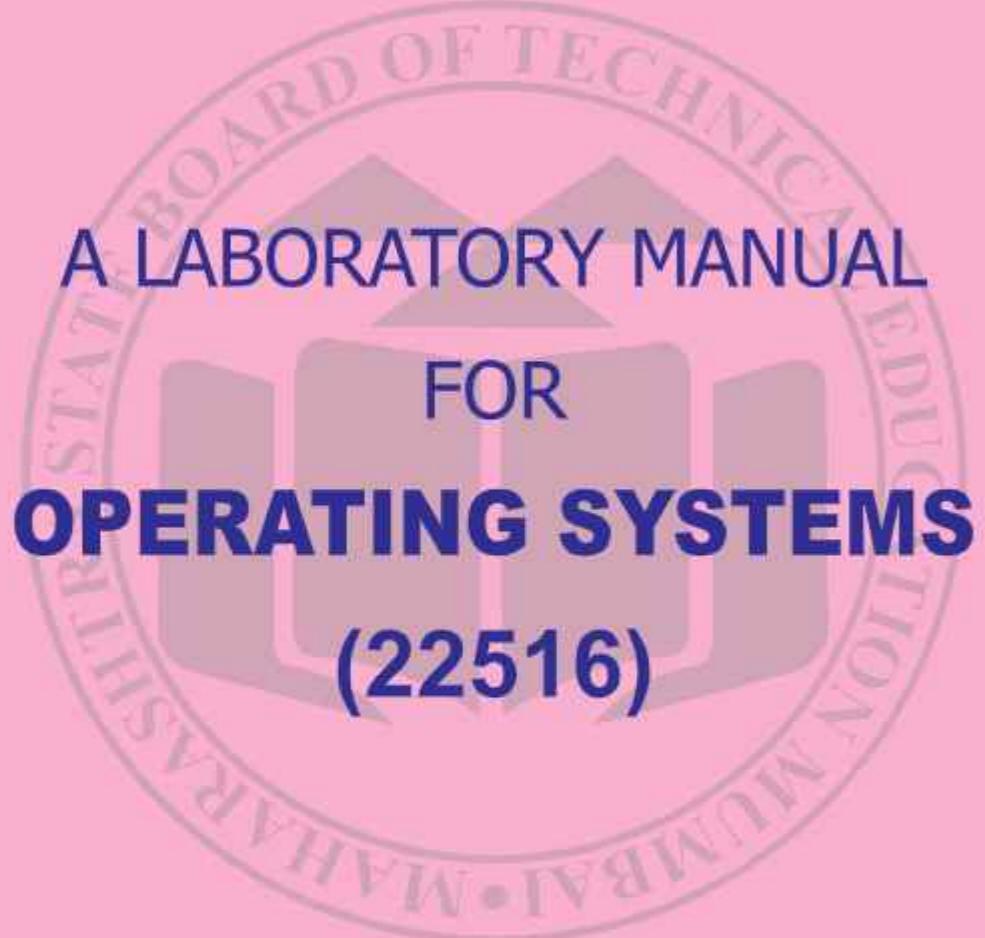


I

Name \_\_\_\_\_  
Roll No. \_\_\_\_\_ Year 20 \_\_\_\_ 20 \_\_\_\_  
Exam Seat No. \_\_\_\_\_

COMPUTER GROUP | SEMESTER - V | DIPLOMA IN ENGINEERING AND TECHNOLOGY



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI  
(Autonomous) (ISO 9001 : 2015) (ISO / IEC 27001 : 2013)

## **VISION**

To ensure that the Diploma level Technical Education constantly matches the latest requirements of technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

## **MISSION**

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the changing technological and environmental challenges.

## **QUALITY POLICY**

We, at MSBTE are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

## **CORE VALUES**

MSBTE believes in the followings:

- Education industry produces live products.
- Market requirements do not wait for curriculum changes.
- Question paper is the reflector of academic standards of educational organization.
- Well designed curriculum needs effective implementation too.
- Competency based curriculum is the backbone of need based program.
- Technical skills do need support of life skills.
- Best teachers are the national assets.
- Effective teaching learning process is impossible without learning resources.

**A Laboratory Manual**

**for**

**Operating System**

**(22516)**

**Semester-V**

**(CO/CW/CM/IF)**



**Maharashtra State**

**Board of Technical Education, Mumbai**

**(Autonomous) (ISO:9001:2015) (ISO/IEC 27001:2013)**



**Maharashtra State Board of Technical Education,  
(Autonomous) (ISO:9001: 2015) (ISO/IEC 27001 : 2013)  
4th Floor, Government Polytechnic Building, 49, Kherwadi,  
Bandra ( East ), Mumbai - 400051.**

**(Printed on May,2019)**



# Maharashtra State Board of Technical Education

## Certificate

This is to certify that Mr. / Ms .....  
Roll No.....of ..... Fourth Semester of  
Diploma in .....  
of Institute.....  
(Code.....) has attained pre-defined practical  
outcomes(POs) satisfactorily in course **Operating System**  
**(22516)** for the academic year 20.....to 20..... as prescribed in  
the curriculum.

Place .....  
Date:.....

Enrollment No.....  
Exam Seat No. ....

Course Teacher

Head of the Department

Principal





## Preface

The primary focus of any engineering laboratory/field work in the technical education system is to develop the much needed industry relevant competencies and skills. With this in view, MSBTE embarked on this innovative ‘I’ Scheme curricula for engineering Diploma programmes with outcome-based education as the focus and accordingly, relatively large amount of time is allotted for the practical work. This displays the great importance of laboratory work making each teacher, instructor and student to realize that every minute of the laboratory time need to be effectively utilized to develop these outcomes, rather than doing other mundane activities. Therefore, for the successful implementation of this outcome-based curriculum, every practical has been designed to serve as a ‘*vehicle*’ to develop this industry identified competency in every student. The practical skills are difficult to develop through ‘chalk and duster’ activity in the classroom situation. Accordingly, the ‘I’ scheme laboratory manual development team designed the practical’s to *focus* on *outcomes*, rather than the traditional age old practice of conducting practical’s to ‘verify the theory’ (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the pre-determined outcomes. It is expected from each student that at least a day in advance, they have to thoroughly read the concerned practical procedure that they will do the next day and understand minimum theoretical background associated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. Students will then become aware about the skills they will achieve through procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students.

Operating systems are an essential part of any computer system. Similarly, a course on operating systems is an essential part of any computer group. We hope that students will also find it useful. It provides a clear description of practical performance, execution and working of Operating System.

Although all care has been taken to check for mistakes in this laboratory manual, yet it is impossible to claim perfection especially as this is the first edition. Any such errors and suggestions for improvement can be brought to our notice and are highly welcome.

## **Programme Outcomes (POs) to be achieved through Practicals of this Course:**

Following programme outcomes are expected to be achieved significantly out of the ten programme outcomes and Computer Engineering and Information Technology programme specific outcomes through the practicals of the course on **Operating System**.

**PO 1. Basic knowledge:** Apply knowledge of basic mathematics, science and basic engineering to solve the problems related to application of computers and communication services in storing, manipulating and transmitting data, often in the context of a business or other enterprise.

**PO 2. Discipline knowledge:** Apply Information Technology knowledge to solve broad-based Information Technology related problems.

**PO 3. Experiments and practice:** Plan to perform experiments, practices and to use the results to solve Information Technology related problems.

**PO 4. Engineering tools:** Apply appropriate Information Technology related techniques/tools with an understanding of the limitations.

**PO 5. The engineer and society:** Assess societal, health, safety and legal issues and the consequent responsibilities relevant to practice in the field of Information technology.

**PO 6. Environment and sustainability:** Apply Information Technology related engineering solutions for sustainable development practices in environmental contexts.

**PO 7. Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of practice in the field of Information Technology.

**PO 8. Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

**PO 9. Communication:** Communicate effectively in oral and written form.

**PO 10. Life-long learning:** Engage in independent and life-long learning along with the technological changes in the IT and allied industry.

**Practical - Course Outcome Matrix**

| S.<br>No. | Title of the Practical   | CO<br>a. | CO<br>b. | CO<br>c. | CO<br>d. | CO<br>e. | CO<br>f. |
|-----------|--|----------|----------|----------|----------|----------|----------|
| 1         | Install and configure Linux (or alike) operating system.   | ✓        | -        | -        | -        | -        | -        |
| 2         | Execute general purpose commands date, time, cal, clear, banner, tty, script, man.   | ✓        | -        | -        | -        | -        | -        |
| 3         | Work with multiple linux terminals and basic commands: who, who am i, login, passwd, su, pwd.  | -        | ✓        | -        | -        | -        | -        |
| 4         | a) Use Operating services (Editor, GUI, File handling.)<br>b) Run commands to start, stop, and restart the specified service in Linux.         | -        | ✓        | -        | -        | -        | -        |
| 5         | Execute process commands- ps, wait, sleep, exit, kill.   | -        | -        | ✓        | -        | -        | -        |
| 6         | Execute file and directory manipulation commands – ls, rm, mv, cp, join, split, cat (file saving and redirection operator), head, tail, touch. | -        | -        | -        | -        | -        | ✓        |
| 7         | Execute file and directory manipulation commands – diff, comm., pr, chmod, mkdir, rmdir, cd, pwd, dir, cmp. (Use wild card character).         | -        | -        | -        | -        | -        | ✓        |
| 8         | Execute text processing tr, wc, cut, paste, spell, sort, grep, more.   | -        | -        | -        | -        | -        | ✓        |
| 9         | Use vi editor and perform all editor commands.   | -        | -        | -        | -        | -        | ✓        |
| 10        | Write and execute Shell Script by using following Control statements features- “if” statement  | -        | -        | -        | -        | -        | ✓        |
| 11        | Write and execute Shell Script by using following Control statements features- “for” statement, exit, break, continue                          | -        | -        | -        | -        | -        | ✓        |
| 12        | Write Shell script to find out whether - Given file exists?  | -        | -        | -        | -        | -        | ✓        |
| 13        | Write Shell script to find out whether - File has read, write, and execute permissions?  | -        | -        | -        | -        | -        | ✓        |
| 14        | Write a program to calculate total waiting and turnaround time of n processes with First Come First Serve CPU scheduling algorithm.            | -        | -        | -        | ✓        | -        | -        |
| 15        | Write a ‘C’ program to implement FIFO page replacement Algorithm.  | -        | -        | -        | -        | ✓        | -        |

## **List of Industry Relevant Skills**

The following industry relevant skills of the competency “Manage operations of Operating System” are expected to be developed in you by performing practicals of this laboratory manual.

1. Executes basic commands to work with Operating System like UNIX.
2. Execute Shell Scripts.
3. Execute program on Process Scheduling.
4. Execute programs on Page Replacement

## **Brief Guidelines to Teachers**

### **Hints regarding strategies to be used:**

1. Teacher shall explain prior concepts to the students before starting each experiment.
2. For practical's requiring tools to be used, teacher should provide the demonstration of the practical emphasizing the skills, which the student should achieve.
3. Involve students in the activities during the conduct of each experiment.
4. Teachers should give opportunity to students for hands-on after the demonstration.
5. Assess the skill achievement of the students and COs of each unit.
6. Teacher is expected to share the skills and competencies to be developed in the students.
7. Teacher should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
8. Teacher may provide additional knowledge and skills to the students even though that may not be covered in the manual but are expected from the students by the industries.
9. Teacher may suggest the students to refer additional related literature of the reference books/websites/seminar proceedings etc.
10. During assessment teacher is expected to ask questions to the students to tap their knowledge and skill related to that practical.

## **Instructions for Students**

Student shall read the points given below for understanding the theoretical concepts and practical applications.

1. Students shall listen carefully the lecture given by teacher about importance of subject, learning structure, course outcomes.
2. Students shall organize the work in the group of two or three members and make a record of all observations.
3. Students shall understand the purpose of experiment and its practical implementation.
4. Students shall write the answers of the questions during practical.
5. Student should feel free to discuss any difficulty faced during the conduct of practical.
6. Students shall develop knowledge of Operating System fundamental and manipulation skills as expected by the industries.
7. Student shall attempt to develop related hands on skills and gain confidence.
8. Students shall refer technical magazines; websites related to the scope of the subjects and update their knowledge and skills.
9. Students shall develop self-learning techniques.
10. Students should develop habit to submit the write-ups on the scheduled dates and time.

