**SCHOOL OF COMPUTER ENGINEERING**

**KIIT DEEMED TO BE UNIVERSITY**

**CS-2092**

**Operating Systems Laboratory**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecture:** | 0 | **Internal Assessment Marks:** | 60 |
| **Tutorial:** | 0 | **End Term Marks:** | 40 |
| **Practical:** | 2 Hrs/Week | **Credit:** | 1 |

**Software Requirements**

* GCC compiler
* Unix platform (Currently used Ubuntu12.04 LTS)

**Hardware Requirements**

* Processor: 386 and above
* RAM: 32 MB and above

**Course objectives**

The objective of this lab is to give an idea about different components of operating system and their interactions happened in an operating system. Specifically, in this course, we mostly focus on the UNIX operating system, where student can get the practical understanding about how to manage the processes with respect to process creation, process communication process synchronization, threading, manage file, etc.

**Course outcomes**

* Able to understand the functionality of different types of modern operating systems and their internals.
* Able to understand the usability and implementation of shell scripting in UNIX platform.
* Able to create multiple processes in UNIX platform, understand different system calls related to process, and analyze the performance of different scheduling algorithms by implementing in UNIX platform.
* Able to understand the concepts of inter process communication and implement it using pipe in multi-programming operating system.
* Able to understand the concepts of deadlock and implementation of deadlock avoidance algorithm in multi-programming system.
* Able to understand the concepts of thread and its implementation using POSIX library.
* Able to understand the synchronization concept by implementing semaphore and monitor using the existing library provided in the LINUX platform.

**Mapping of CO to PO**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **COs POs** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** |
| Able to understand the functionality of different types of modern operating systems and their internals. | H | H | M | M | M |  |  |  |  | M | M |  |
| Able to understand the usability and implementation of shell scripting in UNIX platform. | H | H | H | H | H |  | M |  | M | M | M |  |
| Able to create multiple processes in UNIX platform, understand different system calls related to process, and analyze the performance of different scheduling algorithms by implementing in UNIX platform. | M | H | H | H | H |  |  |  | M | H | H |  |
| Able to understand the concepts of inter process communication and implement it using pipe in multiprogramming operating system. | H | H | H | H | H | H |  |  | M | M | H |  |
| Able to understand the concepts of deadlock and implementation of deadlock avoidance algorithm in multiprogramming system. | H | H | H | H | H | H |  |  | H | H | M |  |
| Able to understand the concepts of thread and its implementation using POSIX (P\_thread) thread in multiprogramming system. | H | M | H | H | M |  |  |  |  |  | M | M |
| Able to understand the synchronization concept by implementing semaphore and monitor using the existing library provided in the LINUX platform. | H | H | H | H | M |  |  |  |  |  | M | M |

**List of Sample experiments**

Lab 1:

* UNIX commands
* How to write and execute a shell program
* Expression evaluation and command line arguments
* Assignments
  + Verify different UNIX commands
  + Write a shell program to add two numbers using command line argument.
  + Write a shell program to find whether a given year is a leap year or not.
  + Write a shell script to extract the data from the date command, and display the result in different format.(if the date command gives “Thu Jan 2 14:21:54 IST 2014” then display the result in “2/Jan/2014/14.21”)

Lab 2:

* logical operator, relational operator used in bash shell
* How to write down the conditional statements in bash shell using logical and relational operator?
* Different control statements (if, elif, case, while, for)
* Assignments
  + Write a shell program to find whether a given year is a leap year or not.
  + Write a shell program to display the prime number between 1 and hundred.
  + Execute the “ps -ef” and “ps -ux” command. Write a shell program that takes the output of these two commands and display the detail about a process whose parent id is 2. The detail about a process will be shown as it is shown during the execution of “ps -ux” command.
  + Write a shell script to extract the data from the date command, and display the result in numerical form with 12 hour format. (if the date command gives “Thu Jan 2 14:21:54 IST 2014” then display the result in “02/01/2014/2:21:54 PM”)
  + Let a directory present in the home directory called “XYZ” that consists of few files and directories. Write a shell script to move all the files present in the “XYZ” directory to a subdirectory called “MyFile” and all the subdirectories present in the “XYZ” directory to a subdirectory called “MyDir”.

