FIRST SEMESTER

Advanced Operating Systems

Course Title: Advanced Operating Systems **Full Marks:** 45 + 30

Course No: C.Sc. 538 **Pass Marks:** 22.5 + 15

Nature of the Course: Theory + Lab **Credit Hrs:** 3

Course Description: Operating systems concepts overview - process management, memory management and storage management, protection and security, distributed system, real-time system, and multimedia system.

Course Objectives:

- Introduce the underlying principles of an operating system, virtual memory and resource management concepts.
- Exposure current state-of-art research in operating system
- Exposure of distributed operating system, real-time operating system and multimedia systems.

Course Contents

Unit 1: Process Management and Synchronization(12 Hrs)

1.1. Process management: Process Model, Modeling Multiprogramming, Inter-process communications: lock variables, sleep and wakeup, semaphores and mutexes.
1.2. Process scheduling: Round robin, *Weighted Round Robin, Round robin with variable quanta (Paper Based Study)*
1.3. Deadlocks: Deadlock detection, avoidance and prevention, *recovery from deadlocks (Paper Based study)*.

Unit 2: Memory Management (13 hrs)

2.1. Paging: Page tables, speeding up page tables, Page tables for large memories, demand Paging, Page replacement algorithms: Brief overview of LRU and LFU, Details of clock, WSclock, *LRFU, LIRS(Paper Based Study)*, Design Issues for paging, Segmentation, Segmentation with paging.
2.2. Storage Management: Free-space management, File system layout, Implementing files and directories, log structured file system, journaling file system, virtual file system, Disk scheduling, flash file systems. *high performance flash disks, Improving the performance of log structured file systems with adoptive block rearrangement, Buffer Cache Management scheme exploiting both temporal and spatial locality(Paper Based Study)*

Unit 3: Protection and Security (8 Hrs)

3.1. System Protection: protection principles and domain, access matrix and its implementation, access controls and rights, capability-based system, language-based protection.
3.2. System Security: Program threats, system and network threats, Cryptography as a Security Tool, User Authentication, Firewalling, Computer security classification, *Guarded Models for intrusion detection (Paper Based Study)*

Unit 4: Distributed Special Purpose Systems (12 Hrs)

4.1. Distributed Operating Systems: Types of Network based Operating Systems, Network structure and topology, Communication structure and protocols, Robustness, Design Issues, Distributed File Systems: Naming and Transparency, Remote File Access, Stateful versus Stateless Service, File Replication.
4.2. Real-Time Systems: Real time system characteristics and kernel features, Implementing real-time operating system, real-time CPU scheduling.
4.3. Multimedia Systems: Multimedia system overview, multimedia kernels, compression, CPU scheduling, disk scheduling.

Outcomes and Assessment

This course can be learnt in effective way only if we give focus in practical aspects of algorithms and techniques discussed in class. Therefore student should be able to simulate and analyze the algorithms by using any high level language. For this purpose student should simulate at least

- Two page replacement algorithms
- Two disk scheduling algorithms
- Deadlock recovery algorithm
- Two File System techniques

Recommended Book:

1. Silberschatz, Galvin and Gagne. Operating System Concepts, 8th Edition. John-Wiley.
2. Andrew S Tanenbaum, Moderm Operating Systems, 3rd Edition 20007, PH1

Algorithms and Complexity

Course Title: Algorithms and Complexity Full Marks: 45+30

Course Noe: C.Sc. 540 Pass Marks: 22.5+15

Credit Hours = 3

Nature of the course: Theory +Lab

Course Description

The course can be viewed as a continuation of Design and Analysis of Algorithms, and its goal is to cover basic algorithmic techniques that are not covered in the subject. In particular, we will discuss NP Completeness in detail, approximation algorithms for NP-hard problems, Randomized Algorithms, online algorithms, PRAM algorithms, Mesh Algorithms and Hypercube Algorithms.

Course Objectives

The purpose of this course is to present depth in optimization paradigms, Some advance algorithm design techniques and moderate level understanding in computational complexity theory.

