CSL 331	SYSTEM SOFTWARE AND MICROPROCESSORS LAB	Category	L	T	P	Credit	Year of Introduction
		PCC	0	0	4	2	2019

Preamble: The aim of this course is to give hands-on experience in how microcontrollers, and microprocessors can be programmed. The course also aims to enable students to design and implement system software. The student should get familiar with assembly level programming of microprocessors and microcontrollers, interfacing of devices to microcontrollers, resource allocation algorithms in operating systems and design and implementation of system software.

Prerequisite: Sound knowledge in Operating systems

CO1	Develop 8086 programs and execute it using a microprocessor kit. **(Cognitive Knowledge Level: Apply) .**
CO2	Develop 8086 programs and, debug and execute it using MASM assemblers **(Cognitive Knowledge Level: Apply)**
CO3	Develop and execute programs to interface stepper motor, 8255, 8279 and digital to analog converters with 8086 trainer kit **(Cognitive Knowledge Level: Apply)**
CO4	Implement and execute different scheduling and paging algorithms in OS **(Cognitive Knowledge Level: Apply)**
CO5	Design and implement assemblers, Loaders and macroprocessors. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO 5	PO 6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓				✓		✓		✓
CO2	✓	✓	✓	✓				✓		✓		✓
CO3	✓	✓	✓	✓				✓		✓		✓
CO4	✓	✓	✓	✓				✓		✓		✓
CO5	✓	✓	✓	✓				✓		✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Test (Internal Exam)** ***Percentage***	**End Semester Examination** ***Percentage***
Remember	**20**	**20**
Understand	**20**	**20**
Apply	**60**	**60**
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**75**	**75**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : **15 marks**
Continuous Evaluation in Lab : **30 marks**
Continuous Assessment Test : **15 marks**
Viva-voce : **15 marks**

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Operating System to Use in Lab **:** Linux
Compiler/Software to Use in Lab **:** gcc
Programming Language to Use in Lab **:** Ansi C
Any compatible assembler can be used for implementation of 8086 programs

Fair Lab Record:

All Students attending the System Software and Microprocessors Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Details of Experiment including algorithm and Result of Experiment. The left hand page should contain a print out of the code used for the experiment and sample output obtained for a set of input.

Syllabus

MICROPROCESSOR LAB

I. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit
II. Exercises/Experiments using MASM (PC required)
III. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language programming
IV. Exercises/Experiments using 8051 trainer kit

SYSTEM SOFTWARE LAB:

I. Experiments related to the operating system.
II. Exercises/Experiments related to the assemblers, loaders and macroprocessors

Text Books

1. Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.
2. Andrew S Tanenbaum, "Modern Operating Systems" , 4th Edition, Prentice Hall, 2015.
3. Leland L. Beck, System Software: An Introduction to Systems Programming, 3/E, Pearson Education Asia, 1997.

Reference Books

1. A. NagoorKani, Microprocessors and Microcontrollers, Second Edition, Tata McGraw Hill
2. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education.
3. William Stallings, "Operating systems", 6th Edition, Pearson, Global Edition, 2015.
4. Garry Nutt, Nabendu Chaki, Sarmistha Neogy, "Operating Systems", 3rd Edition, Pearson Education.
5. D.M. Dhamdhere, Systems Programming and Operating Systems, Second Revised Edition, Tata McGraw Hill.

Practice Questions

MICROPROCESSORS LAB : List of Exercises/ Experiments
(Minimum 10 Exercises (at least 2 questions from each part I, II, III & IV)) : 2 Hrs/week

I. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit

1. Implementation of simple decimal arithmetic and bit manipulation operations.
2. Implementation of code conversion between BCD, Binary, Hexadecimal and ASCII.
3. Implementation of searching and sorting of 16-bit numbers.

II. Exercises/Experiments using MASM (PC Required)

4. Study of Assembler and Debugging commands.
5. Implementation of decimal arithmetic (16 and 32 bit) operations.
6. Implementation of String manipulations.
7. Implementation of searching and sorting of 16-bit numbers.

III. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language Programming

8. Interfacing with stepper motor - Rotate through any given sequence.
9. Interfacing with 8255 (mode0 and mode1 only).
10. Interfacing with 8279 (Rolling message, 2 key lockout and N-key rollover implementation).

11. Interfacing with Digital-to-Analog Converter.

IV. Exercises/Experiments using 8051 trainer kit

12. Familiarization of 8051 trainer kit by executing simple Assembly Language programs such as decimal arithmetic and bit manipulation.
13. Implementation of Timer programming (in mode1).

SYSTEM SOFTWARE LAB: List of Exercises/ Experiments
(Minimum 8 Exercises (at least 3 and 5 questions from each part V and VI)) : 2 Hrs/week

V. Exercises/Experiments from operating system

1. Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.
   a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority
2. Simulate the following file allocation strategies.
   a) Sequential b) Indexed c) Linked
3. Implement the different paging techniques of memory management.
4. Simulate the following file organization techniques
   a) Single level directory b) Two level directory c) Hierarchical
5. Implement the banker's algorithm for deadlock avoidance.
6. Simulate the following disk scheduling algorithms.
   a) FCFS b) SCAN c) C-SCAN
7. Simulate the following page replacement algorithms:
   a)FIFO b)LRU c) LFU

VI. Exercises/Experiments from assemblers, loaders and macroprocessor

1. Implement pass one of a two pass assembler.
2. Implement pass two of a two pass assembler.
3. Implement a single pass assembler.
4. Implement a two pass macro processor
5. Implement a single pass macro processor.
6. Implement an absolute loader.
7. Implement a relocating loader