

## CS501T : Operating Systems and Introduction to Linux (Theory)

### Objective:

The main objective of this paper is to introduce the students to a layer of software called the Operating System, whose job is to manage all the devices of a computer system and provide user programs with a simple interface to the hardware. This paper will familiarize the students with the concepts of processes, memory management, file management, Input/Output management and the potential problem of deadlocks.

| Minimum Class Hours | Exam Time | Marks    |          |       |
|---------------------|-----------|----------|----------|-------|
| 70                  | 2         | External | Internal | Total |
|                     |           | 56       | 19       | 75    |

| Unit  | Topic                     | Number of Class Hours | Marks |
|-------|---------------------------|-----------------------|-------|
| I     | Concepts & Processes      | 14                    | 12    |
| II    | Deadlocks                 | 14                    | 11    |
| III   | Memory Management         | 14                    | 11    |
| IV    | File Systems and I/O      | 16                    | 11    |
| V     | Linux & Shell Programming | 12                    | 11    |
| Total |                           | 70                    | 56    |

### Detailed Syllabus

#### Unit I: Concepts & Processes

**14 hours**

Operating System objectives and functions, Booting the computer, Operating System concepts (Processes, Address Spaces, Files, I/O, The Shell), System Calls for Process Management, Types of Operating Systems (Mainframe Operating Systems, Multiprocessor Operating systems, Personal Computer Operating systems, Embedded Operating systems, Real Time Operating systems), Introduction to Processes (The Process Model, Process Creation, Process Termination, Process Hierarchies, Process States, Implementation of Processes), Interprocess Communication (Race conditions, Critical Regions, Mutual Exclusion with Busy Waiting (Lock variables, Peterson Solution, TSL instruction), Sleep and Wakeup, Semaphores, Message Passing), Process Scheduling (First come First serve, Shortest Job First, Priority Scheduling, Priority Scheduling, Multiple Queues, Round Robin Scheduling), Classical IPC problems (The Dining Philosophers Problem, The Sleeping Barber Problem).

#### Unit II: Deadlocks

**14 hours**

Resources, Introduction to Deadlocks (Conditions for Deadlock, Deadlock modeling), Deadlock detection and recovery (Deadlock detection with one resource of one type, Deadlock detection with multiple resources of each type, Recovery from Deadlock), Deadlock avoidance (Resource trajectories, Safe and unsafe states, the Banker's algorithm for a single resource), Deadlock prevention (attacking the mutual exclusion condition, attacking the hold and wait condition, attacking the no-preemption condition, attacking the circular-wait condition).

#### Unit III: Memory Management

**14 hours**

Memory Management without Abstraction, A Memory Abstraction(address spaces, Swapping, Managing Free Memory), Virtual Memory (Paging, Page Tables), Page Replacement Algorithms (Not-recently-used, First in first out, Second Chance page replacement algorithm, Least Recently used page replacement algorithm), Design issues for Paging Systems (Page size, shared pages, Mapped Files), Implementation issues(Operating System involvement with Paging, Page Fault Handling, Locking pages in Memory), Segmentation (Implementation of pure segmentation, Segmentation with Paging: MULTICS).

**Unit IV: File Systems and I/O Management****16 hours**

Files (File Naming, File structure, File types, File access, File attributes, File operations), Directories (Hierarchical directory systems, Path names, Directory operations), File System Implementation (File system Layout, Implementing Files, Implementing directories, Shared files, Disk space management, File system Backups, File system Consistency, File system Performance)

Principles of I/O hardware (I/O devices, Device Controllers, Direct memory access), Goals of the I/O software, I/O software Layers (Interrupt Handlers, Device Drivers, Device independent I/O software, User space I/O software), Disks (Disk hardware (Magnetic disks, RAID), disk arm scheduling algorithms, Error handling), Clocks (Clock hardware, Clock software).

**Unit V: Introduction to Linux and Shell Programming****12 hours**

File Systems and Architecture; The Shell (sh command, Pattern Matching, escaping, quoting, Redirection, /dev/null, /dev/tty, Pipes, command substitution, shell variables)

Simple Filters (pr, head, tail, cut, paste, sort, uniq, tr)\

**Instruction for Paper Setter**

| Unit         | Questions |           |
|--------------|-----------|-----------|
|              | To Set    | To Answer |
| I            | 2         | 1         |
| II           | 2         | 1         |
| III          | 2         | 1         |
| IV           | 2         | 1         |
| V            | 2         | 1         |
| <b>Total</b> | <b>10</b> | <b>5</b>  |

**Recommended Books****Text books:**

1. **Andrew S Tanenbaum**, *Modern Operating Systems*, (Third Edition), Pearson Education.

