit-I Basic OS Concepts

User's view of the OS - Architectural support - Thread and process scheduling - Preemptive and non-preemptive - FCFS, SJF, Round Robin, Multilevel Queue.

Unit-II Synchronization

Peterson's solution - Bakery algorithm - Hardware-based solutions - Semaphores - Critical regions - Problems of synchronization - Deadlock prevention and recovery - Banker's algorithms.

Unit-III Memory Management

Swapping – Contiguous Memory Allocation – Paging – Segmentation – Segmentation with Paging – Virtual Memory – Demand Paging – Process Creation – Page Replacement – Allocation of Frames – Thrashing.

Unit-IV File Systems

File Concept – Access Methods – Directory Structure – File System Mounting – File Sharing – Protection – File System Structure – File System Implementation – Directory Implementation – Allocation Methods – Free Space Management.

Unit-V I/O Systems

Kernel I/O Subsystem – Disk Structure – Disk Scheduling – Disk Management – Swap Space Management – RAID Structure – Case study on Linux System – Case study on Windows XP.

Outcomes

- Ability to implement CPU scheduling algorithms and resolve problems related to critical regions
- Ability to implement page replacement algorithms like FCFS, LRU, etc.
- Ability to change UNIX access controls to protect the files

Teaching and Evaluation guidelines

- 30% on an Application (Higher Order Thinking), and 50% on Methods and Algorithms (Medium Order Thinking), and 20% on Definition (Lower Order Thinking).

Text Books

1. A.Silberchatz, P.B.Galvin, "Operating System Concepts", Addison Wesley, IX Edition, 2013
2. W.Stallings, "Operating Systems", Prentice Hall, V Edition, 2005

CS305 - MICROPROCESSORS AND MICROCONTROLLERS

Credits: 3

Objective

- To understand the concepts of Architecture of 8086 microprocessor
- To understand the design aspects of I/O and Memory Interfacing circuits
- To understand the architecture and programming of ARM processor

Unit-I The 8086 Microprocessor

Introduction to 8086 – Microprocessor architecture – Addressing modes - Instruction set and assembler directives – Assembly language programming – Modular Programming - Linking and Relocation - Stacks - Procedures – Macros – Interrupts and interrupt service routines – Byte and String Manipulation.

Unit-II 8086 System Bus Structure

8086 signals – Basic configurations – System bus timing –System design using 8086 – IO programming – Introduction to Multiprogramming – System Bus Structure – Multiprocessor configurations – Coprocessor, Closely coupled and loosely Coupled configurations – Introduction to advanced processors.

Unit-III Microcontroller

Architecture of 8051 – Special Function Registers(SFRs) - I/O Pins Ports and Circuits - Instruction set - Addressing modes - Programming 8051 Timers – Interfacing Microcontroller - Serial Port Programming - Interrupts Programming – LCD & Keyboard - External Memory Interface- Stepper Motor.

Unit-IV Introduction to Embedded Systems

Complex systems and micro processors– Embedded system design process – Instruction sets preliminaries - ARM Processor – CPU: programming input and output supervisormode, exceptions and traps – Co-processors- Memory system mechanisms – CPU performance

Unit-V Embedded Computing Platform Design and Optimization

The CPU Bus-Memory devices and systems–Designing with computing platforms – platform level performance analysis - Components for embedded programs-Models of programs-Assembly, linking and loading – compilation techniques- Program level performance analysis – Software performance optimization – Analysis and optimization of program size- Program validation and testing.

Outcomes

- Ability to design and implement programs on 8086 microprocessor
- Ability to design I/O circuits and Memory Interfacing circuits
- Ability to design and develop components of ARM processor

Teaching and Evaluation guidelines

- 30% on An Application (Higher Order Thinking), and 50% on Methods and Techniques for interfacing (Medium Order Thinking), and 20% on Assembly level of programming (Lower Order Thinking).