**Content Page**  
**List of Practicals and Progressive Assessment Sheet**

| Sr. No.            | Title of the practical   | Page No. | Date of performance | Date of submission | Assessment marks(50) | Dated sign. of teacher | Remarks (if any) |
|--------------------|--|----------|---------------------|--------------------|----------------------|------------------------|------------------|
| 1.                 | Install and configure Linux (or alike) operating system.   | 01       |                     |                    |                      |                        |                  |
| 2.                 | Execute general purpose commands date, time, cal, clear, banner, tty, script, man.   | 05       |                     |                    |                      |                        |                  |
| 3.                 | Work with multiple linux terminals and basic commands: who, who am i, login, passwd, su, pwd.  | 10       |                     |                    |                      |                        |                  |
| 4.                 | a) Use Operating services(Editor, GUI, File handling.)<br>b) Run commands to start, stop, and restart the specified service in Linux.          | 14       |                     |                    |                      |                        |                  |
| 5.                 | Execute process commands- ps, wait, sleep, exit, kill.   | 19       |                     |                    |                      |                        |                  |
| 6.                 | Execute file and directory manipulation commands – ls, rm, mv, cp, join, split, cat (file saving and redirection operator), head, tail, touch. | 24       |                     |                    |                      |                        |                  |
| 7.                 | Execute file and directory manipulation commands – diff, comm., pr, chmod, mkdir, rmdir, cd, pwd, dir, cmp. (Use wild card character).         | 33       |                     |                    |                      |                        |                  |
| 8.                 | Execute text processing tr, wc, cut, paste, spell, sort, grep, more.   | 44       |                     |                    |                      |                        |                  |
| 9.                 | Use vi editor and perform all editor commands.   | 53       |                     |                    |                      |                        |                  |
| 10.                | Write and execute Shell Script by using following Control statements features- ‘if’ statement  | 63       |                     |                    |                      |                        |                  |
| 11.                | Write and execute Shell Script by using following Control statements features- ‘for’ statement, exit, break, continue                          | 69       |                     |                    |                      |                        |                  |
| 12.                | Write Shell script to find out whether - Given file exists?  | 76       |                     |                    |                      |                        |                  |
| 13.                | Write Shell script to find out whether -File has read, write, and execute permissions?   | 81       |                     |                    |                      |                        |                  |
| 14.                | Write a program to calculate total waiting and turnaround time of n processes with First Come First Serve CPU scheduling algorithm.            | 86       |                     |                    |                      |                        |                  |
| 15.                | Write a ‘C’ program to implement FIFO page replacement Algorithm.  | 92       |                     |                    |                      |                        |                  |
| <b>Total Marks</b> |  |          |                     |                    |                      |                        |                  |

- To be transferred to Proforma of CIAAN-2017.

## **Practical No. 1: Install and configure Linux (or alike) operating system.**

### **I. Practical Significance:**

Linux is the base of many of open source operating systems designed to replace Windows and Mac OS. It is free to download and install on any computer. Because it is open source, there are a variety of different versions, or distributions, available developed by different groups. To prevent hacking attempts, many organizations keep their Linux operating systems private. Many others make their variations of Linux available publicly so the whole world can benefit at large.

### **II. Relevant Program Outcomes (POs)**

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

Manage operations of Operating System.

The practical is expected to develop the following skills:

Able to install operating system and configure it.

### **IV. Relevant Course Outcome(s)**

Install operating system and configure it.

### **V. Practical Outcome (PrOs)**

Install and configure Linux (or alike) operating system.

### **VI. Relevant Affective domain related Outcome(s)**

1. Follow precautionary measures.
2. Follow installation steps.
3. Follow ethical practices.

### **VII. Minimum Theoretical Background**

Step 1: Download the ISO file.

Step 2: Boot your system with Bootable DVD / USB drive.  
To start the installation click on "Install Ubuntu"

Step 3: Check Install Prerequisite

Step 4: Select the Installation Type

Step 5: Select your respective Time Zone

Step 6: Select your respective Keyboard Layout

Step 7: Set the Hostname of your system and User credentials that will be used after installation.

Installation has started. Once the installation is completed, it will ask to restart the Machine. Click on "Restart Now"

**Step 8: Login Screen after reboot.**

Use the same user and its credentials that we have set during the installation.

We will get below screen after entering the credentials.

Ubuntu Installation is Completed Now.

Similarly any open source installation shall be considered.

### VIII. Resources required (Additional)-Nil

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM- 2GB and above (As per need of OS) | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

### IX. Resources used (Additional)

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

### X. Program Code: Teacher must assign a separate program statement to students.

Install and configure Linux (or similar) operating system on your computer. Write down the steps for same.

## **XI. Result (Output of installation):**

.....  
.....  
.....

## XII. Practical Related Questions

**Note:** Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What are different versions of Linux operating system?
  2. Enlist the steps for booting the operating system.
  3. State names of latest multiuser operating system and its advantages.

(Space for answer)

**XIII. Exercise:**

1. Differentiate between command line OS and GUI OS by giving example.
2. Draw the diagram of multiprogramming system and state concept of it.
3. Which are the extra facilities provided by Unix other than Windows OS?
4. Enlist four features of the following operating system:
  - a. Windows 98
  - b. Windows 2000
  - c. Windows XP

**XIV. References / Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| <b>Performance Indicators</b>     |                            | <b>Weightage</b> |
|-----------------------------------|----------------------------|------------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>       |
| 1.                                | Logic formation            | 10%              |
| 2.                                | Debugging ability          | 10%              |
| 3.                                | Follow ethical practices   | 10%              |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>       |
| 4.                                | Expected output            | 30%              |
| 5.                                | Timely Submission          | 30%              |
| 6.                                | Answer to sample questions | 10%              |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b>      |

*List of Students /Team Members*

1. .....
2. .....
3. .....

| <b>Marks Obtained</b>      |                            |                  | <b>Dated signature of Teacher</b> |
|----------------------------|----------------------------|------------------|-----------------------------------|
| <b>Process Related(15)</b> | <b>Product Related(35)</b> | <b>Total(50)</b> |                                   |
|                            |                            |                  |                                   |

## Practical No. 2: Execute general purpose commands.

### I. Practical Significance:

General purpose commands are inbuilt programs that can be invoked in multiple ways. These commands work interactively from a terminal. A terminal that provides a command line interface using a shell program.

### II. Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System.

The practical is expected to develop the following skills:

1. Able to apply general purpose commands.

### IV. Relevant Course Outcome(s)

Use operating system tools to perform various functions.

### V. Practical Outcome (PrOs)

Execute general purpose commands date, time, cal, clear, banner, tty, script, man.

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

| Sr. No. | Commands     | Meaning   |
|---------|--------------|---|
| 1       | \$date       | It displays system date and time  |
| 2       | \$cal        | Displays calendar for current month.  |
| 3       | \$clear      | Clears the screen   |
| 4       | \$banner OSY | Displays an argument string as a poster with maximum of ten characters per line |
| 5       | \$tty        | Name of terminal  |
| 6       | \$script     | Record login session  |
| 7       | \$man        | It gives manual help for any command.   |

**Specifying the date format**

( + is used to concatenate date command with descriptor)

| <b>Descriptor</b> | <b>Example</b>                      | <b>Meaning</b>     | <b>Example</b> |
|-------------------|-------------------------------------|--------------------|----------------|
| %y                | \$date "+%y"                        | Year( 2 Digits)    | 19             |
| %b                | \$date "+%b"                        | Month abbreviation | Jan            |
| %m                | \$date "+%m"                        | Month digit        | 01             |
| %d                | \$date "+%d/%m/%y"                  | Day of month       | 01 to 31       |
| %j                | \$date "+Today is %jth day of year" | Day of year        | 1 to 366       |

**VIII. Resources required (Additional)**

| <b>Sr. No.</b> | <b>Name of Resource</b> | <b>Broad Specification</b>                                | <b>Quantity</b>   | <b>Remarks (If any)</b> |
|----------------|-------------------------|---|-------------------|-------------------------|
| 1              | Computer System         | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments     |
| 2.             | Operating System        | Unix/Linux/Ubuntu/any other open sources operating system |                   |                         |

**IX. Resources used (Additional)**

| <b>Sr. No.</b> | <b>Name of Resource</b> | <b>Broad Specification</b> | <b>Quantity</b> | <b>Remarks (If any)</b> |
|----------------|-------------------------|----------------------------|-----------------|-------------------------|
| 1              | Computer System         |                            |                 |                         |
| 2.             | Operating System        |                            |                 |                         |

**X. Program Code: Teacher must assign a separate command statements to students.**

1. Write down different options of cal command. (Use \$man cal)
2. Write options of date command. (Use \$man date)

**XI. Result (Output of Commands):**

---

---

---

---

**XII. Practical Related Questions**

**Note:** Below given are few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. How you record all the following activities performed by the user.
2. Give a command to display calendar for month of January.
3. Give single statement command to display the calendar of previous, current and next month.
4. Give the command to display full week day (eg.Sunday) using date command.

**(Space for Answer)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

### XIII. Exercise:

1. What is output of following commands?
    - a. \$cal 04 2019  
.....
    - b. \$date "+Today's information: %D and %B"; cal  
.....
    - c. \$date "+My clock is showing %H hours. %M minutes and %S seconds"  
.....
    - d. \$cal -3
    - e. \$cal -5
    - f. \$cal -2000

(Attach page for output)
  2. Give the syntax of commands for displaying the output. (use date)
    - i. This is ..... Month of the year ..... (Abbreviation for month and 4-digit year)
    - ii. This is the .....'th day of this year.

(Space for answer)

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| <b>Performance Indicators</b>     |                            | <b>Weightage</b> |
|-----------------------------------|----------------------------|------------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>       |
| 1.                                | Logic formation            | 10%              |
| 2.                                | Debugging ability          | 10%              |
| 3.                                | Follow ethical practices   | 10%              |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>       |
| 4.                                | Expected output            | 30%              |
| 5.                                | Timely Submission          | 30%              |
| 6.                                | Answer to sample questions | 10%              |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b>      |

*List of Students /Team Members*

1. .....
2. .....
3. .....

| <b>Marks Obtained</b>      |                            |                  | <b>Dated signature of Teacher</b> |
|----------------------------|----------------------------|------------------|-----------------------------------|
| <b>Process Related(15)</b> | <b>Product Related(35)</b> | <b>Total(50)</b> |                                   |
|                            |                            |                  |                                   |

## Practical No. 03: Work with multiple Linux terminals and basic commands.

### I      Practical Significance:

Work with terminal environment to know about users and set their security. It includes current status of all the users like details about all the users who accesses the terminal for particular duration and path of directory.

### II     Relevant Program Outcomes (POs)

- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III    Competency and Practical skills

Manage operations of Operating System.

The practical is expected to develop the following skills:

1. Able to apply general purpose commands. (user related)

### IV     Relevant Course Outcome(s)

Use operating system tools to perform various functions.

### V      Practical Outcome (PrOs)

Work with multiple Linux terminals and basic commands: who, who am i, login, passwd, su, pwd.

### VI     Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII    Minimum Theoretical Background

| Sr. No. | Commands                 | Syntax            | Description   |
|---------|--------------------------|-------------------|---|
| 1       | who                      | \$who             | It is used to display who are the users connected to our computer currently |
| 2       | who am i                 | \$who am i        | Display the details of the current working directory                        |
| 3       | login                    | login: \$username | Prompt, enter username  |
| 4       | passwd                   | \$passwd uname    | Sets password for users   |
| 5       | su(sudo)                 | \$su ls           | Provides super user privileges  |
| 6       | pwd(Present Working Dir) | \$pwd             | To print the complete path of the current working directory                 |

**VIII Resources required (Additional)**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM- 2GB and above (As per need of OS) | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X Program Code: Teacher must assign a separate command to students.**

List down with all options for **who** commands and write its description.

Note: Write terminal's output in Result Section.

**XI Result (Output of Command):**

.....

.....

.....

.....

.....

## **XII Practical Related Questions**

**Note:** Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Give command for present working directory.
  2. State currently login users by command.
  3. Acquire the status of super user.

### XIII Exercise:

1. Acquire the status of super user.
  2. Write output for following commands
    - i. \$who;clear;who am i
    - ii.\$who;tty:date

(Space for Answer)

**XIV References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV Assessment Scheme**

| Performance Indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>  |
| 1.                                | Logic formation            | 10%         |
| 2.                                | Debugging ability          | 10%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>  |
| 4.                                | Expected output            | 30%         |
| 5.                                | Timely Submission          | 30%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(15) | Product Related(35) | Total(50) |                            |
|                     |                     |           |                            |

## **Practical No. 4: Working with a) operating system services b) Processes**

### **I. Practical Significance:**

The purpose of computer systems is to allow the user to execute programs. So the Operating systems provides an environment where the user can conveniently run programs. The user does not have to worry about the memory allocation or multitasking or anything. These things are taken care of by the operating systems.

### **II. Relevant Program Outcomes (POs)**

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

Manage operations of Operating System.

The practical is expected to develop the following skills:

- a. Able to work with services like editor, GUI etc.
- b. Able to work with Processes like start, stop etc.

### **IV. Relevant Course Outcome(s)**

Install operating system and configure it.

### **V. Practical Outcome (PrOs)**

- a) Use Operating services (Editor, GUI, File handling.)
- b) Run commands to start, stop, and restart the specified service in Linux.

### **VI. Relevant Affective domain related Outcome(s)**

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### **VII. Minimum Theoretical Background**

An Operating System supplies different kinds of services to both the users and to the programs as well. It also provides application programs (that run within an Operating system) an environment to execute it freely. It provides users the services run various programs in a convenient manner. File editing is done with the help of any editor. The default editor that comes with the UNIX operating system is called vi (visual editor).

**The UNIX vi editor is a full screen editor and has two modes of operation:**

Command mode commands which cause action to be taken on the file, and  
Insert mode in which entered text is inserted into the file.

**Editor:**

There are many ways to edit files in Unix. vi Editor works in Unix. This editor enables you to edit lines in context with other lines in the file. You can also use this editor to just read a text file.

An improved version of the vi editor which is called the VIM has also been made available now. Here, VIM stands for Vi IMproved.

**GUI:**

**Linux** system provides both the interface that is GUI and CLI. As per the requirement you can decide when you should use the Linux command line interface (CLI) and when you should use a graphical user interface (GUI).

