

## **CS010 601: Design And Analysis Of Algorithms** (Common with IT010 605)

### **Teaching scheme**

2 hours lecture and 2 hours tutorial per week

**Credits: 4**

### **Objectives**

- *To develop an understanding about basic algorithms and different problem solving strategies.*
- *To improve creativeness and the confidence to solve non-conventional problems and expertise for analysing existing solutions.*

### **Module I (13 hours)**

#### **Introduction and Complexity**

What is an algorithm – Properties of an Algorithm, Development of an algorithm, Pseudo-code Conventions, Recursive Algorithms – Performance Analysis - Space and Time Complexity –Asymptotic Notations – ‘Oh’, ‘Omega’, ‘Theta’, Worst, Best and Average Case Complexity, Running Time Comparison, Common Complexity Functions -Recurrence Relations – Solving Recurrences using Iteration and Recurrence Trees – Example Problems – Profiling - Amortized Complexity.

### **Module II (11 hours)**

**Divide and Conquer** - Control Abstraction, Finding Maximum and Minimum, Costs associated element comparisons and index comparisons, Binary Search, Divide and Conquer Matrix Multiplication, Strassen’s Matrix Multiplication, Quick Sort, Merge Sort. – Refinements.

### **Module III (14 hours)**

**Greedy Strategy** - Control Abstraction, General Knapsack Problem, Minimum Cost Spanning Trees – PRIM’s Algorithm, Kruskal’s Algorithm, Job sequencing with deadlines.

**Dynamic Programming** - Principle of Optimality, Multistage Graph Problem, Forward Approach, Backward Approach, All-Pairs Shortest Paths, Traveling Salesman Problem.

### **Module IV (11 hours)**

**Backtracking** – State Space Tree - Fixed Tuple and Variable Tuple Formulation - Control Abstraction – Generating Function and Bounding Function - Efficiency of the method - Monte Carlo Method – N-Queens Problem, Sum of Subsets.

**Branch and Bound Techniques** – FIFO, LIFO, and LC Control Abstractions, 15-puzzle.

### **Module V (11 hours)**

**Sophisticated Algorithms** - Approximation Algorithms – Planar Graph Coloring, Vertex cover - String Matching Algorithms – Rabin Karp algorithm - Topological Sort - Deterministic and Non-Deterministic Algorithms.

**Lower Bound Theory** - Comparison Trees for Searching and Sorting, lower bound on comparison based algorithms, Sorting, Selection & Merging; Oracles and Adversary Arguments –Merging, Basic concepts of randomized algorithm-Las Vegas algorithm for search.

### Reference Books

1. Ellis Horowitz and Sartaj Sahni, Sanguthevar Rajasekaran, *Fundamentals of Computer Algorithms*, Universities Press, 2<sup>nd</sup> Edition, Hyderabad .
2. Thomas Cormen, Charles, Ronald Rives, *Introduction to algorithm*, PHI Learning
3. Sara Baase & Allen Van Gelder , *Computer Algorithms – Introduction to Design and Analysis*, Pearson Education..
4. Anany Levitin, *Introduction to The Design & Analysis of Algorithms*, Pearson Education, 2<sup>nd</sup> Edition, New Delhi, 2008.
5. Berman and Paul, *Algorithms*, Cenage Learning India Edition, New Delhi, 2008.
6. S.K.Basu , *Design Methods And Analysis Of Algorithms* ,PHI Learning Private Limited, New Delhi,2008.
7. Jon Kleinberg and Eva Tardos, *Algorithm Design*, Pearson Education, New Delhi, 2006.
8. Hari Mohan Pandey, *Design Analysis And Algorithms*, University Science Press, 2008.
9. R. Panneerselvam, *Design and Analysis of Algorithms*, PHI Learning Private Limited, New Delhi, 2009.
10. Udit Agarwal, *Algorithms Design And Analysis*, Dhanapat Rai & Co, New Delhi, 2009.
11. Aho, Hopcroft and ullman, *The Design And Analysis of Computer Algorithms*, Pearson Education, New Delhi, 2007.
12. S.E.Goodman and S. T. Hedetmiemi, *Introduction To The Design And Analysis Of Algorithms*, McGraw-Hill International Editions, Singapore 2000.
13. Richard Neapolitan, Kumarss N, *Foundations of Algorithms*, DC Hearth &company.
14. Sanjay Dasgupta, Christos Papadimitriou, Umesh Vazirani, *Algorithms*, Tata McGraw-Hill Edition.

