Laboratory Manual

for

Operating Systems with Linux Lab.

(MCA-163)

MCA - I Semester

Compiled by:

Dr. Sunil Pratap Singh

(Associate Professor, BVICAM, New Delhi)

Bharati Vidyapeeth's
Institute of Computer Applications and Management (BVICAM)

A-4, Paschim Vihar, Rohtak Road, New Delhi-63

Visit us at: www.bvicam.in

Index

List of Abbreviations

BTL	Bloom's Taxonomy Level
CE	Communication Efficacy
CICP	Conduct Investigations of Complex Computing Problems
CK	Computational Knowledge
CO	Course Outcome
DAC	Departmental Advisory Committee
DDS	Design and Development of Solutions
I&E	Innovation and Entrepreneurship
I&T	Individual & Team Work
IQAC	Internal Quality Assurance Cell
LLL	Life-Long Learning
MTU	Modern Tool Usage
PA	Problem Analysis
PE	Professional Ethics
PEO	Programme Educational Objective
PMF	Project Management and Finance
PO	Programme Outcome
SEC	Societal and Envoirnmental Concern
SED	Stream Editor

Declaration

Department	:	Department of Computer Science and Applications
Course, Year and the Semester to which Lab is offered	:	MCA - I Year, I Semester
Name of the Lab Course	:	Operating Systems with Linux Lab.
Course Code	:	MCA-163
Version No.	:	
Name of Course/Lab Teacher	:	Dr. Sunil Pratap Singh
Laboratory Manual Committee	:	1. Dr. Ritika Wason, Chairperson 2. Dr. Rakhee, Member 3. Mr. Uttam Singh Bist, Member 4. Prof. P. S. Grover, Margdarshak 5. Mr. Amit Sharma, Alumni & Industry Expert 6. Dr. Sunil Pratap Singh, Concerned Subject Teacher, Convener
Approved by	:	DAC
Approved by	:	IQAC

Signature (Course Teacher)	**Signature** (Head of Department)	**Signature** (IQAC Coordinator)

1. Vision of the Department

To become a centre of excellence in the field of Computer Science and Applications to produce quality professionals in software development.

2. Mission of the Department

M1 To produce quality software professionals as per global industry standards.

M2 To foster innovation, entrepreneurial skills, research capabilities and bring all-round development amongst budding professionals.

M3 To promote analytical and collaborative life-long learning skills, among students and faculty members.

M4 To inculcate strong ethical values and professional behaviour while giving equal emphasis to social commitment and nation building.

3. Programme Educational Objectives (PEOs)

The PEO's for the MCA programme are as follows:

PEO1 Exhibit professional competencies and knowledge for being a successful technocrat.

PEO2 Adopt creative and innovative practices to solve real-life complex problems.

PEO3 Be a lifelong learner and contribute effectively to the betterment of the society.

PEO4 Be effective and inspiring leader for fellow professionals and face the challenges of the rapidly changing multi-dimensional, contemporary world.

4. Programme Objectives (POs)

PO1: Computational Knowledge (CK)

Demonstrate competencies in fundamentals of computing, computing specialisation, mathematics, and domain knowledge suitable for the computing specialisation to the abstraction and conceptualisation of computing models from defined problems and requirements.

PO2: Problem Analysis (PA)

Identify, formulate, and analyze complex real-life problems in order to arrive at computationally viable conclusions using fundamentals of mathematics, computer sciences, management and relevant domain disciplines.

PO3: Design and Development of Solutions (DDS)

Design efficient solutions for complex, real-world problems to design systems, components or processes that meet the specifications with suitable consideration to public health, and safety, cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Computing Problems (CICP)

Ability to research, analyze and investigate complex computing problems through design of experiments, analysis and interpretation of data, and synthesis of the information to arrive at valid conclusions.

PO5: Modern Tool Usage (MTU)

Create, select, adapt and apply appropriate technologies and tools to a wide range of computational activities while understanding their limitations.

PO6: Professional Ethics (PE)

Ability to perform professional practices in an ethical way, keeping in mind cyber regulations & laws, responsibilities, and norms of professional computing practices.

PO7: Life-Long Learning (LLL)

Ability to engage in independent learning for continuous self-development as a computing professional.

PO8: Project Management and Finance (PMF)

Ability to apply knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects in multidisciplinary environments.

PO9: Communication Efficacy (CE)

Ability to effectively communicate with the technical community, and with society at large, about complex computing activities by being able to understand and write effective reports, design documentation, make effective presentations, with the capability of giving and taking clear instructions.

PO10: Societal and Envoirnmental Concern (SEC)

Ability to recognize and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities applicable to professional computing practices.

PO11: Individual & Team Work (I&T)

Ability to work in multi-disciplinary team collaboration both as a member and leader as per need.

PO12: Innovation and Entrepreneurship (I&E)

Ability to apply innovation to track a suitable opportunity to create value and wealth for the betterment of the individual and society at large.

5. Institutional Policy for Students' Conduct

The following guidelines shall be followed:-

5.1 All the students in their introductory Lab. shall be assigned a system, which shall be their workplace for the complete semester. Students can store records of all their Lab. assignments on their individual workstations.

5.2 Introductory Lab. shall include an introduction to the appropriate software/tool, followed by a basic Introductory Assignment having Practice Questions. All the students are expected to complete this assignment within a week time, as the same shall be assessed through a lab. test.

