JNTUA COLLEGE OF ENGINEERING (AUTONOMUS): PULIVENDULA
Department of Computer Science & Engineering
B.Tech IV Year I Semester

Subject: Distributed Systems

Lesson plan

Course Code	**:**	15ACS57			
Course Title	**:**	**Distributed Systems**			
Course Structure	**:**	Lectures	Tutorials	Practicals	Credits
		4	0	-	3
Course Coordinator	**:**	D.Raghunath Kumar Babu, Asst.Prof(Adhoc)			
Team of Instructor		Sri G. Murali Asst.Prof & HOD of CSE			

I. Course Overview

The course introduces the main principles underlying distributed systems: processes, communication, naming, synchronization, consistency, fault tolerance, and security. Students will be familiar with some of the main paradigms in distributed systems: object-based systems, file systems, web-based and coordination-based systems. On the completion of the unit, students will understand the fundamentals of distributed computing and be able to design and develop distributed systems and applications.

II. Prerequisite(s):

Level	Credits	Periods / Week	Prerequisites
UG	3	4	Computer organization and Architecture, operating system

III. Assessment:

FORMATIVE ASSESEMENT	
Mid Semester Test I (Theory) for 20 Marks in first two units is conducted at the end of 9th week. Mid Semester Test II (Theory) for 20 Marks in last three units is conducted at the end of the course work.	20 Marks
Multiple Choice Test I for 10 Marks in first two and halfunits is conducted at along with Theory exam. Multiple Choice Test II for 10 Marks in last two and halfunits is conducted at along with Theory exam. Note: After evaluating these tests 30 marks are calculated as 80% from best marks and 20% from other.	10 Marks
Total (Formative)	30 Marks

SUMMATIVE ASSESMENT	
End Semester Examination in all units is conducted for 70 Marks	70 marks
Grand Total	100 Marks

IV. Course objectives:

In Distributed Systems this course, you will learn a range of fundamental and applied techniques in distributed systems. The learning objectives for Distributed Systems are:

1. Apply knowledge of distributed systems techniques and methodologies.
2. Explain the design and development of distributed systems and distributed systems applications.
3. Use the application of fundamental Computer Science methods and algorithms in the development of distributed systems and distributed systems applications.
4. Discuss the design and testing of a large software system, and to be able to communicate that design to others.

V. Course Outcomes:

- The student will explain various architectures used to design distributed systems, such as client-server and peer-to-peer.

- The student will build distributed systems using various interprocess communication techniques, such as remote method invocation, remote events, and tuple spaces.

- The student will build distributed systems using various techniques for tolerating partial failures, such as leasing and replication.

- The student will build distributed systems using various interprocess coordination techniques, such as distributed mutual exclusion, distributed monitors, and tuple spaces.

- The student will explain various distributed algorithms, such as logical clocks and leader election.

- The student will analyze and explain current distributed systems research literature.

VI. Program outcomes:

a An ability to apply knowledge of computing, mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems to real-world problems (fundamental engineering analysis skills)

b An ability to design and conduct experiments, as well as to analyze and interpret data (information retrieval skills)

c An ability to design , implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints such as economic, environmental, social, political, health and safety, manufacturability, and sustainability (Creative Skills) d An ability to function effectively on multi-disciplinary teams (team work)

e An ability to analyze a problem, identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution (engineering problem solving skills)

f An understanding of professional, ethical, legal, security and social issues and responsibilities (professional integrity)

g An ability to communicate effectively both in writing and orally (speaking / writing skills)

h The broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society (engineering impact assessment skills)

i Recognition of the need for, and an ability to engage in continuing professional development and life-long learning (continuing education awareness)

j A Knowledge of contemporary issues (social awareness)

k An ability to use current techniques, skills, and tools necessary for computing and engineering practice (practical engineering analysis skills)

l An ability to apply design and development principles in the construction of software and hardware systems of varying complexity (software hardware interface)

m An ability to recognize the importance of professional development by pursuing postgraduate studies or face competitive examinations that offer challenging and rewarding careers in computing (successful career and immediate employment).

VII. Syllabus:

UNIT I Basic Concepts

Definition of a distributed systems, Examples, Resource sharing and the Web, Challenges, System models, Architectural and fundamental models, Networking Interprocess communication, External data representation and marshalling, Client-server and Group communication.