Text Books

1. Yu-Cheng Liu, Glenn A.Gibson, “Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design”, Second Edition, Prentice Hall of India, 2007
2. Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, “The 8051 Microcontroller and Embedded Systems: Using Assembly and C”, 2nd Edition, Pearson Education, 2011
3. Marilyn Wolf, “Computers as Components - Principles of Embedded Computing System Design”, 3rd Edition “Morgan Kaufmann Publisher (An imprint from Elsevier), 2012

References Books

1. Doughlas V. Hall, “Microprocessors and Interfacing, Programming and Hardware”, Tata McGraw-Hill, 2012
2. Jonathan W. Valvano, “Embedded Microcomputer Systems Real Time Interfacing”, 3rd Edition, Cengage Learning, 2012
3. David. E. Simon, “An Embedded Software Primer”, 1st Edition, Fifth Impression, Addison-Wesley Professional, 2007

CS307 - SOFTWARE ENGINEERING

Credits: 3

Objectives

- To understand the importance of software engineering lifecycle models in the development of software
- To understand the various design principles in modelling a software
- To develop a software which adheres to the standard benchmarks
- To undergo the technical know in the process of software testing

Unit I - Software Process

Introduction – S/W Engineering Paradigm – life cycle models (waterfall, incremental, spiral, WINWIN spiral, evolutionary, prototyping) – Object Oriented life cycle models-system engineering – computer based system – life cycle process – development process.

Unit II - Software Requirements

Functional & non-functional – user-system requirement engineering process – feasibility studies – elicitation – validation & management – software prototyping – S/W documentation – Analysis and modeling – Case Tools (Rational Rose).

Unit III - Design Concepts and Principles

Modular design – design heuristic – S/W architecture – data design – architectural design – transform & transaction mapping –Introduction to SCM process – Software Configuration Items.

Unit IV - Software Testing

Taxonomy of S/W testing – levels - black box testing – testing boundary conditions – structural testing — regression testing– S/W testing strategies – unit testing – integration testing – validation testing – system testing and debugging.

Unit V - Software Project Management

S/W cost estimation - Function point models – COCOMO model –Project Scheduling- Delphi method – S/W challenges – S/W maintenance-Reliability – Reliability and availability models.

Outcomes

- Ability to show the enhance the Software Project Management skills
- Ability to develop a functioning software which benchmarks to the international standards

Teaching and Evaluation guidelines:

- 50% on Methods and Techniques (Medium Order Thinking), 30 % on Application (Higher Order Thinking), and 20% on Tool functions (Lower Order Thinking).

Text Books

1. R.S.PRESSMAN, "Software Engineering - A practitioners approach", VII Edition, McGraw Hill International editions, 2014.
2. STEPHEN R. SCHACH, "Object oriented and classical software Engineering", VIII Ed., McGraw Hill, 2010.

Reference Books

1. Pfleeger and Lawrence “Software Engineering: Theory and Practice”, Pearson Education, fourth edition, 2012.
2. Pankaj Jalote, “An Integrated Approach to Software Engineering”, Narosa, Third edition, 2013.

CS309 - GRAPH THEORY

Credits: 4

Objectives

- Understand the importance of graph theory with respect to computer science applications and application of the basic corollaries and theorems learnt.

Unit-I Introduction

Basic Definition – Graphs, Digraphs, adjacency matrix, Incidence Matrix, basic theorems on simple graphs.

Unit-II Trees and Graphs

Graphs: Graphs, Isomorphism, trees, spanning trees, binary trees, planar graphs, Euler circuits, Hamiltonian graphs, Chromatic numbers, Four-colour problem.

Unit-III Cut-sets and Cut-vertices

Trees and fundamental circuits - Cut-sets and Cut-vertices - Connectivity and Separability - Network flows - 1 and 2 isomorphism.

Unit-IV Planar Graphs

Planar and Dual Graphs - Kuratowski's graphs - Representations of a planar graph - Vector space associated with a graph - Subspaces - Orthogonal vectors and spaces.

Unit-V Representation of Graphs

Matrix Representation of Graphs - Circuit matrix - Cutset matrix - Path matrix - Adjacency matrix - Algorithms for fundamental circuits, cut-vertices and separability.

Outcomes

- To be able to use the concepts learnt in graph theory in designing algorithms for real world applications.

Teaching and Evaluation Guidelines

- 40% on Analysis and Application (Higher Order Thinking), and 60% on Conceptual understanding and Definitions (Lower Order Thinking).