## CS010 602: Internet Computing

### Teaching scheme

2 hours lecture and 2 hours tutorial per week

**Credits: 4**

### Objectives

- *To impart the basic concepts of Internet Computing and Java Programming*
- *To develop understanding about Internet Computing with the help of Java Platform and establishing network connections using Socket Programming*

### Module I (10hours)

Introduction to Java- Genesis of Java- Features of Java –Data Types-Variables and Arrays-Operators- Control Statements – Selection Statements – Iteration Statements- Jump Statements.

### Module II (12 hours)

Creating & using classes in Java – Methods and Classes – Inheritance – Super Class – Method Overriding –Packages and Interfaces – Implementing Interfaces- Exception Handling – Exception Types, Threads-Multithreaded programs, Thread Priorities and Thread synchronization.

### Module III (14hours)

I/O – I/O Basics – Byte Streams and Character Streams, Reading Console Input, Collections Framework, Applets & Applet Architecture-Applet Skelton- Passing Parameters to Applet, Event Handling-Event Model- Event Classes – Event Listener Interfaces, AWT – AWT Classes – AWT Controls – Layout Managers and Menus. Swing- JApplet – Jbuttons - JTables.

### Module IV (13 hours)

Network Programming with Java – Socket Programming in Java-Client Sockets- Server Sockets- Secure Server Sockets- TCP/IP Programming with Java – Datagrams, IP multicasting, Remote Method Invocation.

### Module V (11 hours)

Advanced Java Programming – Accessing Databases with JDBC, Servlets, Image processing using Java – Image Filter – Web Application development using Java Technologies- Java Server Faces.

**Reference Books**

- 1) Herbert Schildt, *Java 2 Complete reference*, 5<sup>th</sup> ed., Tata McGraw Hill, New Delhi, 2010
- 2) Deitel & Deitel *Java How To Program* 7<sup>th</sup> ed., Pearson Education ,New Delhi, 2008
- 3) Cay Horstmann *Big Java* 3<sup>rd</sup> ed., Wiley India Edition, New Delhi, 2009
- 4) Y Daniel Liang *Introduction to Java Programming* 7<sup>th</sup> ed., Pearson Education ,New Delhi, 2010
- 5) R Krishnamoorthy, S Prabhu *Internet & Java Programming*, New Age International Publishers, New Delhi, 2008
- 6) Rajkumar Buyya, S Thamarai Selvi, Xingchen Chu, *Object Oriented Programming with Java*, McGraw Hill, New Delhi, 2009
- 7) P Radha Krishna, *Object Oriented Programming through Java* Universities Press, Hyderabad 2008
- 8) Debasish Jana, *Java and Object Oriented Programming Paradigm*, Prentice Hall of India, New Delhi, 2005
- 9) G Thomas Wu, *An Introduction to Object Oriented Programming with Java*, 4<sup>th</sup> ed., Tata McGraw Hill, New Delhi, 2010
- 10) E Balagurusamy, *Programming with Java A Primer*, 4<sup>th</sup> ed., McGraw Hill, New Delhi, 2010
- 11) John R Hubbard, *Programming with Java*, 2<sup>nd</sup> ed., Schaum's Outlines, Tata McGraw Hill, New Delhi, 2004

## CS010 603 SYSTEM SOFTWARE

**Teaching scheme**

**Credits: 4**

3 hours lecture and 1 hour tutorial per week

**Objectives:-**

*To introduce the techniques adopted in the design and implementation of System Software.*

**Module I (12 Hrs)**

**Introduction:-**

System Software Vs. Application Software, Different System Software–, Macro Processor, Assembler, Linker, Loader, Text Editor, Debugger, Device Driver, Compiler, Interpreter[1] Database Management System, Operating System,[2]