UNIT II Communication in Distributed System

System Model – Inter process Communication – the API for internet protocols – External data representation and Multicast communication. Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation And Objects: Remote Invocation – Introduction – Request-reply protocols – Remote procedure call – Remote method invocation. Case study: Java RMI – Group communication – Publish-subscribe systems – Message queues – Shared memory approaches -Distributed objects – Case study: Enterprise Java Beans -from objects to components

UNIT III Peer to Peer Services and File System

Peer-to-peer Systems – Introduction – Napster and its legacy – Peer-to-peer – Middleware – Routing overlays. Overlay case studies: Pastry, Tapestry- Distributed File Systems –Introduction – File service architecture – Andrew File system. File System: Features-File model -File accessing models – File sharing semantics Naming: Identifiers, Addresses, Name Resolution – Name Space Implementation – Name Caches – LDAP.

UNIT IV Synchronization and Replication

Introduction – Clocks, events and process states – Synchronizing physical clocks- Logical time and logical clocks – Global states – Coordination and Agreement – Introduction – Distributed mutual exclusion – Elections – Transactions and Concurrency Control– Transactions -Nested transactions – Locks – Optimistic concurrency control – Timestamp ordering – Atomic Commit protocols -Distributed deadlocks – Replication – Case study – Coda.

UNIT V Process& Resource Management

Process Management: Process Migration: Features, Mechanism – Threads: Models, Issues, Implementation. Resource Management: Introduction- Features of Scheduling Algorithms –Task Assignment Approach.

TEXT BOOK:

1. George Coulouris, Jean Dollimore and Tim Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education, 2012.

REFERENCES:

1. Pradeep K Sinha, “Distributed Operating Systems: Concepts and Design”, Prentice Hall of India, 2007.

2. Tanenbaum A.S., Van Steen M., “Distributed Systems: Principles and Paradigms”, Pearson Education, 2007.

3. Liu M.L., “Distributed Computing, Principles and Applications”, Pearson Education, 2004.

4. Nancy A Lynch, “Distributed Algorithms”, Morgan Kaufman Publishers, USA, 2003.

IX. Course Plan:

The course plan is meant as a guideline. There may probably be changes.

Lecture No.	Course Learning Outcomes	Topics to be covered	Reference
3/07/2018	Students should understand the essential ingredients of distributed systems and how to build distributed systems that are resilient to transient network failures and other potential anomalies.	**UNIT-I** Definition of a distributed systems, Examples	T1:4.1, T1:4.2 R2:02p
3/07/2018	Learning about the various challenges in distributed systems.	Resource sharing and the Web, Challenges	T1:4.3 R2:3-16
05/07/2018 10/07/2018	Designing of models are used for interfacing from one process to other process.	System models, Architectural and fundamental models,	T1:4.4 R2:36-52
12/07/2018	Ipc system is help for networked based communication.	Networking Interprocess communication,	T1:3.1,T1:3.2,3.4, R1:2.6
17/07/2018	Understanding the data representation and marshalling concepts.	External data representation and marshalling	T1:3.3,3.4, R2:10.2, R4:8.13
19/07/2018	Learning about client server and peer communications.	Client-server and Group communication	R2:5.5,5.6
24/07/2018	Gain Knowledge on IPC and API internet protocols	. **UNIT-II:** Model – Inter process Communication – the API for internet protocols	R2:5.7
26/07/2018	Students gain the knowledge on data Representation and Multi cast Communication.	External data representation and Multicast communication	T1:8.5, W2.1, R1:2.5, R4:7.4
31/07/2018	Understand on Overlay networks and RMI and objects	Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation And Objects	T1:8.6 T1:8.7
02/08/2018	Learning about request reply RPC systems	: Remote Invocation – Introduction – Request-reply protocols – Remote procedure call –objects to components	T1:10.2, R4: 4.5, 4.6,
07/08/2018	Gain the knowledge on enterprise java beans and RMI	Remote method invocation. Case study: Java RMI –Case study: Enterprise Java Beans - from	T2:10.3,10.4, R1:6.4, 6.5, 6.6, R3:4.7, 4.9
09/08/2018	Understand the group communication procedure systems	Group communication – Publish-subscribe systems –	T1: 10.5,10.6, R1:6.7.1, R3:4.11,4.12,4.13
14/08/2018	Sharing of messages between the shared memory and objects	Message queues – Shared memory approaches -Distributed objects –	R2:4.1, T1:4.2, R1:7.1.1,
16/08/2018	Knowledge gain on peer to peer	**UNIT-III:** Peer-to-peer Systems –	T1:4.3, R1:7.3