**File Handling.****Following are the Processes commands:**

**Command to add and remove services in your system start-up:** use `chkconfig`

**Enter the command to show currently running services.**

`ls /etc/init.d`

**Enter the restart command.**

`sudo systemctl`

**To stop service**

`sudo systemctl`,

**To start service.**

`sudo systemctl`

**VIII. Resources required(Additional)-Nil**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

- X. Program Code:** Teacher must assign a separate program statement to students.  
Check all the permissions started on your system. Stop the services which are not required for long time.

**XI. Result (Output of Command):**

.....  
.....  
.....  
.....

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. List various menus you observed on your system?
2. Study the GUI of your Linux system.
3. What is difference between GUI and CLI?

(Space for answer)

.....  
.....  
.....

**XIII. Exercise:**

1. What are system calls provided by file management?
2. Draw and explain services provided by operating system.
3. What are system component of operating system.

**(Space for Answer)**

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| Performance Indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>  |
| 1.                                | Logic formation            | 10%         |
| 2.                                | Debugging ability          | 10%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>  |
| 4.                                | Expected output            | 30%         |
| 5.                                | Timely Submission          | 30%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(15) | Product Related(35) | Total(50) |                            |
|                     |                     |           |                            |

## Practical No. 5: Execute process commands.

### I. Practical Significance:

Process is program in execution. ps Commands is used to manage and to see the active processes on Unix operating system. ps basically means 'Process Status' which is used to display currently running processes in Unix operating system.

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System.

The practical is expected to develop the following skills:

1. Able to execute process commands like ps, wait, sleep, exit, kill.

### IV. Relevant Course Outcome(s)

Execute process commands for performing process management operations.

### V. Practical Outcome (PrOs)

Execute process commands- ps, wait, sleep, exit, kill.

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

| Sr. No. | Commands   | Meaning  |
|---------|------------|--|
| 1       | \$ps       | Display the characteristics of a process. (ie. terminal number, time required, PID no, and command name) |
| 2       | \$wait     | Waits until all background processes are completed and then exits.                                       |
| 3       | \$sleep 30 | Used to execute commands after certain amount of time by sleeping for given seconds.                     |

|   |          |   |
|---|----------|---|
| 4 | \$exit   | Used to quit the shell.   |
| 5 | \$kill 0 | Used to stop execution of particular process by sending an interrupt signal to the process. |

**Options of ps commands:**

|                  |  |
|------------------|--|
| \$ps -f          | Full listing showing PPID of each process.     |
| \$ps -u username | Displays processes of user 'username'          |
| \$ps -a          | Processes of all users                         |
| \$ps -e          | Processes including user and system processes. |

**Options of kill commands:**

|                    |  |
|--------------------|--|
| \$kill 0           | Kills all the processes on the terminal except the login shell by special argument '0' |
| \$kill 120 230 234 | Kills three processes with pid 120 230 234   |
| \$kill -9 0        | Kills all processes including login shell  |
| \$kill -9 \$\$     | Kills login shell  |

**VIII. Resources required (Additional)-Nil**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate command to students.**

- What is process id of your login shell?
- Give PID of all processes, how you will terminal the processes running on your terminal.
- What is difference between wait and sleep?

**XI. Result (Output of Command):**

---

---

---

---

---

---

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. What is name of your login shell?
2. What are various options of kill commands?
3. What are various options of ps commands?
4. Explain about exit command.
5. List the system calls for process management.

(Space for answer)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIII. Exercise:**

1. Observe the output of following commands:  
    \$sleep 30; date  
    \$echo \$\$
  2. Display full listing of all the processes running on your terminal.
- 
- 
- 
- 
- 
- 
- 
- 

3. Write output of following command.  
    \$sleep 60; banner GOOD
- 
- 
- 
- 
- 
- 
- 
-

4. Write all the process commands.

---



---



---



---



---



---



---

#### XIV. References/ Suggestions for Further Reading

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

#### XV. Assessment Scheme

| Performance Indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>  |
| 1.                                | Logic formation            | 10%         |
| 2.                                | Debugging ability          | 10%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>  |
| 4.                                | Expected output            | 30%         |
| 5.                                | Timely Submission          | 30%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

#### *List of Students /Team Members*

1. ....
2. ....
3. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(35) | Product Related(15) | Total(50) |                            |
|                     |                     |           |                            |

## Practical No. 6: Execute file and directory manipulation commands.

### I. Practical Significance:

Basically, the operations perform in Unix/Linux are done on a prompt / terminal which is not stored anywhere. But most of the commands are written to store the information. One way is to store the information in a file. Various operations that can be performed on a file are create, open, read, write, move, close and rename.

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System.

The Practical is expected to develop the following skills:

1. Able to execute file and directory manipulation commands.

### IV. Relevant Course Outcome(s)

Apply file management techniques.

### V. Practical Outcome (PrOs)

Execute file and directory manipulation commands – ls, rm, mv, cp, join, split, cat (file saving and redirection operator), head, tail, touch.

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

File and directory manipulation commands –

**ls command :** It is used to lists files in the current working directory.

Syntax:

\$ls

| <i>Options</i> | <i>Meaning</i>   |
|----------------|--|
| ls -a          | list all files including hidden file starting with '' . '' .. '' |
| ls -d          | list directories - with '*/'                                     |
| ls -i          | list file's inode index number                                   |
| ls -l          | list with long format - show permissions                         |

**Example:**

|                              |  |
|------------------------------|--|
| <code>\$ls -i abc.txt</code> | <i>list contents of abc.txt file</i>                             |
| <code>\$ls -l</code>         | <i>List of all files along with permissions given to it.</i>     |
| <code>\$ls ???</code>        | List the names of files with exact three characters in its name. |
| <code>\$ls d*s</code>        | List of files names which starts for d and ends with s           |
| <code>\$ls -I</code>         | Gives full listing of all files and directories.                 |

**Output**

```

Terminal File Edit View Search Terminal Help
drwxrwxr-x 2 root root 4096 Jan 4 11:06 cdrom
drwxr-xr-x 16 root root 4120 Feb 12 16:28 dev
drwxr-xr-x 129 root root 12288 Feb 12 16:21 etc
drwxr-xr-x 3 root root 4096 Jan 4 11:11 home
lrwxrwxrwx 1 root root 33 Jan 4 11:16 initrd.img -> boot/
initrd.img-3.19.0-25-generic
drwxr-xr-x 23 root root 4096 Jan 4 11:16 lib
drwx----- 2 root root 16384 Jan 4 10:54 lost+found
drwxr-xr-x 2 root root 4096 Aug 5 2015 media
drwxr-xr-x 2 root root 4096 Apr 11 2014 mnt
drwxr-xr-x 2 root root 4096 Aug 5 2015 opt
dr-xr-xr-x 173 root root 0 Feb 12 16:20 proc
drwx----- 2 root root 4096 Aug 5 2015 root
drwxr-xr-x 23 root root 740 Feb 12 16:21 run
drwxr-xr-x 2 root root 12288 Jan 4 11:17 sbin
drwxr-xr-x 2 root root 4096 Aug 5 2015 srv
dr-xr-xr-x 13 root root 0 Feb 12 16:28 sys
drwxrwxrwt 4 root root 4096 Feb 12 16:25 tmp
drwxr-xr-x 10 root root 4096 Aug 5 2015 usr
drwxr-xr-x 13 root root 4096 Aug 5 2015 var
lrwxrwxrwx 1 root root 30 Jan 4 11:16 vmlinuz -> boot/vml
inuz-3.19.0-25-generic

```

**Figure 1****rm command:** It is used to remove the file**Syntax:**`$rm filename`**Example**`$rm abc.txt` Removes abc.txt`$rm abc.txt xyz.txt` Removes abc.txt and xyz.txt files`$rm *` Removes all files**mv command :** It is used to move a file from one location to another.**Syntax:****mv [Option] source destination**

(consider 3 files having name abc.txt, xyz.txt and pqr.txt.)

To rename the file abc.txt to aaa.txt (not exist):

`$ ls`

abc.txt xyz.txt pqr.txt

`$ mv abc.txt aaa.txt``$ ls`

xyz.txt pqr.txt aaa.txt

**Options:****1. -i (Interactive):** The mv command with -i option ask for confirmation`$ mv -i abc.txt aab.txt`

**cp command :** It is used to copies a file to destination file.  
If the command contains two file names, then it copies the contents of **1st file** to the **2nd** file. If the 2nd file doesn't exist, then first it creates one and content is copied to it. But if it existed then it is simply overwritten without any warning.

**\$ cp Source file Destination file**

**Example:**

```
$ ls  
xyz.txt  
$ cp xyz.txt zzz.txt
```

Check it now

```
$ ls  
xyz.txt zzz.txt
```

**join command:** It is used for joining lines of two files on the basis of common field in each file called as a key.

Syntax:

```
$join [-t -o] filename1 filename2  
-t field separator  
-o Specify the column from each table that must be included.
```

**Example:**

```
$cat student1  
Ajay 1990  
Vijay 1992  
Sujit 1991
```

**\$cat student2**

```
Ajay 50  
Vijay 62  
Sujit 70
```

**\$join -t ":" student1 student2**

```
Ajay : 1990 : 50  
Vijay : 1992 : 62  
Sujit : 1991 : 70
```

**split command:** It is used to split the large file into smaller files. Default size is 1000 lines per file.

Syntax:

**\$split -n filename tag name**

-n            number of lines in each smaller file.

-tag name    by default the split builds the output files named xaa,xab,xac.....

If tag name specified, it replaces the x with that tag name.

**Example:**

**Split -100 student**

Check it now

```
$ls  
xaa xab xac
```

**cat command** : It is used to create the file and displaying the contents of file/files.

Syntax:

\$ cat >sample (To create a file)

This is sample file in Unix.

<ctrl d>

\$ cat sample

where cat is a command and sample is argument. This is sample file in Unix.

User can display contents of more than one file and called as concatenation.

\$ cat list1.txt list2.txt (List two files at a time)

**head command**: It is used to read the first ten lines of file. (10 lines by default)

The number of lines to be displayed may be specified in the head command.

Syntax:

\$head -n filename (-n means how many lines to be displayed)

**Example:**

\$head -20 student

Displays the first 20 lines of the file student.

**tail command**: It is used to print last few numbers of lines (10 lines by default) of a certain file, then terminates.

Syntax:

\$tail {[+/-]n [lbc]}filename

| Options | Meaning                                    |
|---------|--|
| +n      | Displays all lines starting from nth line. |
| -n      | Displays n lines from end of the file      |
| l       | Indicates Lines                            |
| b       | Indicates blocks                           |
| c       | Indicates characters                       |

**Example:**

\$tail -20l student

Displays last 20 lines of student file.

\$tail +20l student

Displays all lines starting from 20<sup>th</sup> line of the student file.

**touch command**: It is the easiest way to create new, empty files.

Syntax:

\$touch filename

Example:

\$touch aaa

**File saving and redirection operator:**

Generally, Unix commands are entered at the keyboard and output resulting from these commands are displayed on the computer screen.

The standard input and standard output are actually taken or saved to files.

Example:

Output of first command is taken as input for second command. In this example output of ls command is saved into a file out.txt.

\$ ls > out.txt

Combine 2 text files to another file:

```
$ cat list1.txt list2.txt > todo.txt
```

In this example contents of files list1.txt and list2.txt is saved to todo.txt file.

```
$cat list1.txt>>list2.txt
```

It appends the contents list1.txt to list2.txt. It adds the contents at the end of file.

Now check it

```
$ls list2.txt
```

## VIII. Resources required (Additional)-Nil

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

## IX. Resources used (Additional)

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

## X. Program Code: Teacher must assign a separate program statement to students.

1. Create three files a1 a2 a3.
2. Apply different commands like ls, mv, cp, rm, join, split and check the list of files at the end.

## XI. Result (Output of Code):

.....  
.....  
.....  
.....

## XII. Practical Related Questions

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. What are different options of ls command? Write down the command along with the option and note down the output. (Use \$man ls command to check options)
2. What are different options of mv command?
3. What is use of split command?
4. How to use join command?

(Space for answer)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

### XIII. Exercise:

- 1 Write output of following commands:
  - i) Display all file names which starts with 'a' and ends with 'y'
  - ii) Enlist all the files beginning with 'm' and ending with any range of 1 to 5.
  - iii) Show the contents of the files whose file names contains exactly two characters.
  - iv) Create a file ABCD.txt, create a copy with XXX.txt. Rename the original file with AACD.txt. Delete the file XXX.txt.
  - v) Display the inodes of any two files at the same time.
- 2 List all file processing commands.
- 3 How many lines will be displayed with head command if number is not specified.
- 4 Create two files chapter1 and chapter2 and perform the following operations
  1. Copy contents of chapter1 to chapter2 by asking the user before overwrite.
  2. Display inodes of two files.
  3. Rename the file 'chapter1' to 'Lesson1'
- 5 Execute the following commands:  
\$ls a\*n  
.....  
\$ls s?  
.....  
\$cat abc>>xz  
.....

**(Space for Answer)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| Performance indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>  |
| 1.                                | Logic formation            | 10%         |
| 2.                                | Debugging ability          | 10%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>  |
| 4.                                | Expected output            | 30%         |
| 5.                                | Timely Submission          | 30%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. .....
2. .....
3. .....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(15) | Product Related(35) | Total(50) |                            |
|                     |                     |           |                            |

## Practical No. 7: Execute file and directory manipulation commands

### I. Practical Significance:

Data and programs are stored in **files**. These are organized in **directories**. In a simple way, a **directory** is just a file that contains other files (or directories).