Prerequisites

Linear Algebra, Data Structure and algorithm, Basic Course in Algorithm design, Programming language

Course Contents

Unit 1: Advance Algorithm Analysis and Design Techniques 10 hr

1.1. **Advanced Algorithm Analysis Techniques:** Amortized Analysis(Aggregate Analysis, Accounting Method, Potential method), Probabilistic Analysis, Las Vegas and Monte Carlo Algorithms.
1.2. **Advance Algorithm Design Techniques:** Greedy Algorithms (Tree Vertex Splitting, Job Sequencing with deadlines), Dynamic Programming(Greedy vs Dynamic, String Editing, Optimal BST), backtracking(Sum of Subsets, Knapsack Problem), Randomized Algorithms(Identifying the repeated elements, Primality Testing, Karger's algorithm)

Unit 2: Computational Complexity Theory 10 hr

2.1 **Basic Concepts:** Complexity Theory, Complexity Classes: P, NP, NP-Hard and NP-Complete, Decision Problems and Language Recognition Problems

2.2 **Problem Reduction:** Reduction, Polynomial time reduction, Cooks Theorem, Proving NP-Completeness: Formula Satisfiablity, 3SAT, CLIQUE, Vertex Cover, Hamiltonian Cycle, TSP, Subset Sum.

2.3 **NP-Hard Code Generation Problems:** Code Generation with Common Subexpression, Implementing Parallel Assignment Instructions

2.4 **Coping with NP-Completeness:** Performance ratios for approximation algorithms, Approximated Algorithms: Vertex Cover, TSP, Set Covering, Subset Sum

Unit 3: Online and PRAM Algorithms 12 hr

3.1 **Online Algorithms:** Introduction, Ski Problem, Load Balancing, Paging and Caching: Last-in First-out (LIFO), Longest Forward Distance (LFD), Least Recently Used (LRU)

3.2 **PRAM Algorithms:** Introduction, Computational Model, Fundamental techniques and Algorithms (Prefix Computation, List Ranking), Selection (Maximal Selection with n^2 Processors, Finding Maximum Using n Processors), Merging (A Logarithmic Time Algorithm, Odd-Even Merge), Sorting(Odd-Even Merge Sort, Preparata's Algorithm).

Unit 4: Mesh and Hypercube algorithms 13hr

4.1 **Mesh Algorithms:** Computational Model, Packet Routing (Packet Routing on a Linear Array, A Greedy Algorithm for PPR on a Mesh), Fundamental Algorithms (Broadcasting, Prefix Computation, Data Concentration), Selection (Randomized Algorithm for n=p, Randomized Selection for n>p), Sorting (Sorting on a Linear Array, Sorting on a Mesh)

4.2 **Hypercube Algorithms:** Computational Model (Hypercube, Butterfly Network, Embedding of other Networks, PPR Routing (Greedy algorithm, Randomized Algorithm), Fundamental Algorithms (Broadcasting, Prefix Computation, Data Concentration), Selection (Randomized Algorithm for n=p, Randomized Selection for n>p), Sorting(Odd-Even Merge Sort, Bitonic Sort)

Textbook

1. Ellis Horowitz, Sartaj Sahni,Sanguthevar Rajasekaran, **Computer Algorithms/C++,** 2nd Edition, Universities Press, 2007

Reference Books

1. H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, **Introduction to Algorithms,** 2nd Edition, MIT Press, 2001

2. G. Brassard and P. Bratley, **Fundamentals of Algorithmics,** Prentice-Hall, 1996

Outcomes and Assessment

Student should implement Algorithms described in class and should perform their Empirical Evaluation.

Neural Networks

Course Title: Neural Networks

Full Marks: 45 + 30

Course No: C.Sc. 543

Pass Marks: 22.5 + 15

Nature of the Course: Theory + Lab

Credit Hrs: 3

Course Description:

This course deals with resembling the conscious behavior of brain in an artificially connected logical nodes that can learn itself in a supervised and unsupervised environment. It introduces linear algebra for creating and processing input patterns to generate output. It also emphasizes on probabilistic model for classification and prediction within a tolerable and significant limit.

Unit 1: Introduction to artificial neural networks 5 hrs

Biological neural networks; Pattern analysis tasks: Classification, Regression, Clustering; Computational models of neurons; Structures of neural networks; Learning principles

Unit 2: Linear models for regression and classification 7 hrs

Polynomial curve fitting; Bayesian curve fitting; Linear basis function models; Bias-variance decomposition; Bayesian linear regression; Least squares for classification; Logistic regression for classification; Bayesian logistic regression for classification

Unit 3: Feedforward neural networks 7 hrs

Pattern classification using perceptron; Multilayer feedforward neural networks (MLFFNNs); Pattern classification and regression using MLFFNNs; Error backpropagation learning; Fast learning methods: Conjugate gradient method; Autoassociative neural networks; Bayesian neural networks