**Macro Preprocessor**

Macro Instruction Definition and Invocation. Types of Macros – Parameterised macros, Nested macros, Recursive macros. Basic functions of Macro Preprocessor – Macro expansion, Generation of unique labels. Macro preprocessor design and Algorithm - Handling conditional Macro calls, Nested Macro calls and Recursive Macro calls.[Reference (1)] *Case Study : The C Preprocessor* [Web- Reference (1) ]

**Module - II (15 Hrs)**

**Assembler**

Assembly Language Concepts – Mnemonic Instructions, Assembler Directives and Literals. Instruction formats and Addressing modes. Program Blocks and Control Sections. Basic Functions of Assembler. Assembler output format – Header, Text and End Records. Assembler Design – 2 Pass Assembler – Necessity of two passes and Forward reference. Algorithm for the two passes. Single Pass Assembler – Algorithm for Single Pass assembler. Handling External references – usage of Define and Refer records. Multi pass Assembler, Macro Assembler.[Reference (1)] *Case Study : Microsoft Macro Assembler for MS-DOS* [Reference (1), (5)] - *Microsoft OBJ file format (Basic Structure and Important Records Only)* [ Reference(2)].

**Module - III (12 Hrs)**

**Linker and Loader**

Need for Linking and Loading : The absolute loader, Program Relocation, Relocating Loader, Linking external symbols. Algorithms for the two passes of a Linking Loader.[References (2),(3)] Variants of the basic model – Automatic Library Search, Linkage Editor, Dynamic Linking. [Reference(1)] *Case study : UNIX ELF and Microsoft DLL (basic structure only).*

**Module - IV (11 Hrs)**

**Text Editors** : Overview of Editing, User Interface, Editor Structure. [Reference (1)]

*Case Study : VI Editor* (Basic ideas only)[ Reference (1)]

**Debuggers** : Debugging Functions and Capabilities, Relationship with other parts of the system, Debugging Methods- By Induction, Deduction and Backtracking, . [Reference (1) ,(8)] *Case Study : gdb* (Basic ideas only)

**Module - V (10 Hrs)**

**Device Driver** : Device Characteristics ,Design and anatomy, Types of device driver, General Design – Character Devices and character device drivers, Block Devices and Block device drivers. *Case Study: Device Driver for the PC Speaker* [References(4), (6),(7)]

**REFERENCES:**

1. System Software: An Introduction to Systems Programming – Leland L. Beck, Pearson Education Asia 3<sup>rd</sup> Edition.
2. Systems Programming and Operating Systems – D.M. Dhamdhere, Tata McGraw Hill Second Revised Edition.
3. Systems Programming – John J. Donovan, Tata McGraw Hill Edition 1991.
4. Writing UNIX device drivers - George Pajari -Pearson Education Asia.
5. IBM PC Assembly Language and Programming - Peter Abel Third Edition – Prentice Hall of India
6. Linux Device Drivers - Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman – Third Edition - O.Reilly Books
7. Linux Kernel Internals- M. Beck, H. Bohme, M .Dziadzka, et al – Second Edition – Addison Wesley
8. System Software – J Nithyashri –Second Edition- Tata McGraw Hill

**WEB REFERENCE:**

1. [http://gcc.gnu.org/onlinedocs/gcc-2.95.3/cpp\\_1.html](http://gcc.gnu.org/onlinedocs/gcc-2.95.3/cpp_1.html) The C Preprocessor

Note: separate subjects are provided in the syllabus in the Seventh and Fifth Semesters for the detailed discussion of the subjects marked [1] and [2] respectively.



## CS010 604: Computer Networks

### Teaching scheme

3 hours lecture and 1 hour tutorial per week

**Credits: 4**

### Objectives

*To develop basic knowledge on the mode of operation of different types of computer networks that are used to interconnect a distributed community of computers and various interfacing standards and protocols.*

### Module I (8 hours)

Network requirements, Network Architecture –layering and protocol, OSI Architecture, Internet Architecture, Performance-bandwidth and latency , Delay x bandwidth product, high speed networks .