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System.

The practical is expected to develop the following skills:

1. Able to execute file and directory manipulation commands.

### IV. Relevant Course Outcome(s)

Apply file management techniques.

### V. Practical Outcome (PrOs)

Execute file and directory manipulation commands – diff, comm., pr, chmod, mkdir, rmdir, cd, pwd, dir, cmp. (Use wild card character).

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

Unix uses a hierarchical structure for organizing files and directories. This structure is called as a **directory tree**. The tree has a single root node, the slash character (/), and all other directories are contained below it. When user first log in to the Unix server, the specified directory is called as **Home directory**.

Directory Structure in Unix/Linux:

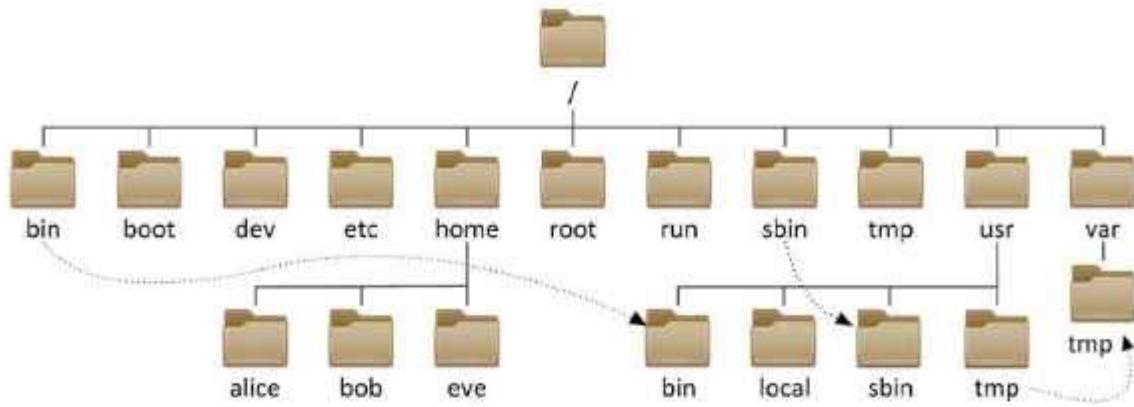


Figure 2

**Directory manipulation commands are as follows:**

1. mkdir
2. cd
3. rmdir
4. pwd

#### 1. ‘mkdir’ command:

It is used to create a new directory in a current directory.

\$mkdir <directory name>

Example:

\$mkdir CM51

User can create more than one directory in a single command.

\$mkdir subject1 subject2

If user wants to create directory on the specific path then syntax is:

**Syntax :**

\$mkdir path/Directory\_name

#### 2. ‘cd’ command:

The ‘cd’ command is used to change directory. You can use it to change to any directory by specifying a valid absolute or relative path.

The syntax is as given below

\$cd <directory name>

##### Example

\$ cd Directory name

\$cd CM51

\$cd . To come out from current working directory.

##### Example:

\$cd ..

\$ cd / It changes to root directory.

##### Example:

\$ cd /

#### 3.’rmdir’ command

It is used to delete/ remove a directory. If the parent directory having subdirectories then first all subdirectories will be deleted then the parent directory is deleted.

\$rmdir <directory name>

#### 4. ‘pwd’ command

pwd stands for **Present Working Directory**. This is most used Linux command to see the specific Unix Directory on which the user is working on.

\$pwd

#### diff command:

This command is used to show difference between two text files. It also tells which line in one has to be changed to make the two files identical.

##### Syntax:

**\$diff filename1 filename2**

The options of the result should be like this –

- a -Added the text to file
- c -Changes are made in the file
- d -Deletion operation is performed
- < -Lines from the first file
- > -Lines from the second file

**\$cat file1.txt**

I need to go to the shop.

I need to buy some mangoes.

When I get home, I'll wash the cat.

**\$cat file2.txt**

I need to go to the shop.

I need to buy some mangoes.

Oh yeah, I also need to buy cheese.

When I get home, I'll wash the cat.

Use the diff command to compare both files.

**\$ diff file1.txt file2.txt**

The above command should give the result as shown below –

2a3

> Oh yeah, I also need to buy cheese.

From the output, 2a3 means “After line 2 in the first file, a line needs to be added: line 3 from the second file”.

#### comm command:

This command compares two sorted files. It compares each line of first file with its corresponding line in the second file and generates three column output.

The first column lists the lines only in first file.

The second column lists the lines only in second file.

The third column lists the lines in both files.

##### Syntax:

**\$comm filename1 filename2**

##### Example:

**\$cat student1**

Harsh

Sujay

Smith

**\$comm student1 student2**

**\$cat student2**

Harsh

Niket

Ashutosh

|                |                   |       |
|----------------|-------------------|-------|
| Sujay<br>Smith | Niket<br>Ashutosh | Harsh |
|----------------|-------------------|-------|

**pr command:****pr - convert text files for printing**

The pr command does minor formatting of files on the terminal screen or for a printer. For example, if you have a long list of names in a file, you can format it onscreen into two or more columns.

The syntax is:

**pr option(s) filename(s)**

pr changes the format of the file only on the screen or on the printed copy; it doesn't modify the original file.

**dir command:**

**It is used to list all files and directories.**

```
sswp1226@ubuntu:~$ dir
32      archive.tar  d3          f      p1      Templates
3241    archive.tar  demo       f1.txt  p2.tar  vaa
3241.tar  asd.tar   Desktop    fh     Pictures  vaaa
41      a.tar       dh        file1   pqr.tar  vaa.tar
-A      awt.tar    Documents  file1.tar Public  vai
abc1    .cvf       Downloads  file2   -r      vaishu
abc1.tar  d       env.txt   m.tar   srt.tar  val.tar
abc2    d1        examples.desktop o.tar   -t      va.tar
abc.tar  d2
sswp1226@ubuntu:~$
```

Figure 3

**chmod command:**

It is used to change or set three permissions (read, write and execute) for all the three categories of users.

Types of users are – user, group and others.

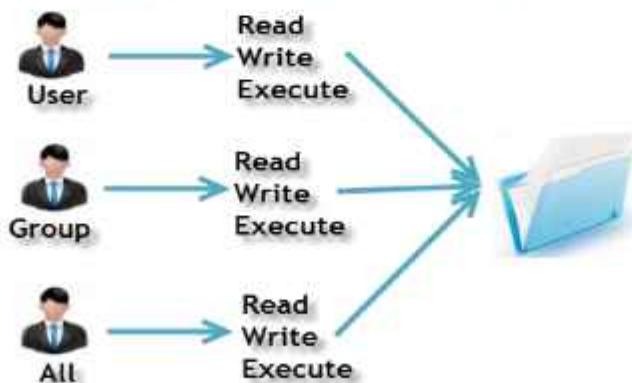
**Owners assigned Permission On Every File and Directory**

Figure 4

To identify the permissions of a file or directory use ls-l command.

**Syntax:**

\$ls -l

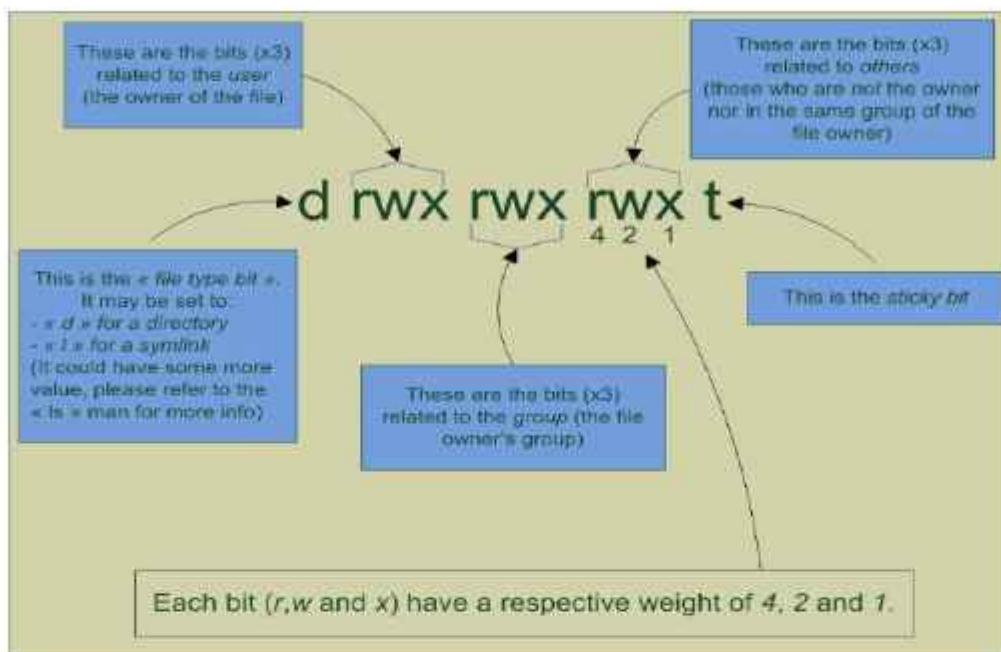


Figure 5

**'chmod'** can be operated in two ways:

- symbolic or alphabetical notation.
- Octal or absolute notation.

### 1. symbolic or alphabetical notation.

It uses symbols for assigning permissions to the users. There are following symbols for users of three categories.

| Option            | Symbols                                      | Meaning   |
|-------------------|--|---|
| <b>Category</b>   | <b>u</b><br><b>g</b><br><b>o</b><br><b>a</b> | Owner of the file and directories.<br>Members of the group belonging to the user.<br>All other system users.<br>All the users |
| <b>Permission</b> | <b>r</b><br><b>w</b><br><b>x</b>             | r to assign read permission to a file<br>w to assign write permission to a file<br>x to assign execute permissions to a file  |
| <b>Operations</b> | <b>+</b><br><b>-</b><br><b>=</b>             | Assigns the permission<br>Removes the permission.<br>Assigns absolute permissions.  |

**Syntax:**

```
$chmod <category><operation><permission><filename>
```

**Example:**

- Assigns the execute permission for user/owner, group and others.

```
$chmod ugo+x abc
```

- Assign all permission for group.

```
$chmod g+rwx abc
```

- Assign execute permissions for the owner and remove read permission of group and others.

```
$chmod u+x,go-r abc
```

- Assign read the permission to all three users and removes all remaining permissions.

```
$chmod ugo=r abc
```

**2. Octal or absolute notation**

This method uses a number to specify each set of permissions for the file. It assigns permissions in three digits. First digit assign permission for owner, second digit for group and third for others. Digits range is 0 to 7.

**Syntax:**

```
$chmod <Three Digit Octal Number> <Filename>
```

```
$chmod [ u g o ] filename
```

**List of Octal numbers:**

| Number   | Assignment                         |
|----------|------------------------------------|
| 0        | No permission                      |
| 1        | Execute permission                 |
| 2        | Write permission                   |
| 4        | Read permission                    |
| 5(4+1)   | Read and execute permission        |
| 6(4+2)   | Read and write permission          |
| 7(4+2+1) | Read, write and execute permission |

**Example:**

```
$chmod 750 abc
```

It assigns all permissions for user, read and execute for group and no permission for others for file abc.

**cmp**

This command is used to compare files when used without options. It uses two filenames as an argument and display the differences on the terminals.

```
$cmp filename1 filename2
```

```
$cat student1
```

```
Harsh
```

```
Sujay
```

```
Smith
```

```
$cmp student1 student2
```

```
Student1 student2 differ : char 6 line1
```

```
$cat student2
```

```
Harshu
```

```
Sujay
```

```
Smith
```

**VIII. Resources required (Additional)-Nil**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate program statement to students.**

1. Create the following structure

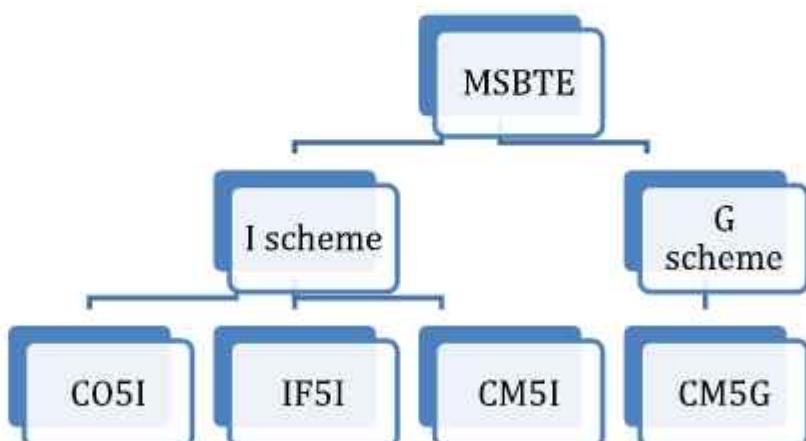


Figure 6

**XI. Result (Output of Code):**

.....

.....

.....

.....

## XII. Practical Related Questions

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. How to shift from Root directory to User (Home) directory?
2. How to see directories?
3. What is default set of permissions given by the system to the directory?
4. Assign all the permissions to your directory for all the users using symbolic and octal method.
5. What is difference between comm and cmp command?

