CIT303 – Operating systems and Applications

Course Name	Operating Systems and Applications
Course Code	CIT303
Course Type	Compulsory
Course Level	Undergraduate
AKTS Credit	5 ECTS
Course hours per week (Institutional)	3
Practice hours per week	2
Laboratory hours per week	-
Academic Semester	2013 -2014 Fall
Course coordinator(s)	Dr. Hüseyin Lort
Instruction system	
Medium language	English
Prerequisite	-
Suggestions related to course	N/A
Training required	N/A
Aim of the course	The major goals of this course are: • Introduce Concepts of modern operating systems and classification. • Describe Von Neumann architecture and the operating system's structure and differences of operating sytems like Windows, Linux, Unix. • Present Unix operating system ,working with directories, Introduction to Unix shell. • Explain concept of a process, algorithms for deadlock detection and avoidance and explain the conditions that lead to deadlock. • Introduce The concepts of physical memory and virtual memory management, describe the processor scheduling policies and disk scheduling techniques. • Describe basic concepts of modern operating systems, their installation, use and management.
Learning outcomes	At the end of this course students should: 1. Understand the basic concepts of modern operating systems, know about history of operating systems and describe the role and purpose of operating systems 2. Understand the Von Neumann architecture and the operating system's structure. 3. Uderstanding the differences of operating sytems like Windows, Linux, Unix, MAC OS. 4. Understand the concept of a process and list the various process state. 5. Learn the algorithms for deadlock detection and avoidance and explain the conditions that lead to deadlock. 6. Understand the concept of how programming languages, operating

	systems, and hardware architectures interact. 7. Explain the concepts of physical memory and virtual memory management. 8. Describe the processor scheduling policies,Understand the concept of disk scheduling techniques. 9. Understand the basic concepts of modern operating systems, their installation, use and management.	
Course Content		

	Week	**Topics**	
		Theory	**Practice**
Course content per week	1-2	**Introduction:** Definition of operating systems, history of operating systems and classifications of operating systems (multiuser, multi-processing, multitasking, multithreading, real-time, distributed).	Understanding of objectives of operating systems, features of operating systems, detailed review of classification of operating systems and exercises.
	3	Von Neumann architecture and the operating system's structure. differences of operating systems	Overview of computer systems and structure **Assignment1**
	4	**Processor Utilization:** Uniprogramming and Multiprogramming. Resource Utilization, Windows Overview, Windows Architecture, Disk Operating System and Operating System Organization.	Understanding of Process Utilization, Microsoft Window Overview. Practice on MS DOS commands
	5	**Introduction to UNIX:** Introduction to basic UNIX commands, UNIX File System, Working with directories, Introduction to Unix shell, Writing and executing simple shell scripts.	Interfacing with UNIX and understanding important shell commands
	6-7	**Deadlock:** Categories of resources, Resource allocation graphs, Conditions for deadlock, Prevention occurrence of a deadlock, Banker's algorithm, Deadlock avoidance, Deadlock detection and recovery.	Review, exercises and Problems solving about deadlock. A**ssignment2**
	8	**Midterm**	
	9	**Memory Managements:** Memory management requirement, Memory partitioning, Dynamic memory partitioning algorithms, Buddy system, Reallocation, Paging, Segmentation.	Solving problems about memory partitions.
	10	**Virtual Memory:** Characteristics of paging and segmentation, Locality and virtual memory, Virtual memory paging, virtual memory segmentation, Combined Paging and Segmentation, Basic Page Replacement Algorithms, Windows memory management.	Solving problems about paging and segmentations. **Assignment3**

	12-13	**Processor Scheduling and Disk Scheduling:** Types of Processor Scheduling, Scheduling algorithms, Traditional UNIX Scheduling, Disk performance parameters, Disk scheduling policies.	Introduction to the UNIX file system and continue to Unix file system.
	14	**Windows and Linux operating systems**: installation, use and management.	Understanding of how to install windows and how to use Linux operating system
	15	**Final exam**	
Course book and references :	William Stallings, Operating Systems, Internal and Design Principles, Fifth Edition, Pearson Prentice-Hall, 2005. ***Resource Books:*** 1. Andrew S. Tanenbaum*, Modern Operating Systems*, Second Edition, Pearson Prentice-Hall, 2001. 2. Ann McIver McHoes and Ida M. Flynn, *Understanding Operating Systems*, Fifth Edition, Thomson, 2008. 3. William S. Davis and T. M. Rajkumar, *Operating Systems*, *A Systematic View*, Sixth Edition, Addison Wesley, 2004.		