### Module II (10 hours)

Direct Link Network, Hardware Building Block, Framing-Byte Oriented Protocol, Bit Oriented Protocol , Clock Based Framing, Reliable Transmission-Stop and Wait, Sliding Window, Ethernet(802.3)-Physical properties, Access protocol, Wireless-Bluetooth, WiFi, Wimax

### Module III (12 hours)

Packet Switching-Switching and Forwarding- Datagram, virtual circuit switching, Source routing Bridges and LAN Switches-Learning Bridges, Spanning tree Algorithms ,Broadcast and Multicast, Limitations of bridges, Simple Internetworking-Service Model, Global Address, Datagram Forwarding in IP, address translation, Routing-network as graph, distance vector, link state, matrix

### Module IV (16 hours)

End to End Protocol, Simple de-multiplexer, Reliable Byte stream, TCP-Issues, segment format, connection establishment and termination sliding window revisited, triggering transmission, adaptive retransmission, RPC-fundamentals ,TCP Congestion control –additive increase, slow start, fast retransmit and fast recovery, congestion avoidance mechanism, DEC bit, Random Early Detection bit, Source Based Congestion avoidance

### Module V (14 hours)

Applications -WWW, E-mail, Name Service, Network Management, Web Services Custom Application protocol, Generic Application Protocol ,Overlay Networks-Peer to Peer Networks.

### Reference Books

- 1.Computer Networks A Systems Approach-Larry L.Peterson and Bruce S.Davie,4<sup>th</sup> Edition .Morgan Kaufman
2. Introduction to data communication and networking Behrouz Forozan TMH.
- 3 .Computer networks ,Andrew S Tanenbaum ,PHI
- 4.Data communication, computer networks and open systems, Halsall F ,Addison Wesley.





## CS010 605 SOFTWARE ENGINEERING

**Teaching scheme**

**Credits: 4**

3 hours lecture and 1 hour tutorial per week

**Objectives:-**

*To familiarize the steps in designing a Computer Software System following the conventions in Engineering Design.*

*To introduce the fundamentals of Structured and Object Oriented Designs and Design Tools.*

**Module I (12 Hrs)**

The Evolving role of Software – Software – The changing Nature of Software – Legacy software ,Introduction to CASE tools, A generic view of process– A layered Technology – A Process Framework – The Capability Maturity Model Integration (CMMI) – Process Assessment – Personal and Team Process Models. Product and Process. Process Models – The Waterfall Model – Incremental Process Models – Incremental Model – The RAD Model – Evolutionary Process Models – Prototyping – The Spiral Model – The Concurrent Development Model – Specialized Process Models – the Unified Process.

**Module - II (12 Hrs)**

Management: Functions - Project planning - Software productivity - Productivity metrics - Cost estimation - COCOMO & COCOMO II - Project control - Work breakdown structures, Gantt charts, PERT charts - Dealing with deviations - Team organization - centralized, de-centralized, mixed - An assessment of organizations - Risk management – Configuration Management. Introduction to project management and planning CASE tools.

**Module - III (12 Hrs)**

Requirements Engineering : Requirements Engineering tasks – Initiating the requirements Engineering Process-Eliciting Requirements – Developing Use cases – Building the Analysis Models – Elements of the Analysis Model – Analysis pattern – Negotiating Requirements – Validating Requirements. SRS Document.

**Module - IV (12 Hrs)**

Design activity & its objectives – Function Oriented and Object Oriented Design-Modularization techniques - module structure and its representation, interface and information hiding, categories, specific techniques to accommodate change, stepwise refinement, top-down and bottom-up design - Handling anomalies. Case Study with UML and CASE Tool support.

**Module - V (12 Hrs)**

Implementation Techniques - Programming principles and guidelines – Structured Programming. Software Testing Fundamentals-Test Case Design-White-Box Testing-Basis Path Testing-Control Structure Testing- Black-Box Testing- Various levels of Testing : Modules to System. Case study : Test case design and Testlog preparation

**References**

1. Roger S.Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill International edition, Seventh edition.
2. Ian Sommerville, Software Engineering, 8th Edition, Pearson Education, 2008  
(UNIT V)
3. Stephan Schach, Software Engineering, Tata McGraw Hill, 2007
4. Pfleeger and Lawrence Software Engineering: Theory and Practice, Pearson Education, second edition, 2001

## CS010 606L01: DISTRIBUTED SYSTEMS

### Teaching scheme

2 hours lecture and 2 hours tutorial per week

**Credits: 4**

### Objectives

- To impart an introduction to distributed systems and distributed computing.
- To develop basic knowledge on distribution of data and file systems in distributed environment.
- To provide exposure to distributed database concepts.