**(Space for answer)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIII. Exercise:**

1. Write the command for performing the following tasks sequentially
  - a. Display your current directory.
  - b. Create a directory 'subject' in the current directory.
  - c. Create a file 'sample' in the directory 'subject'.
  - d. Remove the write permission for the owner for 'sample' using symbolic method.
  - e. Delete the file 'sample'. What is an error message displayed?
2. What are the permissions assigned to the file/files after execution of following commands?
  - a. \$chmod 700 abc
  - .....
  - b. \$chmod u+rwx, go-rwx file1 file2
  - .....
  - c. \$chmod 536 xyz
  - .....
3. Create new files pqr and prq1.  
Perform the commands  
\$chmod ugo=r pqr  
o/p.....  
\$chmod ugo+r prq1  
o/p.....
4. Assign read and write permission for the owner, write permission for the group and execute permission for others using octal method for file mfile.

5. Write commands to assign following permissions to the file OSY using octal method.
- a. \_\_\_\_\_
  - b. \_rw\_r\_xr\_\_
  - c. \_r\_xr\_xr\_x
6. Write commands to assign following permissions to the file OSY using symbolic method.
- a. \_rwxr\_xr\_\_
  - b. \_rwxrwxrwx

(Space for Answer)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| <b>Performance Indicators</b>     |                            | <b>Weightage</b> |
|-----------------------------------|----------------------------|------------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>       |
| 1.                                | Logic formation            | 10%              |
| 2.                                | Debugging ability          | 10%              |
| 3.                                | Follow ethical practices   | 10%              |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>       |
| 4.                                | Expected output            | 30%              |
| 5.                                | Timely Submission          | 30%              |
| 6.                                | Answer to sample questions | 10%              |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b>      |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| <b>Marks Obtained</b>      |                            |                  | <b>Dated signature of Teacher</b> |
|----------------------------|----------------------------|------------------|-----------------------------------|
| <b>Process Related(15)</b> | <b>Product Related(35)</b> | <b>Total(50)</b> |                                   |
|                            |                            |                  |                                   |

## **Practical No. 8: Execute text processing commands.**

### **I. Practical Significance:**

There are many text processing commands. A filter is a program that takes a flow of data from the standard input, processes or filters it and send the result to standard output. cut, wc, tr, sort, grep are all simple filters.

### **II. Relevant Program Outcomes (POs)**

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

Manage operations of Operating System

The practical is expected to develop the following skills:

1. Able to execute commands for text processing.

### **IV. Relevant Course Outcome(s)**

Apply file management techniques.

### **V. Practical Outcome (PrOs)**

Execute text processing tr, wc, cut, paste, spell, sort, grep, more.

### **VI. Relevant Affective domain related Outcome(s)**

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### **VII. Minimum Theoretical Background**

1. **tr (Unix):-** tr is a command in Unix-like operating systems. It is an abbreviation of **translate** or **transliterate**, indicating its operation of replacing or removing specific characters in its input data set.

#### **Syntax**

The syntax of tr command is:

\$ tr [OPTION] SET1 [SET2]

#### **Translation**

If both the SET1 and SET2 are specified and '-d' OPTION is not specified, then tr command will replace each character in SET1 with each character in same position in SET2.

## 1. Convert lower case to upper case

- The following tr command is used to convert the lower case to upper case  

```
$ tr
operating system
OPERATING SYSTEM
fifth semester
FIFTH SEMESTER
```
- The following command will also convert lower case to upper case  

```
$ tr [:lower:] [:upper:]
java
JAVA
```
- You can also use ranges in tr. The following command uses ranges to convert lower to upper case.  

```
$ tr a-z A-Z'
```

## 2. wc

The wc (word count) command in Unix/Linux operating systems is used to find out number of lines, words and characters in a file.

The syntax of wc command as shown below:

```
$wc [-l -w -c] <filename>
-l counts only number of lines
-w counts only number of words
-c counts only number of characters
```

Example:

```
$cat student1
Ajay
Vijay
Sujit
$wc student1
3 3 14
```

Means 3 lines, 3 words and 14 characters.

```
# wc [options] filenames
[root@tecmint ~]# wc tecmint.txt
12 16 112 tecmint.txt
```

The following are the options and usage provided by the command.

| Sr. No. | Command | Description                                 |
|---------|---------|---|
| 1       | wc-l    | Prints the number of lines in a file.       |
| 2       | wc-w    | Prints the number of words in a file.       |
| 3       | wc-c    | Displays the count of bytes in a file.      |
| 4       | wc-m    | Prints the count of characters from a file. |
| 5       | wc-L    | Prints only the length of the longest line  |

### 3. Cut

Cut command:

It is used to copy the specified columns to the standard output file. It is used to cut portion of a file. It takes filenames as command line arguments or input from standard input. It does not delete the selected portion of a file.

Syntax:

`$cut [-f/-c] [n1,n2] filename`

Where n1,n2 are the field/character number.

-f              Displays specified field. Also uses -d for field separator.

-c              Displays specified column character by character.

Example 1:

```
$cat student1
Ajay    1990  50
Vijay   1992  60
Sujit   1991  70
```

```
$cut -f1,2 student1
```

```
Ajay    1990
Vijay   1992
Sujit   1991
```

In this output it will display 1<sup>st</sup> and 2<sup>nd</sup> fields values

Example 2:

```
$cat numbers
```

```
12345
```

```
23123
```

```
45656
```

```
76543
```

```
$cut -c4,5 student2
```

```
45
```

```
23
```

```
56
```

```
43
```

In this output it displays 4<sup>th</sup> and 5<sup>th</sup> character from each line.

### 4. Paste

Paste command is one of the useful commands in Unix or Linux operating system. It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output. When no file is specified, or put dash ("") instead of file name, paste reads from standard input and gives output as it is until a interrupt command [Ctrl-c] is given.

Syntax:

`paste [OPTION]... [FILES]...`

Create files for demonstration:

```
$cat stdname
```

```
Ajay
```

```
Vijay
```

```
Sujit
```

```
$cat birthyear
```

```
1980
```

```
2001
```

```
1996
$paste stdname birthyaer
Ajay 1980
Vijay 2001
Sujit 1996
```

## 5. Spell

Spell is a spell-checking program which scans a text file for misspelled words and prints each misspelled word on its own line.

Syntax:

`spell [OPTIONS] FILE ...`

`-n` Print line numbers.

(For each misspelled word, also print the number of the line it was in.)

`-o` Print file names.

(For each misspelled word, also print the name of the file it was in.)

`spell -on words.txt words2.txt`

Checks the spelling of files `words.txt` and `words2.txt`, printing the file name (`-o`) and the line number (`-n`) of each misspelled word.

```
Jaishree@DESKTOP-OK05FND ~
$ cat student1
Sujay          2001
Bhavesh        2000
Pinank         2000

Jaishree@DESKTOP-OK05FND ~
$ sort student1
Bhavesh        2000
Pinank         2000
Sujay          2001

Jaishree@DESKTOP-OK05FND ~
$ cat numbers
12
45
2
34
8

Jaishree@DESKTOP-OK05FND ~
$ sort -n numbers
2
8
12
34
45
```

Figure 7

## 6. Sort

SORT command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ASCII. Using options in sort command, it can also be used to sort numerically.

When we have a mix file with both uppercase and lowercase letters then first the lower case letters would be sorted following with the upper case letters .

| Option | Meaning                               |
|--------|---------------------------------------|
| -o     | write the <b>output to a new file</b> |
| -n     | sort a file numerically               |
| -r     | Sort in Reverse Order                 |

## 7. grep (Global Regular expression printer)

It is used to search a pattern/word inside files.

Syntax:

\$grep <option> <pattern> <filename>

| Options | Meaning  |
|---------|--|
| -c      | Displays the count of the number of occurrences of the pattern   |
| -l      | Displays the list of the filenames which contains a pattern.     |
| -n      | Displays line numbers along with the lines containing a pattern. |

\$cat OS

Unix

Linux

Ubuntu

\$grep -n Unix OS

It searches Unix pattern and displays line number where that pattern matched.

## 8. more

If the file is too large for its contents to fit in one screen, it will scroll off your screen when the file is displayed with cat command then use more command.

Syntax:

\$more [“Space bar” /f/b/j/k/200G/G] filename

| Options | Meaning  |
|---------|--|
| [Space] | scrolls the display, one screenful of data at a time |
| j       | scrolls forward one line                             |
| f       | Scrolls forward one screen                           |
| b       | scrolls backwards one screen                         |
| k       | scrolls backward one line                            |
| 200G    | Goes to line number 200                              |
| G       | Goes to last line of the file                        |

**VIII. Resources required (Additional)**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM- 2GB and above (As per need of OS) | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate program statement to students.**

Write the commands for:

Counting number of words in the 'data.txt'

Counting number of lines in 'data.txt'

Counting all characters in the 'data.txt'

**XI. Result (Output of Command):**

.....

.....

.....

.....

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. Give Applications of Paste Command.
2. How to move cursor to end of a line?
3. What are the options of wc command?
4. What are different types of filters used in Linux?
5. What is difference between \$cat abc and \$cat abc|more (File abc shall consists of more than 25 lines in it)

**(Space for answer)**

**XIII. Exercise:**

1. Write the significance of following.
  - i. **Only one character is specified**  
\$ paste -d "|" number state capital  
1|Arunachal Pradesh|Itanagar  
2|Assam|Dispur  
3|Andhra Pradesh|Hyderabad  
4|Bihar|Patna  
5|Chhattisgrah|Raipur
  - ii. **More than one character is specified**  
\$ paste -d "|," number state capital

- 1|Arunachal Pradesh,Itanagar
- 2|Assam,Dispur
- 3|Andhra Pradesh,Hyderabad
- 4|Bihar,Patna
- 5|Chhattisgrah,Raipur

**iii. -s (serial), Combination of -d and -s, –version (write its syntax and example)**

1. Create a new file and practice executing shell commands from within the editor.  
Capture the results of some shell commands into the file.
  2. How to get help?
2. Try the commands and write output with its meaning

i. tr “[a-f]” “[0-5]” < employee (employee is name of file)

.....  
.....  
.....  
.....

ii. tr -s “ “ < employee

.....  
.....  
.....  
.....

iii. tr -d “f” < employee

.....  
.....  
.....  
.....

**(Space for Answer)**

**XIV. References/ Suggestions for Further Reading**

1. <https://www.thegeekstuff.com/2012/12/linux-tr-command>
2. <https://www.tecmint.com/wc-command-examples/>
3. <http://web.stanford.edu/class/cs273a/presentations.aut11/UnixTextProcessingPrimer.pdf>
4. <https://www.ibm.com/developerworks/aix/library/au-unixtext/index.html>

**XV. Assessment Scheme**

| <b>Performance indicators</b>     |                            | <b>Weightage</b> |
|-----------------------------------|----------------------------|------------------|
| <b>Process related (35 Marks)</b> |                            | <b>70%</b>       |
| 1.                                | Logic formation            | 30%              |
| 2.                                | Debugging ability          | 30%              |
| 3.                                | Follow ethical practices   | 10%              |
| <b>Product related (15 Marks)</b> |                            | <b>30%</b>       |
| 4.                                | Expected output            | 10%              |
| 5.                                | Timely Submission          | 10%              |
| 6.                                | Answer to sample questions | 10%              |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b>      |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| <b>Marks Obtained</b>      |                            |                  | <b>Dated signature of Teacher</b> |
|----------------------------|----------------------------|------------------|-----------------------------------|
| <b>Process Related(35)</b> | <b>Product Related(15)</b> | <b>Total(50)</b> |                                   |
|                            |                            |                  |                                   |

## Practical No. 9: Use vi editor and perform all editor commands.

### I. Practical Significance:

There are many ways to edit files in Unix. Editing files using the screen-oriented text editor **vi** is one of the best ways. This editor enables you to edit lines in context with other lines in the file. Vi is intended as a plain text editor (similar to Notepad on Windows, or Textedit on Mac) as opposed to a word processing suite such as Word or Pages. It does, however have a lot more power compared to Notepad or Textedit.

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System

The practical is expected to develop the following skills:

1. Able to execute commands in the given modes.
2. Able to identify vi editor commands.

### IV. Relevant Course Outcome(s)

Apply file management techniques.

### V. Practical Outcome (PrOs)

Use vi editor and perform all editor commands.