**Reference books**

1. **William Stallings**, *Operating Systems: Internals and Design Principles*, (Sixth Ed.), Pearson Education, New Delhi.
2. **D.M Dhamdhare**, *Operating systems – A concept based approach*, Second Edition, Tata McGraw Hill 2002
3. **Abraham Silberschatz and Peter B. Galvin**, *Operating System Concepts*, Fourth edition, Addison-Wesley.

## CS501P : Operating Systes and Introduction to Linux (Practical)

### Objective:

The paper aim at helping the students to learn the various Unix/Linux commands while working in the Unix/Linux environment. The paper also aims in teaching the students to write shell script programs to perform some computations and handling the various unix commands through scripts. This is a practical-oriented paper.

| Minimum Class Hours | Exam Time | Marks    |          |       |
|---------------------|-----------|----------|----------|-------|
| 30                  | 3         | External | Internal | Total |
|                     |           | 30       | 10       | 40    |

| Unit  | Topic   | Number of Class Hours | Marks |
|-------|---|-----------------------|-------|
| I     | Understanding unix commands, general purpose utilities, navigating the file system  | 06                    | 19    |
| II    | Handling Ordinary Files, Basic File Attributes, More File Attributes, The Vi editor | 06                    |       |
| III   | The Shell, Simple Filters,  | 06                    |       |
| IV    | Regular Expressions ,The Process  | 06                    |       |
| V     | Shell Programming   | 06                    |       |
| Total |   | 30                    | 19    |

### Detailed Syllabus

#### Unit I

**06 hours**

**Understanding Unix Commands** (Internal and External Commands, Arguments, Options and Filenames, Flexibility of Command Usage (Combining Commands & Splitting the Command Line), man pages(Navigating & Searching, brief description of Sections)

**General-Purpose Utilities:** cal, date, calendar, who, tty, uname, passwd, lock, echo, tput, bc, script, spell and ispell

**Navigating the File System** (The Three types of Files, Parent-Child relationship, pwd, absolute pathnames, Using absolute pathname for a command, cd, mkdir, rmdir, ls, relative pathnames)

#### Unit II

**06 hours**

**Handling Ordinary Files** (cat, cp, rm, mv, more, lp, file, wc, od, split, cmp, comm, diff)

The ls Command (ls -l, -d, -x, -a, -R option)

**Basic File Attributes** (File permissions, chmod)

**More File Attributes** (File systems, inodes, chown, listing by modification and access times, touch, ln)

**The Vi Editor** (Three modes, Input Mode, Saving text and quitting, Command mode, the repeat factor, deletion, navigation, pattern search, joining lines, repeating the last command, undoing last editing instructions, search and replace)

#### Unit III

**06 hours**

**The Shell** (sh command, Pattern Matching, escaping, quoting, Redirection, /dev/null, /dev/tty, Pipes, command substitution, shell variables)

**Simple Filters** (pr, head, tail, cut, paste, sort, uniq, tr)

**Unit IV****06 hours****Regular expressions** (grep, Regular expression, egrep)**The Process** (The Shell process, parents and children, ps, system processes, mechanism of process creation, running jobs in background, kill, nice)**Unit V****06 hours****Shell Programming** (Shell scripts, read, command line arguments, exit, logical operators, if, case, expr, sleep and wait, while, until, for, \$@)**Recommended Book:****1. Yashvant P. Kanetkar**, *Unix Shell Programming*, BPB Publications**Reference Books:****1. Sumitabha Das**, *UNIX Concepts & Applications*, (Second Ed.), Tata McGraw Hill, 2001**2. Newham Cameron**, *Learning the Bash Shell*, (Third Ed), O’Rielly**Instructions for Paper Setter: This paper is purely practical-oriented**

| Unit              | To be set | To be Answered | Marks |
|-------------------|-----------|----------------|-------|
| <b>I, II, III</b> | 3         | 2              | 12    |
| <b>IV, V</b>      | 2         | 1              | 07    |