### Module I (10 hours)

Introduction to Distributed Systems, evolution, characteristics, design issues, user requirements, Distributed computing models-workstation model, workstation-server model, processor-pool model. Protocols for distributed systems -VMTP and FLIP.

### Module II (12 hours)

Client server communication, Group communication, IPC - Message passing – features. RPC – model, implementation, stub generation, RPC messages, communication protocols marshalling. Distributed shared memory – Architecture, design issues, thrashing, replacement strategy. Synchronization – clock synchronization, event ordering, mutual exclusion.

### Module III (14 hours)

Distributed file system: Components of DFS, design issues, interfaces, implementation, File Caching and Replication. Sun Network File System – architecture and implementation, Google File System. Naming- Namespace and contexts and name resolution.

### Module IV (12 hours)

Distributed system management: Features of scheduling algorithms, Task assignment approach, load balancing, load sharing, Process migration mechanisms, Threads – design issues, Fault tolerance – failures, Byzantine failures.

### Module V (12 hours)

Distributed Databases: Distributed DBMS architecture, distributed query processing, transactions, concurrency control, deadlock management and Distributed Database Recovery protocols-2PC, Network Partitioning.

### Reference Books

1. Sunita Mahajan, Seema shah, *Distributed Computing* ,Oxford University Press, first edition, 2010
2. George Coulouris, Jean Dellimore and Tim Kindberg, *Distributed Systems – Concepts and designing*, Pearson Education Asia, fourth Edition 2006, New Delhi.
3. Pradeep. K, Sinha, *Distributed Operating Systems* ,PHI Edition, first Edition,1997.
4. Andrew S Tenenbaum, *Distributed Operating Systems*, Pearson Education Asia



**CS010 606L02 Micro controller Based Systems**

(Common with EE010 503 and EC010 502)

**Teaching scheme**

2 hours lecture and 2 hours tutorial per week

**Credits: 4**

**Objectives**

- *To impart the basic concepts of microcontrollers and their programming in assembly language and in C.*
- *It also focused on the 8051 microcontroller which is a widely used microcontroller.*

**Pre-requisites:** *Microprocessor systems, Advanced microprocessor and peripherals*

**Module I (10 hours)**

Microcontroller - Features of 8051-Architecture of 8051-Pin diagram of 8051-memory organization-External memory interfacing-stacks- addressing modes-instruction set.

**Module II (12 hours)**

8051 programming in C-data types and time delay – I/O programming – logical operation – data conversation program –basics of serial communication connection to RS232- serial port programming in assembly and C.

**Module III (14 hours)**

Basics of interrupts,-interrupt sources- interrupt enable register-interrupt priority-interrupt control system-interrupt handling-single step operation- port bit latches and buffers-port structures and operation- accessing external memory.

**Module IV (12 hours)** Timer 0& -Timer1- T MOD SFR-mode0,mode 1,mode2,mode3-TCON SFR-serial interface-SCON SFR-mode0,mode 1,mode 2,mode3-block schematics-baud rates-power on reset circuit-ONCE mode-on chip oscillator-external program & data memory timing diagrams.

**Module V (12 hours)**

PIC microcontrollers: Overview and features-PIC16C6X/7X FSR-Reset action-PIC memory organization-instructions-addressing modes.