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

#### Starting the vi Editor

The following table lists out the basic commands to use the vi editor –

| Sr. No. | Command            | Description  |
|---------|--------------------|--|
| 1       | <b>vi filename</b> | Creates a new file if it already does not exist, otherwise opens an existing file. |

|   |                       |   |
|---|-----------------------|---|
| 2 | <b>vi -R filename</b> | Opens an existing file in the read-only mode. |
| 3 | <b>view filename</b>  | Opens an existing file in the read-only mode. |

Following is an example to create a new file test1 if it already does not exist in the current working directory –

The above command will generate the following output

```
$vi test1
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

"test1" [New File]

You will notice a tilde (~) on each line following the cursor. A tilde represents an unused line. If a line does not begin with a tilde and appears to be blank, there is a space, tab, newline, or some other non-viewable character present.

#### **Operation Modes**

While working with the vi editor, we usually come across the following two modes –

**Command mode** – This mode enables you to perform administrative tasks such as saving the files, executing the commands, moving the cursor, cutting (yanking) and pasting the lines or words, as well as finding and replacing. In this mode, whatever you type is interpreted as a command.

**Insert mode** – This mode enables you to insert text into the file. Everything that's typed in

this mode is interpreted as input and placed in the file.

vi always starts in the command mode. To enter text, you must be in the insert mode for which simply type i. To come out of the insert mode, press the Esc key, which will take you back to the command mode.

#### **Moving within a File**

To move around within a file without affecting your text, you must be in the command mode (press Esc twice).

| Sr. No. | Command | Description  |
|---------|---------|--|
| 1       | k       | Moves the cursor up one line                         |
| 2       | j       | Moves the cursor down one line                       |
| 3       | h       | Moves the cursor to the left one character position  |
| 4       | l       | Moves the cursor to the right one character position |

**Control Commands:-****Editing Files**

To edit the file, you need to be in the insert mode. There are many ways to enter the insert mode from the command mode –

| Sr. No. | Command | Description   |
|---------|---------|---|
| 1       | i       | Inserts text before the current cursor location             |
| 2       | I       | Inserts text at the beginning of the current line           |
| 3       | a       | Inserts text after the current cursor location              |
| 4       | A       | Inserts text at the end of the current line                 |
| 5       | o       | Creates a new line for text entry below the cursor location |
| 6       | O       | Creates a new line for text entry above the cursor location |

**Page movement Description:-**

| Sr. No. | Command  | Description                   |
|---------|----------|-------------------------------|
| 1       | CTRL+d - | Move forward 1/2 screen       |
| 2       | CTRL+f - | Move forward one full screen  |
| 3       | CTRL+u - | Move backward 1/2 screen      |
| 4       | CTRL+b   | Move backward one full screen |
| 5       | CTRL+e   | Moves screen up one line      |
| 6       | CTRL+y   | Moves screen down one line    |
| 7       | CTRL+i   | Redraws screen                |

### **Deleting Characters**

Here is a list of important commands, which can be used to delete characters and lines in an open file –

| Sr. No. | Command | Description   |
|---------|---------|---|
| 1       | x       | Deletes the character under the cursor location                       |
| 2       | X       | Deletes the character before the cursor location                      |
| 3       | Dw      | Deletes from the current cursor location to the next word             |
| 4       | d^      | Deletes from the current cursor position to the beginning of the line |
| 5       | d\$     | Deletes from the current cursor position to the end of the line       |
| 6       | D       | Deletes from the cursor position to the end of the current line       |
| 7       | dd      | Deletes the line the cursor is on                                     |

### **Change Commands**

You also have the capability to change characters, words, or lines in vi without deleting them.

| Sr. No. | Command | Description  |
|---------|---------|--|
| 1       | cc      | Removes the contents of the line, leaving you in insert mode.  |
| 2       | cw      | Changes the word the cursor is on from the cursor to the lowercase w end of the word.  |
| 3       | r       | Replaces the character under the cursor. vi returns to the command mode after the replacement is entered.                          |
| 4       | R       | Overwrites multiple characters beginning with the character currently under the cursor. You must use Esc to stop the overwriting.  |
| 5       | s       | Replaces the current character with the character you type. Afterward, you are left in the insert mode.                            |
| 6       | S       | Deletes the line the cursor is on and replaces it with the new text. After the new text is entered, vi remains in the insert mode. |

### **Copy and Paste Commands**

You can copy lines or words from one place and then you can paste them at another place using the following commands –

| Sr. No. | Command | Description   |
|---------|---------|---|
| 1       | yy      | Copies the current line.  |
| 2       | yw-     | Copies the current word from the character the lowercase w cursor is on, until the end of the word. |
| 3       | p       | Puts the copied text after the cursor   |
| 4       | P       | Puts the yanked text before the cursor.   |

**Ex mode command**

Press Esc key and then (:) colon to enter ex-mode commands. A colon is displayed at the left hand corner of the last line on your screen.

| Command | Action                                 |
|---------|--|
| :w      | Saves file and remains in editing mode |
| :x      | Saves file and quits editing mode      |
| :wq     | Save and quit                          |
| :q      | Quits vi when no changes are made      |
| :q!     | Quits vi cancelling the changes        |
| :sh     | Escape to Unix shell                   |

**Advanced Commands**

There are some advanced commands that simplify day-to-day editing and allow for more efficient use of vi –

**Word and Character Searching**

The vi editor has two kinds of searches: **string** and **character**. For a string search, the / and ? commands are used. When you start these commands, the command just typed will be shown on the last line of the screen, where you type the particular string to look for.

These two commands differ only in the direction where the search takes place –

- The / command searches forwards (downwards) in the file.
- The ? command searches backwards (upwards) in the file.

The n and N commands repeat the previous search command in the same or the opposite direction, respectively. Some characters have special meanings. These characters must be preceded by a backslash (\) to be included as part of the search expression.

| Sr. No. | Character | Description  |
|---------|-----------|--|
| 1       | ^         | Searches at the beginning of the line (Use at the beginning of a search expression). |
| 2       | .         | Matches a single character.  |

|   |    |  |
|---|----|--|
| 3 | *  | Matches zero or more of the previous character.  |
| 4 | \$ | End of the line (Use at the end of the search expression).   |
| 5 | [  | Starts a set of matching or non-matching expressions.  |
| 6 | <  | This is put in an expression escaped with the backslash to find the ending or the beginning of a word. |
| 7 | >  | This helps see the '<' character description above.  |

### Running Commands

The vi has the capability to run commands from within the editor. To run a command, you only need to go to the command mode and type :! command. For example, if you want to check whether a file exists before you try to save your file with that filename, you can type :! ls and you will see the output of ls on the screen. You can press any key (or the command's escape sequence) to return to your vi session.

### Replacing Text

The substitution command (:s/) enables you to quickly replace words or groups of words within your files. Following is the syntax to replace text –

:s/search/replace/g

The g stands for globally. The result of this command is that all occurrences on the cursor's line are changed.

### VIII. Resources required (Additional)

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

### IX. Resources used (Additional)

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate program statement to students.**

Create a file OS and type at least 30 lines in it.

- i) Write command to save the file
- ii) Write the command to insert text at the end of line.
- iii) Replace some character into beginning of second line.
- iv) Write a command to delete 10 lines at a time.
- v) Write a command to move cursor 3 word to the right

**XI. Result (Output of Code):**

.....  
.....  
.....  
.....

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. What is the difference between Type o and Type O?
2. State the three ways quit the vi-editor.
3. What is effect of (Type this at colon mode and observe the changes in your file)
  - a. set nu
  - b. set ic
  - c. set nonu
  - d. set noai
4. Give vi command to delete line to the left of cursor.
5. What is difference between yank and delete?
6. What are different modes in Vi editor?
7. Observe the output of following commands:

- i) at colon mode

:abbr MSBTE Maharashtra State Board Of Technical Education

Now in the beginning of your file type MSBTE and press space bar

Write down the output

.....

- ii) Go to beginning of your file press 4YY

Then move to end of your file and press 'P'

Write the purpose of the command

**(Space for answer)**

.....  
.....  
.....  
.....



**XIII. Exercise:**

1. Write the significance of following.
  - i. \$ vi temp.txt
  - ii. Insert multiple lines.
  - iii. Delete contents using commands.
2. Create a new file and practice executing shell commands from within the editor. Capture the results of some shell commands into the file.
3. How to get help?

**(Space for Answer)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>
4. <https://www.openvim.com/tutorial.html>

**XV. Assessment Scheme**

| Performance indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (35 Marks)</b> |                            | <b>70%</b>  |
| 1.                                | Logic formation            | 30%         |
| 2.                                | Debugging ability          | 30%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (15 Marks)</b> |                            | <b>30%</b>  |
| 4.                                | Expected output            | 10%         |
| 5.                                | Timely Submission          | 10%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. .....
2. .....
3. .....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(35) | Product Related(15) | Total(50) |                            |
|                     |                     |           |                            |

## Practical No. 10: Execute Shell Script by using if statements.

### I. Practical Significance:

If is a statement that allows the programmer to make a decision in the program based on conditions he specified. If the condition is met, the program will execute certain lines of code otherwise, the program will execute other tasks the programmer specified.

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System

The practical is expected to develop the following skills:

1. Able to execute script for single decision if statement.
2. Able to execute script for double decision if statement.

### IV. Relevant Course Outcome(s)

Apply file management techniques.

### V. Practical Outcome (PrOs)

Write and execute Shell Script by using following Control statements features- "if" statement.

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

The if...else...fi statement is the control statement that allows Shell to execute statements in a controlled way and make the right choice.

Syntax:

if [ expression ]

then

Statement(s) to be executed if expression is true

else

Statement(s) to be executed if expression is not true

fi

The Shell *expression* is evaluated in the above syntax. If the resulting value is *true*, given *statement(s)* are executed. If the *expression* is *false*, then no statement will be executed.

Relational operators are used along with if statement.

### Relational operators used by *if*

| Operator | Meaning                  |
|----------|--------------------------|
| -eq      | Equal to                 |
| -ne      | Not equal                |
| -gt      | Greater than             |
| -ge      | Greater than or equal to |
| -lt      | Less than                |
| -le      | Less than or equal to    |

### General Syntax

#### 1. Single Decision: -

Syntax:-

```
if [ condition ]
then
### series of code goes here
fi
```

e.g.

```
if [ "$X" -lt "0" ]
then
echo "X is less than zero"
fi
```

#### 2. Double Decision: -

Syntax:-

```
if [ condition ]
then
    Statements if the condition is satisfied
else
    Statements of code if the condition is not satisfied
fi
```

#### 3. Multiple if condition: -

Syntax:-

```
if [ condition1 ]
then
    Statements for condition1
elif [ condition2 ]
then
    Statements for condition2
else
    Statements if the condition is not satisfied
fi
```

**4. Double -bracket:-**

Syntax:-  
 if ((condition))  
 then  
     Statements goes here  
 fi

To type any program in Linux you need the compiler. vi editor is used for creating file in Linux.

Write a shell script to display welcome message with today's date in it.

In vi editor write a program code and save and quit with filename 'sample.sh'.

\$vi sample.sh  
 (Now press escape and i to change the input mode and type following program)  
 #sample.sh

```
#use of if-then statement
if whoami; then
var1='whoami'
echo "Welcome $var1"
fi
if date; then
echo "It displays todays date"
fi
echo "This is end of script"
```

To run the script

./sample.sh

Save this program press escape+shift+:(colon) and type wq (:wq)  
 "Welcome User1"

Mon Feb 25 22:29:29 PST 2019

"It displays todays date"

"This is end of script"

**VIII. Resources required (Additional)**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate program statement to students.**

Execute shell script by considering example to find passing grades of students using if statement:

1. Single Decision.
2. Double Decision.
3. Multiple if statements.

**Note:** Attach the code at the end.

**XI. Result (Output of Code):**

---

---

---

---

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

- A. Write and execute script for nested if statements.
- B. Write difference between
  1. if [ condition]
  2. if((condition))
- C. Write script for finding greatest number among given three number.

(Space for answer)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIII. Exercise:**

1. Correct the following script and write its output.
  1. if [ ! -r "\$1" ] then echo "File \$1 is not readable – skipping."; fi
  2. if [ "\$X" -nt "/etc/passwd" ]; Then  
          echo "X is a file which is newer than /etc/passwd"  
      if

(Space for Answer)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>
4. <https://www.shellscript.sh/test.html>

**XV. Assessment Scheme**

| Performance Indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (35 Marks)</b> |                            | <b>70%</b>  |
| 1.                                | Logic formation            | 30%         |
| 2.                                | Debugging ability          | 30%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (15 Marks)</b> |                            | <b>30%</b>  |
| 4.                                | Expected output            | 10%         |
| 5.                                | Timely Submission          | 10%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(35) | Product Related(15) | Total(50) |                            |
|                     |                     |           |                            |

## Practical No. 11: Execute Shell Script by using for statements.

### I. Practical Significance:

The **for** loops iterate through a set of values until the list is exhausted. In simple terms the for loop operates on lists of items. It repeats a set of commands for every item in a list.

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System

The practical is expected to develop the following skills:

1. Able to execute script for 'FOR' loop.

### IV. Relevant Course Outcome(s)

Apply file management techniques.

### V. Practical Outcome (PrOs)

Write and execute Shell Script by using following Control statements feature- "for" statement, exit, break, continue.

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

**Syntax of FOR loop:** -

```
for var in item1 item2 ... itemN
```

```
    do
```

```
        command1
```

```
        command2
```

```
        ....
```

```
        ....
```

```
        commandN
```

```
    done
```

**Example:**

```
for var in 0 1 2 3 4 5 6 7 8 9
do
    echo $var
done
```

So far you have looked at creating loops and working with loops to accomplish different tasks. Sometimes you need to stop a loop or skip iterations of the loop. Following two statements that are used to control shell loops—

**Three-expression bash for loops syntax**

This type of for loop can also be used similar to the C programming language. It is having by a three-parameter loop control expression; consisting of an initializer (EXP1), a loop-test or condition (EXP2), and a counting expression (EXP3).

**Three-expression bash for loops syntax**

A representative three-expression example in bash as follows:

```
#!/bin/bash
for (( c=1; c<=5; c++ ))
do
    echo "Welcome \$c times"
done
```

- The break statements:** The break statement is used to terminate the execution of the entire loop, after completing the execution of all of the lines of code up to the break statement using **break** keyword.
- The continue statement:** The continue statement is similar to the break command, except that it causes the current iteration of the loop to exit, rather than the entire loop.
- Exit:**

The syntax is as follows:

**exit N**

The exit statement is used to exit from the shell script with a status of N.

Write a shell script to calculate the factorial of a given number.

In vi editor write a program code and save and quit with filename ‘fact.sh’.

\$vi fact.sh

(Now press escape and i to change the input mode and type following program)

Similar other loops are also used in Linux such as while, case..esac  
while statement

**Syntax:**

