**ACADEMIC-GRADUATE STUDIES AND RESEARCH DIVISION**
**FIRST SEMESTER 2023-2024**
**Course Handout Part II**

Date: 09.08.2023

In addition to Part-I (General handout for all courses appended to the timetable) this portion gives further specific details regarding the course:

**Course No.** : **CS G623**
**Course Title** : **Advanced Operating Systems**
**Instructor-in-Charge** : **Dr. Pragati Shrivastava**

**Description :** Overview of advanced operating systems: motivation for their design, and various types of advanced operating systems; Distributed operating systems: architecture of distributed systems, theoretical foundation of distributed systems, deadlock detection/resolution, agreement protocols, file systems, distributed shared memory, scheduling, fault tolerance and recovery; Multiprocessor operating systems: multiprocessor system architectures, multiprocessor operating system design issues, the reads, process synchronization, process scheduling and memory management; Data base operating systems: introduction, concurrency control: theoretical and algorithmic aspects; Case Study: Amoeba and Mach

**Scope:**

Over the last few decades, considerable advancements have taken place in the area of Advanced Operating Systems, mostly referred to as Distributed Operating Systems. The course aims to introduce the design and implementation issues of Advanced Operating Systems. The scope of this course is on understanding how the Distributed Operating Systems work in an environment where we have independent machines (both hardware and software) connected with each other over a computer network. The students would get to know the fact that Distributed OSs have at their center the reasoning that you should use faster machines for more tasks that need speed, and slower ones for the tasks that do not. Also, central to the design of the distributed OS is making the design transparent to the user. The course would help the student understand how a Distributed OS enables the distributed system to act as a virtual uniprocessor system.

**Objectives:**

- To get a comprehensive understanding of the advanced aspects of the Distributed Operating Systems
- To learn about the *microkernel* based distributed operating systems where the user level processes that are separated from the kernel can run on remote machines
- To explore case studies like Sun NFS, HDFS, GFS, MapReduce, Vector Clocks, Causal ordering, Agreement protocols, Mutual Exclusion, distributed file systems etc.
- To implement the case studies through programming in the practical sessions

**Text Book:**

T1 M. Singhal & N. Shivaratri, “Advanced Concepts in Operating Systems: Distributed, Database and Multiprocessor Operating Systems”, Tata McGraw Hill, 2001.

**Reference Books:**

R1. Ajay D. Kshemkalyani, and Mukesh Singhal "Distributed Computing: Principles, Algorithms, and Systems", Cambridge University Press, 2008 (Reprint 2013).
R2 Distributed System, by Martin van Steen and A. S. Tanenbaum, 3rd Edition 2023.
R3. Distributed Operating Systems – The Logical Design by A. Goscinski, AW, 1991.
R4. Modern Operating Systems – Design and Implementation by A. S. Tanenbaum, 4th Edition, 2014, PHI.
R5. Distributed Systems-Concepts and Design by G. Coulouris, 5th Edition, 2011, Pearson.
R6. Introduction to Reliable and Secure Distributed Programming (second edition) by Cachin, Guerraoui and Rodrigues, Springer,

**Course Plan:**

| **Lect No.** | **Learning Objectives** | **Topics to be covered** | **Chapter No. in Text Book** |
|---|---|---|---|
| 1 - 2 | To learn about the basics of distributed operating systems | Introduction to distributed systems, Characteristics of a distributed system, Challenges in building distributed systems, | Chapter 1 |
| 3 - 4 | To understand the need to have a virtualized environment and its characteristics | Traditional Vs Virtualized architecture, Different types of hypervisors, Need for virtualization, Uses of Virtual machines | Class notes |
| 5 - 7 | To learn about the need for logical clocks in building a distributed system and to understand causal ordering of messages | Limitations of a distributed system, Lamport's logical clock, Fidge-Mattern's vector clock, causal ordering of messages, global state of a distributed system, Cuts. | Chapter 5 |
| 8 - 11 | To understand the problem of critical section in a distributed environment | Lamport's DME, Ricart-agrawala, and Maekawa's algorithms; Suzuki-kasami broadcast algorithm, and Raymond's tree based algorithm. | Chapter 6 |
| 12 - 15 | To understand deadlocks which are seen as challenges in building distributed systems. Also to know about deadlock handling without using avoidance algorithms like Bankers | Resource Vs. Communication deadlock, Strategies to handle deadlock, Ho-Ramamoorthy algorithm, Path-Pushing, Edge-Chasing, Diffusion Computation-based algorithms. | Chapter 7 |
| 16 - 19 | To understand the importance of consensus in distributed systems | System model, Classification of agreement problems, Solutions to Byzantine agreement problems. | Chapter 8 |
| 20 - 25 | To learn about Distributed transparent storage specifics like accessing data stored at different machines and characteristics of Bigdata | Bigdata characteristics, Structured vs. unstructured data, Mechanisms for building DFSs, Design Issues, Case studies: Sun DFS, Sprite DFS, Hadoop Distributed File System (HDFS): Hadoop cluster, HDFS communications, HDFS read/write operations. | Chapter 9 and Class notes. |

| | | | |
|---|---|---|---|
| 26 - 31 | To understand the need to run a single computation at multiple places | Issues in Load Distribution, Components of a load distribution algorithm, Load Distribution Algorithms, Case studies: System-V, Sprite, and Condor, HDFS MapReduce: Programming on Hadoop. | Chapter 11 and Class notes. |
| 32 | Research paper discussions. | Recent research on DFS and Distributed schedulers. | IEEE/ACM |
| 33 - 36 | To learn to build a logical memory consisting of RAM contents of different machines | Algorithms for implementing Distributed Shared Memories (DSMs), Memory Coherence models, and Coherence Protocols, Case study: Integrated Virtual Memory at Yale (IVY). | Chapter 10 |
| 37 - 40 | To understand the process for recovery process from failures in distributed systems | Classification of failures, Synchronous and Asynchronous Check Pointing and Recovery algorithms. | Chapter 12 |

**Evaluation Scheme**:

| **EC No.** | **Evaluation Component** | **Duration**(Min) | **Weightage (%)** | **Date & Time** | **Nature of Component** |
|---|---|---|---|---|---|
| 1 | Mid Sem Test | 90 | 25 | 09/10- 4:00-5:30 pm | Closed Book |
| 2 | Term Project-1 | Take Home | 10 | | Open Book |
| 3 | Term Project-2 | Take Home | 10 | | Open Book |
| | Continuous Class Interaction | | 10 | | Open Book |
| 4 | Design of Lab Experiments | During lab hours | 10 | | Open Book |
| 5 | Comprehensive | 180 | 35 | 07/12 AN | Closed Book |

***(40% evaluation to be completed by mid sem grading)***

Note: The labs for the course will be based on implementing the design aspects of various components of Distributed Operating Systems like distributed middleware, thread synchronizations, logical clock implementations, distributed file systems, and distributed scheduling or load balancing etc. using HDFS/ MapReduce platforms.

**Consultation Hour:** To be announced in the class

**Notices:** Notices regarding the course will be put up on Google Classroom

**Makeup Policy:** No makeup exam allowed without prior permission.

**Academic Honesty and Integrity Policy**: Academic honesty and integrity are to be maintained by all the students throughout the semester and no type of academic dishonesty is acceptable.

**INSTRUCTOR-IN-CHARGE**
**CS G623**