Text Books

1. Narsingh Deo, "Graph Theory With Applications To Engineering And Computer Science" PHI Eastern Economy Edition , 2009
2. E.S.Page and L.B.Wilson, "An introduction to computational combinatorics", Cambridge University Press, 1979
3. D.E.Knuth, O.Patashuk, R.L.Graham, "Concrete Mathematics", 1994
4. Douglas. B. West, "Introduction to Graph Theory", Second edition. Prentice Hall,2001

CS311 – ADVANCED COMPUTER ARCHITECTURE

Credits: 4

Objectives

- To understand the fundamental knowledge in architecture design, pipelined processor design, and their impacts on performance
- To understand the fundamental knowledge in memory hierarchy
- To assess the communication and the computing possibilities of parallel system architecture

Unit-I Parallel computer models

Flynn's classification - Parallel and vector computers - System, implicit and explicit parallelism - Multi-vector and SIMD computers - PRAM and VLSI models.

Unit-II Program and network properties

Data and control dependence - Hardware and software parallelism - Partitioning and scheduling - Interconnection architectures.

Unit-III Performance laws

Metrics and measures - Amdahl's law for fixed workload - Bounded speed-up model - Scalability analysis and approaches.

Unit-IV Symbolic Processors

CISC and RISC architectures - Super scalar processors and their features - Memory hierarchy. Linear Pipeline Processors - Basic considerations - Basics of non-linear pipeline processors – Instruction pipeline design – Arithmetic pipeline design – Superscalar and Super pipeline design.

Unit-V Parallel Computing

Concepts & terminology, Parallel computer memory architecture, Parallel programming models, designing parallel programs, parallel algorithms.

Outcomes

- Ability to understand parallelism both in terms of a single processor and multiple processors
- Ability to understand parallel hardware constructs

Teaching and Evaluation guidelines

- 30% on An Application (Higher Order Thinking), and 50% on diagrams and architecture (Medium Order Thinking), and 20% on Definition (Lower Order Thinking).

Text Book

1. K. Hwang, "Advanced Computer Architecture, Parallelism, Scalability, Programmability", McGraw Hill, New York, 2010.

Reference Book

1. D. A. Patterson and J. L. Hennessy, "Computer Architecture: A Quantitative Approach", Harcourt Asia, Morgan Kaufmann, 1999.

CS315 - MICROPROCESSORS AND MICROCONTROLLERS LABORATORY

Credits: 2

Objectives

- To understand and learn the assembly language programming of various microprocessor architectures.
- To obtain the practical training of interfacing the peripheral devices with the processor.
- To control the components of a microprocessor based system through the use of interrupts.
- To impart a practical knowledge on assembling PC hardware, installation and troubleshooting the Microprocessor and Microcontrollers.

Experiments

Experiment Using 8086 Microprocessor

- Study of 8086 Microprocessor Trainer Kit
- 16-bit Arithmetic Operations (Addition, Subtraction, Multiplication and Division)
- Code Conversions
- 8086 Interface

Experiments Using 8051 Microcontroller

- Arithmetic operations in 8051
- ADC & DAC Interfacing
- Stepper Motor and DC Motor Interface

Outcomes

- Ability to obtain knowledge to do programs in assembly language programming using the trainer kits
- Ability to utilize development kits effectively for the real time applications of various peripheral devices with the processor
- Ability to design interfacing devices with the microprocessor

CS313 –OPERATING SYSTEMS LABORATORY

Credits: 3

Objectives

- To understand and write program in Unix environment
- To design and implement the scheduling algorithms
- To design and implement advanced file system operations

EXPERIMENTS (Operating Systems)

1. Implementation of System Calls Using Fork(),Sleep (), Wait ()
2. Implementation of CPU Scheduling Algorithm
    a. First Come First Serve Scheduling
    b. Shortest Job First Scheduling
    c. Priority Scheduling
    d. Round Robin Scheduling
3. Designing a Command Shell in Java
4. Implementation of Producer-Consumer Problem Using Semaphore
5. Implementing Bakery Algorithm (Critical Section Problem)
6. Implementing Banker's Algorithm (Deadlock Avoidance)
7. Simulate Paging technique of Memory Management
8. Implementation of File Systems
    a. Basic File Operations
    b. File Operation – I
    c. File Operation – II
9. Implementing Page Replacement Algorithm
    a. First In First Out (FIFO Page Replacement)
    b. Least Recently Used (LRU Page Replacement)
    c. Optimal Page Replacement

Outcomes

- Familiarize with the shell commands in Unix environment
- Ability to write system level programs