```
while [condition]
do
```

“

“

“

done

**Example:**

```
a=1
while [ $a -le 5 ]
do
```

```
echo "value of a=" $a
a=`expr $a + 1`
done

case .. esac statement;
Syntax:
case value in
pattern 1) commands;;
pattern 2) commands;;
pattern 3) commands;;
..
*) Default
esac // indication of end of case statement
```

**Example:**

```
#!/bin/bash
NOW=$(date +"%a")
case $NOW in
    Mon)
        echo "Full backup";;
    Tue|Wed|Thu|Fri)
        echo "Partial backup";;
    Sat|Sun)
        echo "No backup";;
*)
esac
```

```
$cat fact.sh
echo "Enter a number"
read num
fact=1
while [ $num -gt 1 ]
do
    fact=$((fact * num))
    num=$((num - 1))
done
echo "factorial of a number is $fact"
Save this program press escape+shift+:(colon) and type wq (:wq)
To compile the program
$./fact.sh or $sh fact.sh
Enter the number : 5
Factorial of a given number is 120
```

**VIII. Resources required (Additional)**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM-2GB and above (As per need of OS)  | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate program statement to students.**

1. Execute shell script by considering example like printing table of given number by FOR loop
2. Execute shell script by considering example like printing following output by FOR loop

\*\*\*\*\*  
\*\*\*\*  
\*\*  
\*

**Note:** Attach the code at the end.

**XI. Result (Output of Code):****XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. Give output of the following

```
a.
#!/bin/sh
NUMBERS="1 2 3 4 5 6 7"
for NUM in $NUMS
do
    Q=`expr $NUM % 2`
    if [ $Q -eq 0 ]
    then
        echo "Number is an even number!!"
    continue
    fi
    echo "Found odd number"
done
```

```
b.
#!/bin/sh
a=0
while [ $a -lt 10 ]
do
    echo $a
    if [ $a -eq 5 ]
    then
        break
    fi
    a=`expr $a + 1`
```

2. State the difference between Iteration and Recursion.
3. Write a shell script to display Fibonacci series for n numbers.
4. Write a shell script to display tables of 2 to 10 numbers. (Like  $2 * 1 = 2$ )
5. Write a shell script to accept five digit number and perform addition of all digits.

**(Space for answer)**

**XIII. Exercise:**

Execute the script for the following.

1. The for loop using day of week list.
2. The while loop to print different \* patterns.
3. The case statement for performing various mathematical operations.

**(Space for Answer)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>
4. <https://www.cyberciti.biz/faq/bash-for-loop/>
5. [https://bash.cyberciti.biz/guide/For\\_loop](https://bash.cyberciti.biz/guide/For_loop)

**XV. Assessment Scheme**

| Performance indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (35 Marks)</b> |                            | <b>70%</b>  |
| 1.                                | Logic formation            | 30%         |
| 2.                                | Debugging ability          | 30%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (15 Marks)</b> |                            | <b>30%</b>  |
| 4.                                | Expected output            | 10%         |
| 5.                                | Timely Submission          | 10%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| Marks Obtained          |                         |           | Dated signature of Teacher |
|-------------------------|-------------------------|-----------|----------------------------|
| Process Related<br>(35) | Product Related<br>(15) | Total(50) |                            |
|                         |                         |           |                            |

## Practical No. 12: Write Shell script to find out whether - Given file exists?

### I. Practical Significance:

In UNIX data and programs are stored in files. These are organized in directories. In a simple way, a directory is just a **file** that contains other files (or directories). It can be done to find out if a file exists or not on the file system with the help of BASH shell and IF command,

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

The practical is expected to develop the following skills:

1. Able to execute shell script to find out whether - Given file exists?

### IV. Relevant Course Outcome(s)

Apply file management techniques.

### V. Practical Outcome (PrOs)

Write Shell script to find out whether - Given file exists?

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

This can be done by the shell script by using test conditions. Also if..else..fi statement is used which allows us to make decisions based on the success or failure of a command.

Following is the list of some other flags which we can use in File test operators.

| Shell   | Returns true if                            |
|---------|--|
| -a file | True if file exists.                       |
| -b FILE | True if it is block special                |
| -c FILE | True if it is character special            |
| -d file | True if file exists and is a directory.    |
| -e file | True if file exists.                       |
| -f file | True if file exists and is a regular file. |

-s file      True if file exists and has a size greater than zero.

**Example:**