Evaluation

Quizzes: 30%
Midterm exam: 30%
Final exam: 40%

Semester Activities	**Number**	**Contribution percentage to course mark %**
Midterm Exam	**1**	**30**
Quizzes	**2**	**30**
Final Exam	**1**	**40**
TOTAL		**100**

3 Theory Hour X 12 + 1 Practice Hour X 12 + 1 hour midterm + 2 hour final + 4 hours quizzes+ 4 hours X 12 studing + 3 hours X 5 assignments + 20 hours research in library= 150/30 = 5 ECTS credit

Programme and learning outcomes

Learning Outcomes (LO)	Programme Outcomes (PO)																
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PO 13	PO 14	PO 15	PO 16	PO 17
LO1	4								5								
LO2	2				2				3								
LO3	5								5								
LO4			5						5								
LO5				3					5								
LO6	3								5	4							
LO7	2								5								
LO8	3								5	2							
LO9	4		3						5								

*Contribution Level:
1 very low **2** low **3** medium **4** high **5** very high

CITT Department Programme Outcomes

1. Having adequate level of knowledge and skills in current/new computing and educational technologies.
2. Having sufficient communication and teaching skills in teaching profession.
3. Being able to teach updated computing technologies efficiently in English.
4. Being able to identify information technology problems through using various analysis and synthesis.
5. Being pragmatic to develop and apply persistent information technology solutions to educational and business problems.
6. Being able to use critical and computational thinking skills to produce alternative solutions at every level of project development life-cycle.
7. Being capable to work in disciplinary and interdisciplinary teamwork.
8. Being sensitive, reactive and responsive to professional, social and ethical issues. Having social and ethical awareness in teaching and in providing solutions to problems.
9. Having adequate level of knowledge and skills in current/new computer hardware, operating systems and computer networks.
10. Adequate level of knowledge and skills in current/new programming languages, programming paradigms (procedural and object-oriented) and programming environments (visual, console-based programming).
11. Being able to analyse, plan and manage educational software design and project development.
12. Having the capability of evaluating and criticising educational software design and development.

13. Adequate level of knowledge in using and integrating current/new e-learning and distance education systems such as learning management systems (LMS).
14. Having sufficient skills and knowledge in using instructional technology and material design.
15. Having skills to apply and use special teaching approaches, theories, teaching strategies, methods and techniques (such as to those people with disabilities).
16. Using appropriate measurement and evaluation techniques to assess students' learning and development in addition to supporting them with good level of feedback.

CITT Department Programme Outcomes

10. Having adequate level of knowledge and skills in current/new computing and educational technologies.
11. Having sufficient communication and teaching skills in teaching profession.
12. Being able to teach updated computing technologies efficiently in English.
13. Being able to identify information technology problems through using various analysis and synthesis.
14. Being pragmatic to develop and apply persistent information technology solutions to educational and business problems.
15. Being able to use critical and computational thinking skills to produce alternative solutions at every level of project development life-cycle.
16. Being capable to work in disciplinary and interdisciplinary teamwork.
17. Being sensitive, reactive and responsive to professional, social and ethical issues. Having social and ethical awareness in teaching and in providing solutions to problems.
18. Having adequate level of knowledge and skills in current/new computer hardware, operating systems and computer networks.
10. Adequate level of knowledge and skills in current/new programming languages, programming paradigms (procedural and object-oriented) and programming environments (visual, console-based programming).
11. Being able to analyse, plan and manage educational software design and project development.
12. Having the capability of evaluating and criticising educational software design and development.
13. Adequate level of knowledge in using and integrating current/new e-learning and distance education systems such as learning management systems (LMS).
14. Having sufficient skills and knowledge in using instructional technology and material design.
15. Having skills to apply and use special teaching approaches, theories, teaching strategies, methods and techniques (such as to those people with disabilities).
16. Using appropriate measurement and evaluation techniques to assess students' learning and development in addition to supporting them with good level of feedback.