**Reference Books**

1. Muhammad Ali mazidi, Janice Gillispie Mazidi, Rolin D Mc kinlay , *The 8051 microcontroller and embedded systems*, person, second edition., 2006
2. V Udayashankara, M S Mallikarjunaswamy , 8051 Microcontroller hardware & software application, TMH
3. Ajay V Deshmukh, Microcontrollers, theory and applications, TMH
4. Kenneth J Ayala, *The 8051 microcontroller.*, Penram International
5. 1 Satish Shah, *8051 microcontrollers MCS 51 family and its variants* ,Oxford higher education

## CS010 606L03: User Interface Design

### Teaching scheme

2 hours lecture and 2 hours tutorial per week

Credits: 4

### Objectives

- *To impart the basic concepts of User Interface Design.*
- *To develop understanding about human computer interaction methods that utilize more general, widespread and easier-to-learn capabilities.*

### Module I (8 hours)

Introduction: Importance of user interface – definition, importance of good design, brief history – Graphical User Interface – Web User Interface – Theories, Principles and Guidelines of User interface design

### Module II (10 hours)

Design Process: Obstacles in development path designing for people-Understanding Human Interaction with computers, Importance of Human Characteristics, Human consideration, Human Interaction speeds – Understanding Business function

### Module III (15 hours)

Screen Designing: Design goals - screen meaning and purpose, organizing screen elements-ordering of screen data and content – screen navigation and flow – visually pleasing composition – amount of information – focus and emphasis – presenting information simply and meaningfully – information retrieval on web – Statistical graphics – Technological considerations in Interface Design.

### Module IV (15 hours)

Menus and navigation schemes-structures of menus-functions of menus- contents of menus - formatting of menus – phrasing the menu- selecting menu choices-navigating menus-kinds of graphical menus- Selection of windows-Window characteristics-components of window-window presentation styles-types of windows-window management-organising window functions-window operations-Selection of device based and screen based controls - text and messages – icons and images – Multimedia – colours- uses, problems, choosing colours.

### Module V (12 hours)

Distributed and Collaborative Interaction-Device consistency-distribution of the user interface-event distribution-graphical package layer-programmable API-Model semantics distribution-data layer distribution-asynchronous collaboration-Software tools-specification methods- interface building tools –evaluation and critiquing tools-Interaction devices-keyboard and function keys - pointing devices- speech recognition, digitization and generation – image and video displays – printers.

### Reference Books

1. Wilbert O. Galitz, *The Essential Guide to User Interface Design*, 2<sup>nd</sup> Edn., Wiley Dreamtech, Delhi, 2002
2. Ben Shneiderman, *Designing the User Interface*, 3<sup>rd</sup> Edn., Pearson Education Asia, Delhi, 2002
3. Dan R. Olsen, *Human Computer Interaction*, Cengage, New Delhi, 2009
4. John M. Carroll, *Human Computer Interaction*, Pearson Education Asia, Delhi, 2002
5. Alan Cooper, *The Essentials of User Interface Design*, Wiley Dreamtech, Delhi, 2002





## **CS010 606L04 : UNIX Shell Programming (Common with IT010 606L03)**

**Teaching scheme**  
2 hours lecture and 2 hour tutorial per week

**Credits: 4**

### **Objectives**

- *To provide a fair knowledge of Unix concepts and gain sharp skills in Unix Shell programming*

### **Module 1. (8 hours)**

Introduction to Unix:- Architecture of Unix, Features of Unix , Basic Unix Commands - Unix Utilities:- Introduction to unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands - Text processing utilities and backup

### **Module 2. (13 hours)**

Introduction to Shells:-Unix Session, Standard Streams, Redirection, Pipes, tee Command, Command Execution, Command-Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment Customization. Regular expressions, Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count characters, words or lines, Comparing Files.

### **Module 3. (12 hours)**

grep:-Operation, grep Family, Searching for File Content.  
sed:-Scripts, Operation, Addresses, commands, Applications, grep and sed.  
awk:-Execution, Fields and Records, Scripts, Operations, Patterns, Actions, Associative Arrays, String Functions, Mathematical Functions, User Defined Functions, Using System commands in awk, Applications of awk, grep and sed

### **Module 4. (15 hours)**

Interactive Shells - Korn Shell, C Shell and BASH - Shell Features, Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, Options, Startup Scripts, Command History, Command Execution Process.

Shell Programming - Korn Shell, C Shell and BASH -

Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.