Unit 4: Radial basis function networks 8 hrs

Regularization theory; RBF networks for function approximation; RBF networks for pattern classification

Unit 5: Kernel methods for pattern analysis 8 hrs

Statistical learning theory; Support vector machines for pattern classification; Support vector regression for function approximation; Relevance vector machines for classification and regression

Unit 6: Self-organizing maps 5 hrs

Pattern clustering; Topological mapping; Kohonen’s self-organizing map

Unit 7: Feedback neural networks **5 hrs**

Pattern storage and retrieval; Hopfield model; Boltzmann machine; Recurrent neural networks

Text Books:

1. B.Yegnanarayana, Artificial Neural Networks, Prentice Hall of India, 1999
2. Satish Kumar, Neural Networks – A Classroom Approach, Tata McGraw-Hill, 2003
3. S.Haykin, Neural Networks – A Comprehensive Foundation, Prentice Hall, 1998
4. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006

Object-Oriented Software Engineering

Course Title: Object-Oriented Software Engineering
Course No: C.Sc. 539
Nature of the Course: **Theory + Case Studies+ Project**

Full Marks: 45 + 30
Pass Marks: 22.5 + 15
Credit Hrs: 3

Course Objectives:

- This course aims to give students both a theoretical and a practical foundation in Object - Oriented software engineering.
- In the theoretical part, students will learn about the principles and methods of Object Oriented software engineering, including current and emerging Object Oriented software engineering practices and support tools.
- In the practical part, students will become familiar with the development of Object Oriented software products from an industry perspective.

Unit 1: 4 Hrs
Software life cycle models, Requirement analysis and specification, object oriented software development,

Unit 2: 5 Hrs
Introduction to object-orientation, object- oriented system development-function /data methods, object oriented analysis, construction and testing, object -oriented programming with examples

Unit 3: 10 Hrs
Architecture-model architecture, requirements, analysis, design, implementation and test model, analysis, construction, Real-time-classification of real time systems, database-RDBMS, Object DBMS, components-use of components, component management, testing-on testing, Unit, integration, system and the testing process

Unit 4: 6 Hrs
Managing object-oriented software engineering, Project selection and preparation, product development organization, project organization and management, project staffing, software quality assurance, software metrics

Unit 5: 5 Hrs
Object oriented analysis and design, hierarchical object-Oriented design, object Modeling technique and responsibility-driven design

Unit 6: 15 Hrs
Case studies and project
Warehouse management system, Telecom

References

1. Ivar Jacobson-Object-Oriented software engineering
2. Ian Sommerville-Software Engineering
3. Grady Booch-Object-oriented analysis and design

Parallel and Distributed Computing

Course Title: Parallel and Distributed Computing **Full Marks:** 45 + 30
Course No: C.Sc. 544 **Pass Marks:** 22.5 + 15
Nature of the Course: Theory + Lab **Credit Hrs:** 3

Course Description:

This course covers a broad range of topics related to parallel and distributed computing, including foundations, models, and algorithms.

Course Contents:

Unit 1: Foundations

1.1. Parallel and Distributed Computing: the Scene, the Props, the Players (2 hrs)

A Perspective, Parallel Processing Paradigms, Modeling and Characterizing Parallel Algorithms, Cost vs. Performance Evaluation, Software and General Purpose PDC

1.2.Semantics of Concurrent Programming (2 hrs)

Models of Concurrent Programming, Semantic Definitions, Axiomatic Semantic Definitions, Denotational Semantic Definitions, Operational Semantic Definitions

1.3.Formal Methods: A Petri Nets Bases Approach (2 hrs)

Process Algebras, PETRI Nets, High-Level New Models

1.4. Complexity Issues in Parallel and Distributed Computing (3 hrs)

Introduction, Turing Machines as the Basis and Consequences, Complexity Measures for Parallelism, Parallel Complexity Models and Resulting Classes, VLSI Computational Complexity, Complexity Measures for Distributed Systems, Neural Networks and Complexity Issues

1.5. Distributed Computing Theory (3 hrs)

The Computational Model, A Simple Example, Leader Election, Sparse Network Covers and Their Applications, Ordering of Events, Resource Allocation, Tolerating Processor Failures in Synchronous Systems, Tolerating Processor Failures in Asynchronous Systems, Other Types of Failures, Wait-Free Implementations of Shared Objects

Unit 2: Models

2.1.PRAM Models (5hrs)