**Sample Shell Programs**

- Write a script that will accept two file names from the command line, copy the first file to the second file and then display the contents of the combined file. Proper error message should be displayed in case the copy is not successful.
- Write a script that will read a filename from the command line and will change the name of the file to filename.aal where aal is the login\_name of the user. (E.g. if the filename is Lucky and the user’s login\_name is harry then, the filename will be changed to Lucky.harry).
- Peter’s basic salary is input through the keyboard. His dearness allowance is 40% of his basic salary, and house rent allowance is 18% of the basic salary. Write a script to calculate his gross salary.
- The distance between two cities (in km.) is input through the keyboard. Write a script to convert and print this distance in meters, feet, inches and centimeters.
- The length and breadth of a rectangle and radius of a circle are input through the keyboard. Write a script to calculate the area and perimeter of the rectangle as well as the area and circumference of the circle.
- If a five digit number is input through the keyboard, write a script to calculate the sum of its digits.
- Write a script that accepts a string inputted through the keyboard and echoes a suitable message if it does not have at least 10 characters.
- In a company, an employee is paid as follows: If his basic salary is less than Rs. 5000, then HRA = 10% of basic salary and DA = 90% of basic. If his salary is either equal or above Rs. 5000, then HRA = Rs. 900 and DA = 98% of basic salary. If the employee’s salary is input through the keyboard, write a script to find his gross salary.
- Write a script that will accept a filename from the keyboard and determine whether the file exists. If the file exists then its contents will be displayed else an error message will be displayed.
- Write a script that accepts two directory names as arguments and deletes those files in the second directory that are identical to the files in the first.
- The marks obtained by a student in five different subjects are input through the keyboard. The student gets a division as per the following rules:

- i. Percentage above or equal to 60 – First division
- ii. Percentage between 50 and 59 – Second division
- iii. Percentage between 40 and 49 – Third division
- iv. Percentage less than 40 – Fail

Write a script to find the division obtained by the student.

12. Write a shell script that will prompt the user to enter a character. The script will then determine whether the user entered a lowercase letter, an uppercase letter, a digit or a special symbol.
13. If the cost price and selling price of an item is input through the keyboard, write a script to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred.
14. An integer is input through the keyboard. Write a script to find out whether it is an odd or even number.
15. Write a shell script, which receives any year from the keyboard and determines whether the year is leap or not. If no argument is supplied the current year should be assumed.
16. Write a shell script that will display all the multiples of 5 between 5 and 100.
17. Write a shell script that will display the multiplication table of any given number.
18. Write a shell script to find the factorial of any number entered through the keyboard.
19. Write a script to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs. 12 per hour for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour.
20. Write a script that converts a decimal number to a hexadecimal number. [Hint: use bc]
21. Two numbers are entered through the keyboard. Write a script to find the value of one number raised to the power of another.
22. Write a script to print all prime numbers between 1 and 150.
23. Write a shell script that displays a list of all files in the current directory to which you have read, write and execute permissions.
24. Write a shell script that will receive any number of filenames as arguments. The shell script should check whether every argument supplied is a file or a directory. If it is a directory it should be appropriately reported. If it is a filename then name of the file as well as the number of lines present in it should be reported.
25. Write a script that will receive any number of filenames as arguments. The script should check whether such files already exist. If they do, then it should be reported. If these files do not exist then check if a sub-directory called *mydir* exists in the current directory. If it doesn't exist then it should be created and in it the files supplied as arguments should get created. If *mydir* already exists then it should be reported along with the number of files that are currently present in *mydir*.
26. Write a script that accepts a filename as argument and displays the last modification time if the file exists, and a suitable message if it doesn't.
27. Write shell script to convert file names from UPPERCASE to lowercase file names or vice versa.
28. Write a shell script that backs up all files in a directory into a Backup directory for every day of the week. In other words, on Monday all files go in a "Monday" or "1" backup directory, on Tuesday they all go into a "Tuesday" directory, and so forth. If a directory for today already exists, overwrite the files in it, otherwise create the directory.
29. Write a program that checks if any of a list of users given on the command line is logged in. For each user it should say whether he/she is logged in or not.

\*\*\*\*\*