### **Module 5. (12 hours)**

Process management:- Creation, Hierarchies, Sending signals to processes, exec, termination, Zombie, waitpid etc - Network management:- tools, Client server mechanism, address resolution, ping, telnet, ftp, dns and squid – X Window System:- Overview, Architecture, starting and stopping X, X clients and display

**Reference Books**

1. Behrouz A. Forouzan, Richard F. Gilberg, " Unix and shell Programming.", Cengage Learning
2. Sumitabha Das , "Unix the ultimate guide", TMH. 2nd Edition.
3. Kernighan and Pike, "Unix programming environment", PHI. / Pearson Education
4. Graham Glass, King Ales, " Unix for programmers and users", 3rd edition, Pearson Education
5. Maurice J. Bach, "The Design of the Unix Operating System", First Edition, Pearson Education, 1999

## CS010 606L05: Embedded Systems

### Teaching scheme

2 hours lecture and 2 hours tutorial per week

**Credits: 4**

### Objectives

- *To impart the basic concepts of Embedded System and its applications*
- *To develop understanding about micro controllers and programming the micro controller for the development of Embedded systems.*

### Module I (-12 hours)

Introduction to Embedded Systems-Classification of Embedded Systems-Application areas of Embedded Systems, Typical Embedded System- Memory-Sensors and Actuators-Embedded Firmware - Characteristics and Quality Attributes of Embedded Systems

### Module II (13 hours)

Application Specific Embedded System – Domain Specific Embedded System, Designing Embedded Systems with 8bit Microcontrollers- Factors to be considered in selecting a Controller- Designing with 8051 microcontroller- 8052 microcontroller, Programming the 8051 microcontroller – Addressing modes of 8051 – the 8051 Instruction set

### Module III ( 13 hours)

Hardware Software Co-Design and Program Modeling – Computational models in Embedded Design, Embedded Hardware Design and development – Electronic Design Automation Tools, Embedded Firmware Design and Development - Embedded Firmware Design Approaches - Embedded Firmware Development Languages – Programming in Embedded C.

### Module IV (12 hours)

Real Time Operating System based Embedded System Design – Operating System Basics – Types of Operating Systems – Tasks- Process- Threads – Multiprocessing and Multitasking – Task Scheduling – Task Communication – Task Synchronization – Introduction to Vx Works and Micro C/OS-II RTOS

### Module V (10 hours)

The Embedded System Development Environment – Integrated Development Environment , The Embedded Product Development Life Cycle – EDLC- Objectives of EDLC – Different phases of EDLC – Modeling the EDLC

**Reference Books**

1. Shibu K V, *Introduction to Embedded Systems*, McGraw Hill, New Delhi, 2009
2. Raj Kamal, *Embedded Systems Architecture, Programming and Design*, 2<sup>nd</sup> ed., Tata McGraw Hill, New Delhi, 2008
3. Frank Vahid & Tony Givargis, *Embedded System Design A Unified Hardware/Software Introduction*, Wiley - India Edition, New Delhi, 2010
4. Wayne Wolf, *Computers as Components Principles of Embedded Computing System Design*, 2<sup>nd</sup> ed., Elsevier, Gurgaon, 2009
5. Steven F Barrett & Daniel J Pack, *Embedded Systems Design and Applications with the 68HC12 and HCS12*, Pearson Education, Delhi, 2008.

## CS010 606L06: Advanced Software Environments

### Teaching scheme

2 hours lecture and 2 hours tutorial per week

**Credits: 4**

### Objectives

- To impart the basic concepts of Windows programming..
- To develop understanding about the new software environment and develop of software to meet the growing demand of the industry.

**Pre-requisites:** Knowledge required to study this subject (OOP concepts))

### Module I (10 hours)

Windows Programming – Components of Windows API- Distinction with ordinary programs – Event Driven Programming – WinMain Function – Creating Windows – Message loop – Window procedures - Menus & Buttons – Drawing on Windows, Advanced User Interface concepts, Developing application issues and solutions.

### Module II (10 hours)

MFC Features & Advantages – MFC Classes – Life cycle of an MFC application – The CWinApp Classes – Creating windows – Message maps and event handling – Menus & Buttons - Drawing on MFC windows – Handling mouse & Keyboard events.

### Module III (13 hours)

X-Windows – Clients & Servers - Basic Architecture of X-Windows systems – Layers in XWindows Architecture – XWindows Programming – Simple Hello World Application in X. Command line options and resources – connecting to X-Display – creating windows and graphics context – Handling events – creating child windows.