	systems	Introduction	
28/08/2018	Know the Middleware and routing overlays	– Napster and its legacy – Peer-to-peer – Middleware – Routing overlays.	R2:4.4
30/08/2018	Gathering the knowledge on DFS and Tapestry and their case studies	Overlay case studies: Pastry, Tapestry- Distributed File Systems –	R2:4.5,4.6
4/09/2018	Ability to know the file service architecture and Andrew system	Introduction – File service architecture – Andrew File system.	T1:7.1, R1:7.4, R4:5.4
6/09/2018	Gathering about file model and also accessing models	File System: Features-File model -File accessing models	T1:7.2
11/09/2018	Knowing the name semantics and address,Identifiers	– File sharing semantics Naming: Identifiers, Addresses,	T1:7.3,7.4, R1:7.5, R4:5.5
13/09/2018	Students Gain knowledge on Name Resolution ,Caches and LDAP	Name Resolution – Name Space Implementation – Name Caches – LDAP.	T1:12.1,12.2, R1:5.2, 5.3, 5.4, R4:5.3
18/09/2018	Ability to know the Clocks ,events and also their process states	UNIT-IV: Introduction – Clocks, events and process states – Synchronizing physical clocks-	T1:12.3,12.4, R1:5.9, R4: 6.7,7
20/09/2018	Knowing about the synchronization clocks		T1:12.5,12.6, w4.1, w4.2, R1: 5.5, 5.6, 5.7, R3:18.4, R4:10.5,10.6,10.7, 10.8, R6:5.5,6.8
25/09/2018	Students get clarity and logical time and clocks and their coordination	Logical time and logical clocks – Global states – Coordination and Agreement –	T1:11.2, R1:10.1,10.2, R4:11, R6:8
27/09/2018	Knowing the knowledge on Distributed mutual exclusion and properties	Introduction – Distributed mutual exclusion – Elections –	T1:11.3,11.4,R1:10.3
04/10/2018	Gathering knowledge on Transactions and Concurrency Control & transactions	Transactions and Concurrency Control– Transactions -	T1:11.5,11.6, R1:4.4
09/10/2018	Learn for nested transactions ,lock, concurrency controls	Nested transactions – Locks – Optimistic concurrency control –	R2:9.1,9.2, R1:12.1,8.1.2, R3:12.5, R3:2.23, R4:10.6
11/10/2018	Students gain knowledge on protocols,deadlocks and replications factor	Timestamp ordering – Atomic Commit protocols -Distributed deadlocks – Replication – Case study – Coda.	T1:9.3,9.4, R2:1:8.2,8.3,8.4
16/10/2018	Knowledge gain on process and its migration,features	UNIT-V: Process Management: Process Migration: Features,	T1:13.1,13.2, R1:12.2, 12.4, 12.7, R5:13.4
18/10/2018	Learn to threads and models,issues	Mechanism – Threads: Models, Issues,	T1:13.3

		Implementation.	
23/10/2018 25/10/2018	Students gain the knowledge on resource management ,algorithms	Resource Management: Introduction-Features of Scheduling Algorithms –Task Assignment Approach.	T1:13.4

X. Mapping course outcomes leading to the achievement of the programme outcomes:

Course	**Program Outcomes**											
Outcomes	**a**	**b**	**C**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
1	**H**		**S**									
2		**H**			**S**							
3					**S**	**H**						

S= Supportive
H=Highly Related

Justification of Course syllabus covering Course Outcomes:

By covering the syllabus a student can understand the architecture, Inter process Communication,group,Marshalling,Replication,Transactions,Process working status can be learned and useful for all in feature.

Justification of CO's –PO's Mapping Table:

By mapping CO-1 to the PO's E which are related to the course CO1: The student is able to analyze the Distributed systems

By mapping CO-2 to the PO's A, which are related to the course CO2: The student is able to interface between the process through network Communication.

By mapping CO-3 to the PO's C & M which are related to the course CO3: The student is able to understand the purpose of Andrew file system & LDAP.

By mapping CO-4 to the PO's E which are related to the course CO4: The student is able to understand the Purpose of peer to peer processing system.

By mapping CO-5 to the PO's K & M which are related to the course CO5: The student is able to understand the Purpose of clocks and transactions.

By mapping CO-6 to the PO's E which are related to the course CO6: The student is able to understand the concept of time stamp ordering and Deadlocks.

By mapping CO-7 to the PO’s E which are related to the course CO7: The student is able to differentiate the deadlock methods and its mutual exclusions.

By mapping CO-8 to the PO’s C which are related to the course CO8: The student is able to understand Scope and Life time of Time stamp ordering and deadlocks.

By mapping CO-9 to the PO’s K which are related to the course CO9: The student is able to understand the purpose of process and threads and its applications.

By mapping CO-10 to the PO’s I & M which are related to the course CO10: The student is able to develop applications using distributed architectures ,client server & peep-peer communication and