```
#!/bin/bash
echo -e "Enter the name of the file :"
read file_name
if [ -f $file_name ]
then
else
echo "$file_name not exists"
fi
```

**Ordinary Files** – An ordinary file is a file on the system that contains data, text, or program instructions. In this tutorial, you look at working with ordinary files.

**Directories** – Directories store both special and ordinary files. For users familiar with Windows or Mac OS, Unix directories are equivalent to folders.

**Special Files** – Some special files provide access to hardware such as hard drives, CD-ROM drives, modems, and Ethernet adapters. Other special files are similar to aliases or shortcuts and enable you to access a single file using different names.

### VIII. Resources required (Additional)-Nil

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM- 2GB and above (As per need of OS) | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

### IX. Resources used (Additional)

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

### X. Program Code: Teacher must assign a separate program statement to students.

Write a shell script which accepts input as a file or directory and displays message whether it is file or directory.

**XI. Result (Output of Code):**

---

---

---

---

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. What is the command to run the script?
2. What are file test options with its meaning?
3. What will be the output of these commands (file=abc.txt)
  - a) if [ ! -f "\$file" ]; then  
    echo "\$file"  
    fi
  - b) test -f "\$file" || echo "\$file"
  - c) [ -f "\$file" ] || echo "\$file"

(Space for answer)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---



**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| <b>Performance Indicators</b>     |                            | <b>Weightage</b> |
|-----------------------------------|----------------------------|------------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>       |
| 1.                                | Logic formation            | 10%              |
| 2.                                | Debugging ability          | 10%              |
| 3.                                | Follow ethical practices   | 10%              |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>       |
| 4.                                | Expected output            | 30%              |
| 5.                                | Timely Submission          | 30%              |
| 6.                                | Answer to sample questions | 10%              |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b>      |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| <b>Marks Obtained</b>       |                             |                  | <b>Dated signature of Teacher</b> |
|-----------------------------|-----------------------------|------------------|-----------------------------------|
| <b>Process Related (15)</b> | <b>Product Related (35)</b> | <b>Total(50)</b> |                                   |
|                             |                             |                  |                                   |

## Practical No. 13: Write Shell script to check and grant File permissions.

### I. Practical Significance:

In Unix files come with permissions, a way to decide who can read, write or execute a file. These permissions are divided into three parts: those for the owner (user) of the file, those for the group to which the owner belongs and then permissions for all the other users from the point of view of what is allowed to do in a file or directory, permissions are for reading, writing and executing.

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

The practical is expected to develop the following skills:

1. Able to execute shell script to check and grant file permissions.

### IV. Relevant Course Outcome(s)

Apply file management techniques.

### V. Practical Outcome (PrOs)

Write Shell script to find out whether - File has read, write, and execute permissions

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

#### Test Commands

| Test    | Returns true if                                       |
|---------|---|
| -r file | True if file exists and is readable.                  |
| -s file | True if file exists and has a size greater than zero. |
| -w file | True if file exists and is writable.                  |
| -x file | True if file exists and is executable.                |

**VIII. Resources required (Additional)-**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM- 2GB and above (As per need of OS) | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional)**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate program statement to students.**

Write Shell script to find out whether - File has read, write, and execute permissions

*(This practical can be performed in any of the compiler like C, vi editor etc)***Note: Attach the code at the end.**

**XI. Result (Output of Code):**

---

---

---

---

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. What are permissions of a file?
2. How to assign permission to a file?
3. What happens when exception is thrown by main method?
4. How to check permissions of all files and directories?
5. What are the test commands to check the permission of a file?

**(Space for answer)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIII. Exercise:**

1. Write a shell script which displays the list of all executable files in the current working directory.
2. Write a shell script which displays a list of all the files in the current directory to which user has read, write and execute permission.
3. Write a shell script which accepts a filename and assigns it all the permissions.

**(Space for Answer)**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix/>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| <b>Performance Indicators</b>     |                            | <b>Weightage</b> |
|-----------------------------------|----------------------------|------------------|
| <b>Process related (15 Marks)</b> |                            | <b>30%</b>       |
| 1.                                | Logic formation            | 10%              |
| 2.                                | Debugging ability          | 10%              |
| 3.                                | Follow ethical practices   | 10%              |
| <b>Product related (35 Marks)</b> |                            | <b>70%</b>       |
| 4.                                | Expected output            | 30%              |
| 5.                                | Timely Submission          | 30%              |
| 6.                                | Answer to sample questions | 10%              |
| <b>Total (25 Marks)</b>           |                            | <b>100%</b>      |

*List of Students /Team Members*

1. .....
2. .....
3. .....

| <b>Marks Obtained</b>          |                                 |                  | <b>Dated signature<br/>of Teacher</b> |
|--------------------------------|---------------------------------|------------------|---------------------------------------|
| <b>Process<br/>Related(15)</b> | <b>Product Related<br/>(35)</b> | <b>Total(50)</b> |                                       |
|                                |                                 |                  |                                       |

## Practical No. 14: Implement scheduling algorithms.

### I. Practical Significance:

In Multiprogramming systems, the Operating system schedules the processes on the CPU to have the maximum utilization of it and this procedure is called CPU scheduling. The Operating System uses various scheduling algorithm to schedule the processes. First come first serve (FCFS) scheduling algorithm simply schedules the jobs according to their arrival time.

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

Manage operations of Operating System.

The practical is expected to develop the following skills:

1. Able to find out turnaround time and average waiting time and performance of the algorithm.

### IV. Relevant Course Outcome(s)

Apply scheduling algorithms to calculate turnaround time and average waiting time.

### V. Practical Outcome (PrOs)

Write a program to calculate total waiting and turnaround time of n processes with First Come First Serve CPU scheduling algorithm.

### VI. Relevant Affective domain related Outcome(s)

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### VII. Minimum Theoretical Background

In multiprogramming systems, when there is more than one runnable process (i.e., ready),

the operating system must decide which one to activate. The decision is made by the part of the operating system called the scheduler, using a scheduling algorithm.

CPU scheduling selects the processes from memory that are ready to execute, and allocates the CPU to one of them.

For example, in FCFS the job which comes first in the ready queue will get the CPU first. The lesser the arrival time of the job, the sooner will the job get the CPU. FCFS

scheduling may cause the problem of starvation if the burst time of the first process is the longest among all the jobs.

### VIII. Resources required (Additional):

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM- 2GB and above (As per need of OS) | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

### IX. Resources used (Additional):

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

X. **Program Code:** Consider the processes P1, P2, P3, P4 given in the below table, arrives for execution in the same order, with Arrival Time 0, and given Burst Time, Find using the FCFS scheduling algorithm.

- 1.Turn Around Time for each process
- 2.Waiting Time for each process
- 3.Average Turn Around time
- 4.average waiting time

| Process | Burst Time |
|---------|------------|
| P1      | 21         |
| P2      | 6          |
| P3      | 3          |
| P4      | 2          |

**Note: Attach the code at the end.**

*(This practical can be performed in any of the compiler like Turbo C, vi editor etc)*

**XI. Result (Output of Code):**

**XII. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. Compare SJF, Priority and RR with respect to turnaround time and average waiting time.
2. State the conditions for preemptive and non-preemptive scheduling algorithm.
3. Give the reason of problems arises in FCFS.
4. Write a formula for turnaround Time.
5. Write a formula for Average waiting Time.

**(Space for answer)**

**XIII. Exercise:**

- 1 Draw the Gantt chart for above mention example.
- 2 The jobs are scheduled for execution as follows:  
Solve the problem by using FCFS and preemptive SJF  
Find Average waiting time using Gantt chart

| Process | Arrival Time | Burst Time |
|---------|--------------|------------|
| P1      | 0            | 10         |
| P2      | 1            | 4          |
| P3      | 2            | 14         |
| P4      | 3            | 8          |

- 3 Calculate average waiting time using RR algorithm for the following set of processes with the length of the CPU burst time given in milliseconds. (Time quantum 20 ms)

| Process | Burst Time |
|---------|------------|
| P1      | 12         |
| P2      | 45         |
| P3      | 78         |
| P4      | 90         |

(Space for Answer)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**XIV References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV Assessment Scheme**

| Performance Indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (35 Marks)</b> |                            | <b>70%</b>  |
| 1.                                | Logic formation            | 30%         |
| 2.                                | Debugging ability          | 30%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (15 Marks)</b> |                            | <b>30%</b>  |
| 4.                                | Expected output            | 10%         |
| 5.                                | Timely Submission          | 10%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. ....
2. ....
3. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(35) | Product Related(15) | Total(50) |                            |
|                     |                     |           |                            |

## **Practical No. 15: Implement page replacement algorithm.**

### **I. Practical Significance:**

When there is a page fault, the referenced page must be loaded. If there is no available frame in memory, then one page is selected for replacement. If the selected page has been modified, it must be copied back to disk (swapped out). A page replacement algorithm is needed to decide which page needs to be replaced when new page comes in.

### **II. Relevant Program Outcomes (POs)**

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming / technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

Manage operations of Operating System.

The Practical is expected to develop the following skills:

1. Able to write a program to solve page fault problem and page hit.
2. Able to develop a program to implement page replacement algorithm.

### **IV. Relevant Course Outcome(s)**

Calculate efficiency of different memory management techniques.

### **V. Practical Outcome (PrOs)**

Write a 'C' program to implement FIFO page replacement algorithm.

### **VI. Relevant Affective domain related Outcome(s)**

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices.

### **VII. Minimum Theoretical Background**

#### **In First Out (FIFO) –**

This is the simplest page replacement algorithm. In this algorithm, operating system keeps track of all pages in the memory in a queue, oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal.

**VIII. Resources required (Additional):**

| Sr. No. | Name of Resource | Broad Specification                                       | Quantity          | Remarks (If any)    |
|---------|------------------|---|-------------------|---------------------|
| 1       | Computer System  | Processor (i3-i5), RAM- 2GB and above (As per need of OS) | As per batch Size | For all Experiments |
| 2.      | Operating System | Unix/Linux/Ubuntu/any other open sources operating system |                   |                     |

**IX. Resources used (Additional):**

| Sr. No. | Name of Resource | Broad Specification | Quantity | Remarks (If any) |
|---------|------------------|---------------------|----------|------------------|
| 1       | Computer System  |                     |          |                  |
| 2.      | Operating System |                     |          |                  |

**X. Program Code: Teacher must assign a separate program statement to students.**

Page Reference Stream:

1 2 3 2 1 5 2 1 6 2 5 6 3 1 3 6 1 2 4 3

**Note: Attach the code at the end.****XI. Result (Output of Code):**

.....

.....

.....

.....

**XII. Practical Related Questions***Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.*

1. State the advantages and disadvantages of FIFO.
2. What is page hit?
3. What is page fault?
4. Count number of page hits in above mention example given in section X

**(Space for answer)**



**XIII. Exercise:**

- 1 Consider the reference string 9,7,6,1,7,6,1,2,7,2 the number of frames in the memory is 3. Find out the number of page faults and page hit respective to
    1. FIFO Page replacement algorithm
    2. Optimal Page replacement algorithm
    3. LRU Page replacement algorithm
  - 2 Compare FIFO and LRU.
- .....
- .....
- .....
- .....

(Space for Answer)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**XIV. References/ Suggestions for Further Reading**

1. <https://www.tutorialspoint.com/unix>
2. <http://www.ee.surrey.ac.uk/Teaching/Unix/>
3. <https://www.cs.sfu.ca/~ggbaker/reference/unix/>

**XV. Assessment Scheme**

| Performance indicators            |                            | Weightage   |
|-----------------------------------|----------------------------|-------------|
| <b>Process related (35 Marks)</b> |                            | <b>70%</b>  |
| 1.                                | Logic formation            | 30%         |
| 2.                                | Debugging ability          | 30%         |
| 3.                                | Follow ethical practices   | 10%         |
| <b>Product related (15 Marks)</b> |                            | <b>30%</b>  |
| 4.                                | Expected output            | 10%         |
| 5.                                | Timely Submission          | 10%         |
| 6.                                | Answer to sample questions | 10%         |
| <b>Total (50 Marks)</b>           |                            | <b>100%</b> |

*List of Students /Team Members*

1. .....
2. .....
3. .....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(35) | Product Related(15) | Total(50) |                            |
|                     |                     |           |                            |





## List Of Laboratory Manuals Developed by MSBTE

### **First Semester:**

|   |                           |       |
|---|---------------------------|-------|
| 1 | Fundamentals of ICT       | 22001 |
| 2 | English                   | 22101 |
| 3 | English Work Book         | 22101 |
| 4 | Basic Science (Chemistry) | 22102 |
| 5 | Basic Science (Physics)   | 22102 |

### **Second Semester:**

|    |   |       |
|----|---|-------|
| 1  | Bussiness Communication Using Computers     | 22009 |
| 2  | Computer Peripherals & Hardware Maintenance | 22013 |
| 3  | Web Page Design with HTML                   | 22014 |
| 4  | Applied Science (Chemistry)                 | 22202 |
| 5  | Applied Science (Physics)                   | 22202 |
| 6  | Applied Machines                            | 22203 |
| 7  | Basic Surveying                             | 22205 |
| 8  | Applied Science (Chemistry)                 | 22211 |
| 9  | Applied Science (Physics)                   | 22211 |
| 10 | Fundamental of Electrical Engineering       | 22212 |
| 11 | Elements of Electronics                     | 22213 |
| 12 | Elements of Electrical Engineering          | 22215 |
| 13 | Basic Electronics                           | 22216 |
| 14 | 'C' programming Language                    | 22218 |
| 15 | Basic Electronics                           | 22225 |
| 16 | Programming in "C"                          | 22226 |
| 17 | Fundamentals of Chemical Engineering        | 22231 |

### **Third Semester:**

|    |   |       |
|----|---|-------|
| 1  | Applied Multimedia Techniques             | 22024 |
| 2  | Advanced Surveying                        | 22301 |
| 3  | Highway Engineering                       | 22302 |
| 4  | Mechanics of Structures                   | 22303 |
| 5  | Building Construction                     | 22304 |
| 6  | Concrete Technology                       | 22305 |
| 7  | Strength Of Materials                     | 22306 |
| 8  | Automobile Engines                        | 22308 |
| 9  | Automobile Transmission System            | 22309 |
| 10 | Mechanical Operations                     | 22313 |
| 11 | Technology Of Inorganic Chemicals         | 22314 |
| 12 | Object Oriented Programming Using C++     | 22316 |
| 13 | Data Structure Using 'C'                  | 22317 |
| 14 | Computer Graphics                         | 22318 |
| 15 | Database Management System                | 22319 |
| 16 | Digital Techniques                        | 22320 |
| 17 | Principles Of Database                    | 22321 |
| 18 | Digital Techniques & Microprocessor       | 22323 |
| 19 | Electrical Circuits                       | 22324 |
| 20 | Electrical & Electronic Measurment        | 22325 |
| 21 | Fundamental Of Power Electronics          | 22326 |
| 22 | Electrical Materials & Wiring Practice    | 22328 |
| 23 | Applied Electronics                       | 22329 |
| 24 | Electrical Circuits & Networks            | 22330 |
| 25 | Electronic Measurements & Instrumentation | 22333 |
| 26 | Principles Of Electronics Communication   | 22334 |
| 27 | Thermal Engineering                       | 22337 |
| 28 | Engineering Matrology                     | 22342 |
| 29 | Mechanical Engineering Materials          | 22343 |
| 30 | Theory Of Machines                        | 22344 |

### **Fourth Semester:**

|    |  |       |
|----|--|-------|
| 1  | Hydraulics   | 22401 |
| 2  | Geo Technical Engineering                            | 22404 |
| 3  | Chemical Process Instrumentation & Control           | 22407 |
| 4  | Fluid Flow Operation                                 | 22409 |
| 5  | Technology Of Organic Chemicals                      | 22410 |
| 6  | Java Programming                                     | 22412 |
| 7  | GUI Application Development Using VB.net             | 22034 |
| 8  | Microprocessor                                       | 22415 |
| 9  | Database Managment                                   | 22416 |
| 10 | Electric Motors And Transformers                     | 22418 |
| 11 | Industrial Measurements                              | 22420 |
| 12 | Digital Electronics And Microcontroller Applications | 22421 |
| 13 | Linear Integrated Circuits                           | 22423 |
| 14 | Microcontroller & Applications                       | 22426 |
| 15 | Basic Power Electronics                              | 22427 |

|    |                                     |       |
|----|-------------------------------------|-------|
| 16 | Digital Communication Systems       | 22428 |
| 17 | Mechanical Engineering Measurements | 22443 |
| 18 | Fluid Mechanics and Machinery       | 22445 |
| 19 | Fundamentals Of Mechatronics        | 22048 |

### **Fifth Semester:**

|    |   |       |
|----|---|-------|
| 1  | Design of Steel and RCC Structures                                      | 22502 |
| 2  | Public Health Engineering   | 22504 |
| 3  | Heat Transfer Operation   | 22510 |
| 4  | Environmental Technology  | 22511 |
| 5  | Operating Systems   | 22516 |
| 6  | Advanced Java Programming   | 22517 |
| 7  | Software Testing  | 22518 |
| 8  | Control Systems and PLC's   | 22531 |
| 9  | Embedded Systems  | 22532 |
| 10 | Mobile and Wireless Communication                                       | 22533 |
| 11 | Industrial Machines   | 22523 |
| 12 | Switchgear and Protection   | 22524 |
| 13 | Energy Conservation and Audit   | 22525 |
| 14 | Power Engineering and Refrigeration                                     | 22562 |
| 15 | Solid Modeling and Additive Manufacturing                               | 22053 |
| 16 | Guidelines & Assessment Manual for Micro Projects & Industrial Training | 22057 |

### **Sixth Semester:**

|    |  |       |
|----|--|-------|
| 1  | Solid Modeling                                 | 17063 |
| 2  | Highway Engineering                            | 17602 |
| 3  | Contracts & Accounts                           | 17603 |
| 4  | Design of R.C.C. Structures                    | 17604 |
| 5  | Industrial Fluid Power                         | 17608 |
| 6  | Design of Machine Elements                     | 17610 |
| 7  | Automotive Electrical and Electronic Systems   | 17617 |
| 8  | Vehicle Systems Maintenance                    | 17618 |
| 9  | Software Testing                               | 17624 |
| 10 | Advanced Java Programming                      | 17625 |
| 11 | Mobile Computing                               | 17632 |
| 12 | System Programing                              | 17634 |
| 13 | Testing & Maintenance of Electrical Equipments | 17637 |
| 14 | Power Electronics                              | 17638 |
| 15 | Illumination Engineering                       | 17639 |
| 16 | Power System Operation & Control               | 17643 |
| 17 | Environmental Technology                       | 17646 |
| 18 | Mass Transfer Operation                        | 17648 |
| 19 | Advanced Communication System                  | 17656 |
| 20 | Mobile Communication                           | 17657 |
| 21 | Embedded System                                | 17658 |
| 22 | Process Control System                         | 17663 |
| 23 | Industrial Automation                          | 17664 |
| 24 | Industrial Drives                              | 17667 |
| 25 | Video Engineering                              | 17668 |
| 26 | Optical Fiber & Mobile Communication           | 17669 |
| 27 | Therapeutic Equipment                          | 17671 |
| 28 | Intensive Care Equipment                       | 17672 |
| 29 | Medical Imaging Equipment                      | 17673 |

### **Pharmacy Lab Manual**

#### **First Year:**

|   |                                     |      |
|---|-------------------------------------|------|
| 1 | Pharmaceutics - I                   | 0805 |
| 2 | Pharmaceutical Chemistry - I        | 0806 |
| 3 | Pharmacognosy                       | 0807 |
| 4 | Biochemistry and Clinical Pathology | 0808 |
| 5 | Human Anatomy and Physiology        | 0809 |

#### **Second Year:**

|   |                                |      |
|---|--------------------------------|------|
| 1 | Pharmaceutics - II             | 0811 |
| 2 | Pharmaceutical Chemistry - II  | 0812 |
| 3 | Pharmacology & Toxicology      | 0813 |
| 4 | Hospital and Clinical Pharmacy | 0816 |

## **HEAD OFFICE**



Secretary,  
Maharashtra State Board of Technical Education  
49, Kherwadi, Bandra (East), Mumbai - 400 051  
Maharashtra (INDIA)  
Tel: (022)26471255 (5 -lines)  
Fax: 022 - 26473980  
Email: [-secretary@msbte.com](mailto:-secretary@msbte.com)  
**Web -[www.msbte.org.in](http://www.msbte.org.in)**

### **REGIONAL OFFICES:**

#### **MUMBAI**

Deputy Secretary (T),  
Mumbai Sub-region,  
2<sup>nd</sup> Floor, Govt. Polytechnic Building,  
49, Kherwadi, Bandra (East)  
Mumbai - 400 051  
Phone: 022-26473253 / 54  
Fax: 022-26478795  
Email: [rbtermumbai@msbte.com](mailto:rbtermumbai@msbte.com)

#### **PUNE**

Deputy Secretary (T),  
M.S. Board of Technical Education,  
Regional Office,  
412-E, Bahirat Patil Chowk,  
Shivaji Nagar, Pune  
Phone: 020-25656994 / 25660319  
Fax: 020-25656994  
Email: [rbtepn@msbte.com](mailto:rbtepn@msbte.com)

#### **NAGPUR**

Deputy Secretary (T),  
M.S. Board of Technical Education  
Regional Office,  
Mangalwari Bazar, Sadar, Nagpur - 440 001  
Phone: 0712-2564836 / 2562223  
Fax: 0712-2560350  
Email: [rbteng@msbte.com](mailto:rbteng@msbte.com)

#### **AURANGABAD**

Deputy Secretary (T),  
M.S. Board of Technical Education,  
Regional Office,  
Osmanpura, Aurangabad -431 001.  
Phone: 0240-2334025 / 2331273  
Fax: 0240-2349669  
Email: [rbteau@msbte.com](mailto:rbteau@msbte.com)