### Module IV (13 hours)

CORBA – Introduction – Features – Fundamental concepts in Distributed objects – CORBA IDL – stub & Skeleton - implementing a simple CORBA server and CORBA client with C++.

### Module V (14 hours)

CORBA object reference – Managing references at server – CORBA factories – CORBA object creation in C++ & JAVA – CORBA Exceptions – Destroying CORBA objects - comparison of CORBA & DCOM Architectures.

### Reference Books

1. Yashwanth Kanetkar , Visual C++ Programming ,BPB Publications ,New Delhi, 2005.
2. Mike Blaszcza, Professional MFC with Visual C++ 6, 4<sup>th</sup> Edition, Shroff publishers & Distributors Private Limited, New Delhi, 2003.
3. Nabajyoti Bakakati, X Window System programming , 2<sup>nd</sup> Edition, Prentice-Hall of India Private Limited,New Delhi, 2001.
4. Jason Pritchard ,COM & CORBA side by side , Pearson Edition New Delhi, 2000.



## CS010 607: Operating Systems Lab

### Teaching scheme

3 hours practical per week

**Credits: 2**

### Objectives

- *To provide a practical exposure of all algorithms and behaviour of processes in the system with respect to all its timings.*
- *This lab also explains the allocation of process in the memory with some memory management techniques.*

(Implement the following on LINUX platform. Use C for high level language implementation)

1. Basic UNIX commands
2. Shell programming
  - Command syntax
  - Write simple functions with basic tests, loops, patterns
3. Write programs using the following system calls of UNIX operating system:  
fork, exec, getpid, exit, wait, close, stat, opendir, readdir
4. Write programs using the I/O system calls of UNIX operating system (open, read, write, etc)
5. Write C programs to simulate UNIX commands like ls, grep, etc.
6. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time
7. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time
8. Implement the Producer – Consumer problem using semaphores.
9. Implement inter-process communication using shared memory.
10. Implement some memory management schemes

Example for expt 10:

Free space is maintained as a linked list of nodes with each node having the starting byte address and the ending byte address of a free block. Each memory request consists of the process-id and the amount of storage space required in bytes. Allocated memory space is again maintained as a linked list of nodes with each node having the process-id, starting byte address and the ending byte address of the allocated space.

When a process finishes (taken as input) the appropriate node from the allocated list should be deleted and this free disk space should be added to the free space list. [Care should be taken to merge contiguous free blocks into one single block. This results in deleting more than one node from the free space list and changing the start and end address in the appropriate node]. For allocation use first fit, worst fit and best fit.



## CS010 608 Mini Project

### Teaching scheme

3 hours practical per week

Credits: 2

### Objectives

- *To estimate the ability of the student in transforming the theoretical knowledge studied so far into application software.*
- *For enabling the students to gain experience in organisation and implementation of a small project and thus acquire the necessary confidence to carry out main project in the final year.*
- *To understand and gain the knowledge of software engineering practices, so as to participate and manage large software engineering projects in future.*

In this practical course, each group consisting of two/three members (four in special cases) is expected to design and develop practical solutions to real life problems related to industry, institutions and computer science research. Software life cycle should be followed during the development. The theoretical knowledge, principles and practices gained from various subjects should be applied to develop effective solutions to various computing problems. The knowledge gained during various practical subjects to work with various software tools, Designing tools, programming languages, operating systems, etc. should be utilized in various stages of development. Structured/ Object Oriented design techniques may be used for the project. Software Requirements Specification (SRS), Modeling Techniques, Design and Testing strategies should be documented properly.

A committee consisting of minimum three faculty members will perform the internal assessment of the mini project. A report on mini project should be submitted for evaluation and project work should be presented and demonstrated before the panel of examiners.

#### **Internal Continuous Assessment (50 marks)**

40% - Design and development (30% by guide and 10% by committee)

30% - Final result and Demonstration (15% by guide and 15% by committee)

20% - Report (10% by guide and 10% by committee)

10% - Regularity in the class (by guide)

#### **End Semester Examination (Maximum Marks-100)**

20% - Demonstration of mini project

50% - Practical test connected with mini project

20% - Viva voce

10% - Project report