Lab 3:

* Different Loop statements ( while, for, until)
* Assignments
  + Write a shell script to reverse a given integer.
  + Write a shell script to verify whether the given string is a palindrome or not.
  + Write a shell program which takes maximum 8 integer type arguments through command line and do the following operation:
    - If the first argument/last result (a) is divisible by send argument (b) then new result=a/b
    - Else If (a%b != 0) and b is divisible by 5 then new result=a\*b
    - Else if (a>b) then new result=a-b
    - Else new result=a+b

Lab 4:

* Implementation of array in shell script.
* Assignments
  + Write a shell script that will take 3 file names as command line argument and concatenate first two files line by line and store the result in third file.
  + Write a shell script that will store all the files in different directory named as x\_1, x\_2,… where maximum size of each directory is X.

Lab 5:

* Discuss a generalized way to design the CPU scheduling algorithm
* Assignments
  + FCFS algorithm implementation
* Home assignments
  + SJF(non-preemptive) algorithm implementation

Lab 6:

* Discuss a generalized way to implement preemptive CPU scheduling algorithms
* Assignments
  + Preemptive SJF algorithm implementation
* Home assignments
  + Round robin algorithm implementation

Lab 7:

* Discuss basics of round robin algorithm implementation
* How to create a processes using fork system call
* Use of wait function
* Assignments
  + Create different process tree using fork system call
  + Write a program where a process waits for all its child processes

Lab 8:

* Discuss about process communication using pipe
  + how to create pipe between two processes
  + What is the use dup function? How it can be utilized in pipe line commands
* Assignments
  + Write a program that creates n level process structure where each level consists of exactly one process. Here every parent process makes some modification to the original message and sends it to its child and this process repeats till it reached at the last child in the process tree.
* Home assignment
  + Write a program to create a ring topology among n number of process and message has to be passed using the pipe in clock wise or anti clock wise direction.

Lab 9:

* Discuss the thread creation using pthread library
* Assignments
  + Write a program to create two threads where one thread adds half of the element from the beginning of the array and another thread add half of the element from the end of the array. And at the end main process shows the sum of all the elements in the given array.
  + Write a program to implement the merge sort using multiple threads

Lab 10:

* Discuss different functions related to Mutex lock using POSIX library
* Discuss different functions related to conditional variable using POSIX library
* Assignment
  + Write a program for reader-writer problem using lock.

Lab 11:

* Discuss different functions related to semaphore using POSIX library
* Assignment
  + Write a program to implement critical section using semaphore

Lab 12:

* Producer consumer problem using semaphore
* Producer consumer problem using mutex lock
* Dining philosophers problem using semaphore
* Dining philosophers problem using mutex lock

**\*Quiz and Test:**

**Guidelines**

1. *Students should be regular and come prepared for each laboratory class.*
2. *Students should bring their lab record and practice note books to every class.*
3. *The prescribed text and reference books and class notes can be kept ready for reference if required.*
4. *Students have to complete their lab experiments in the lab and be capable to explain and show the modifications, output results as and when required by the Faculty/Lab Programmers/Teaching Assistants responsible for that lab. All these are to ensure that the students’ capability is built up to understand, debug and modify codes as per the requirements.*
5. *In case a student misses a class, it is his/her responsibility to complete the missed experiment(s).*
6. *The code written by the student should meet the following:*
7. *Program should have proper input prompt messages (****Menu Driven****) and descriptive output.*
8. *Input validation should be done (data type, range error etc.) and give appropriate error messages and suggest corrective actions.*
9. *Comment lines should be used to give Students Name, Roll No, Branch, Section, Subject, Problem Statement, describe functions, Expected Input & Output and all the key Logic.*
10. *Program should be indented properly. Variables and functions should be meaningfully named. [USE CODING STANDARDS].*
11. *Try to optimize the code wherever possible.*
12. *All programs should be verified by different values and lengthy inputs. [with minimum VALIDATIONS]*
13. *Once the experiment(s) get executed, they should show the program and results to the Faculty/Lab Programmers/Teaching Assistants responsible for that lab. and copy the same in their observation book and get signed which ensures him/her with day to day performance marks*
14. *Student should submit his/her record by/in the next lab session to ensure the lab record marks gets evaluated. Failing to do so reduces the lab record marks.*
15. *Lab examination questions need not necessarily be limited to the questions in the laboratory manual, rather it could include some variations and/or combinations of questions.*
16. *Students are strictly advised to take care of their personal belonging all the time. The University is not responsible and liable to any lost of personal belonging in the computer labs.*

**Scheme of Evaluation**

Maximum marks for Lab is **100**, which is divided into Continuous Internal Assessment and **End-Sem** Final Evaluations.

|  |  |
| --- | --- |
| Internal Assessment (Continuous evaluation over the semester): | **60 Marks** |
| End-Tem Evaluation (At the End of the Semester): | **40 Marks** |
| Minimum Marks for passing in the Lab: | **50 Marks** |