Introduction; Techniques for the Design of Parallel Algorithms; The PRAM Model; Optimality and Efficiency of Parallel Algorithms; Basic PRAM Algorithms; The NC-Class; P-Completeness: Hardly Parallelizable Problems; Randomized Algorithms and Parallelism; List Ranking Revisited: Optimal O(log n) Deterministic List Ranking; Taxonomy of Parallel Algorithms; Deficiencies of the PRAM Model

2.2.Broadcasting with Selective Reduction: A Powerful Model of Parallel Computation (3 hrs)

Introduction; A Generalized BSR Model; One Criterion BSR Algorithms; Two Criteria BSR Algorithms; Three Criteria BSR Algorithms; Multiple Criteria BSR Algorithms

2.3.Dataflow Models (3 hrs)

Kinds of Dataflow; Data-Driven Dataflow Computing Models; Demand-driven Dataflow Computing Models; Unifying Data-Driven and Demand-Driven

2.4.Partitioning and Scheduling (4 hrs)

Program Partitioning; Task Scheduling; Scheduling System Model; Communication Models; Optimal Scheduling Algorithms; Scheduling Heuristic Algorithms; Scheduling Nondeterministic Task Graphs; Scheduling Tools; Task Allocation; Heterogeneous Environments

2.5.Checkpointing in Parallel and Distributed Systems (3 hrs)

Introduction; Checkpointing Using Task Duplication; Techniques for Consistent Checkpointing

2.6.Architecture for Open Distributed Software Systems (2 hrs)

Introduction to Open Distributed Systems Architecture; Computational Model; Engineering Model; ODP Application

Unit 3: Algorithms

3.1.Fundamentals of Parallel Algorithms (3 hrs)

Introduction; Models of Parallel Computation; Balanced Trees; Divide and Conquer; Partitioning; Combining

3.2.Parallel Graph Algorithms (5 hrs)

Graph-Theoretic Concepts and Notation; Tree Algorithms; Algorithms for General Graphs; Algorithms for Particular Classes of Graphs

3.3.Data Parallel Algorithms (5 hrs)

Chapter Overview; Machine Model; Impact of Data Distribution; CU/PE Overlap; Parallel Reduction Operations; Matrix and Vector Operations; Mapping Algorithms onto Partitionable Machines; Achieving Scalability Using Set of Algorithms

Recommended Books:

1. Parallel and Distributed Computing Handbook, Albert Y. Zomaya, Editor, McGraw-Hill
2. Introduction to Parallel Computing; Ananth Grama, Anshul Gupta, George Kaprypis, Vipin Kumar; Pearson Education
3. An Introduction to Parallel Programming, Peter Pacheco
4. Quinn, Michael J., Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2004.
5. Quinn, M., Parallel Computing: Theory and Practice, McGraw Hill, 1994.

Seminar I

Course Title: Seminar I
Course No: C.Sc. 542
Nature of the Course: Seminar

Full Marks: 25
Pass Marks: 12.5
Credit Hrs: 1

Course Description: The seminar-I is of full marks 25 offered in the curriculum of the M. Sc. first year first semester. A student pursuing the seminar prepares a seminar report and presents the seminar in the department. Once accepted by the department, the students have to submit the final copy of the report.

Introduction:
Each student is required to write a comprehensive report about the seminar. The report should consist of 5 to 10 pages describing the topic selected. Students can choose the seminar topics of their relevant subject area. The students are suggested to select the research oriented topics rather than just informative ones. The report should be in the format as described below;

Arrangement of Contents:
The sequence in which the seminar report material should be arranged and bound should be as follows:

1. Cover Page & Title Page
2. Abstract
3. Chapters:
    a. Introduction
    b. Previous Works, Discussions and Findings
    c. Conclusion
4. References

Format of References
1. References

A list of all publications (articles, texts, monographs, etc.) must be supplied as the last section of the paper. Each article or paper used must be listed alphabetically by last name of the author and the list must be numbered sequentially. The following are examples of the format for various types of entries in the list.

Journal: Stalling, W., RSA and its computational aspects, *Infoworld*, 12, 28 (Jul. 2012)**,** 42-49.

Book: Bishop, M. and Boneh, D., *Elements of Computer Security*, Pearson Education., 2009.

Proceedings: Shamir, A., Controlling attacks on public key cryptography. *Proceedings of OOPSLA 86* (Sept. 1986., Portland), 405-416.