5.3 Each week the instructor, in parallel to respective topics covered in the theory lecture, shall assign a set of practical problems to the students in form of Assignments (A, B, C, ...). The problems in these assignments shall be divided into two parts. The first set of Problems shall be compulsory for all the students and its record need to be maintained in the Prcatical File, having prescribed format, as given in Appendix-A. All the students should get the weekly assigntment checked and signed in the Practical File by the respective teacher in the immediate succeeding week. The second sets of problems are Advanced Problems and shall be optional. Student may solve these advanced problems for their further practice.

5.4 Cellular phones, pagers, CD players, radios and similar devices are prohibited in the classrooms, laboratories and examination halls.

5.5 Laptop-size Computers / Tablets may be used in lectures for the purpose of taking notes or working on team-projects.

5.6 The internal practical exam shall be conducted towards the end of the semester and shall include the complete set of Lab exercises conducted as syllabus. However, students shall be assessed on continuos basis through overall performances in regular lab. tests, both announced and surprise and viva-voce.

5.7 The respective faculty shall prepare and submit sufficient number of

practical sets of computing problems to the Dean (Examinations), atleast two weeks prior to the actual exam. It is the responsibility of the faculty to ensure that a set should not be repeated for more than 5 students in a given batch.

5.8 The exam shall be of 3 hours duration where the student shall be expected to implement solutions to his/her assigned set of problems on appropriate software tools in the lab.

5.9 Once implemented, student shall also appropriately document code implemented in the assigned answer sheets, which shall be submitted at the end of the examination. All the students shall also appear for viva-voce examination during the exam.

5.10 Co-operate, Collaborate and Explore for the best individual learning outcomes but copying or entering into the act of plagiarism is strictly prohibited.

6. Learning Outcomes of Laboratory Work

The student shall demonstrate the ability to:

- Verify and Implement the concepts and theory learnt in class.
- Code and use Software Tools to solve problems and present their optimal solutions.
- Apply numerical/statistical formulas for solving problems/questions.
- Develop and apply critical thinking skills.
- Design and present Lab as well as project reports.
- Apply appropriate methods for the analysis of raw data.
- Perform logical troubleshooting as and when required.
- Work effectively as a member of a team in varying roles as need be.
- Communicate effectively, both oral and written.
- Cultivate ethics, social empathy, creativity and entrepreneurial mindset.

7. Course/Lab Outcomes (COs)

CO1 Build the Linux operating system and configure it. [BTL3]

CO2 Discover Linux commands for working with Linux Environment. [BTL4]

CO3 Appraise the Process Management algorithms, Process Management system calls, Inter Process Communication and CPU Scheduling algorithms. [BTL5]

CO4 Create programs using systems calls for memory management and File Management in C programming, also simulate Deadlock avoidance algorithm using C. [BTL6]

8. Mapping of CO's with PO's

Table 1: Mapping of CO's with PO's

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓							
CO2	✓	✓	✓	✓	✓	✓	✓					
CO3	✓	✓	✓	✓	✓	✓	✓		✓	✓		
CO4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

9. Course/Lab Description

Course (Lab) Title	:	Operating Systems with Linux Lab
Course (Lab) Code	:	MCA-163
Credits	:	01
Pre-requisites	:	Fundamentals of Computer Systems, Introduction to Programming
Academic Session	:	July to December
Contact Hours/Week	:	02 (01 Labs of 02 hours/Week)
Internal Assessment	:	40 Marks
External Assessment	:	60 Marks

10. Grading Policy

Item	Points	Marks	Remarks
Weekly Lab Assignments including Practical Files	10	10	Closed Book/Open Book
Internal End-Term Practical Examination	20	10	Closed Book
Viva-Voce	20	20	Closed Book
External End-Term Examinations	60	60	Closed Book (conducted and evaluated by the University)
	Total	**100**	

11. Lesson Plan

Week No.	Lab No.	Topics/Concepts to be Covered
1.	1.	Installation of VirtualBox and Linux
2.	2.	Basic Commands of Linux OS.
3.	3.	Basic Commands of Linux OS.
4.	4.	Process Management Commands and System Calls
5.	5.	Process Management Commands and System Calls
6.	6.	Memory Management Commands and System Calls
7.	7.	Memory Management Commands and System Calls
8.	8.	File Management Commands and System Calls
9.	9.	File Management Commands and System Calls
10.	10.	File Management in C programming
11.	11.	File Management in C programming
12.	12.	File Management in C programming
13.	13.	Buffer reserved for revision

12. Assignments

P1 Install VirtaulBox and then configure Linux (Ubantu) in VirtualBox. (CO1)

P2 Run `ps` and note the PID of your shell. Log out and log in again and run `ps` again. What do you observe? (CO2)

P3 Enter the following commands, and note your observations: (i) `who` and `tty`, (ii) `tput clear`, (iii) `id`, (iv) `ps` and `echo $$`. (CO2)

P4 Run the following commands, and then invoke `ls`. What do you conclude? (CO2)

```
echo > README [Enter]
echo > readme [Enter]
```

P5 Create a directory, and change to that directory. Next, create another directory in the new directory, and then change to that directory too. Now, run `$ cd` without any arguments followed by `pwd`. What do you conclude? (CO2)

P6 Create a file `mca` containing the words "Hello MCA Class!". Now create a directory `bvicam`, and then run `mv mca bvicam`. What do you observe when you run both `ls` and `ls bar`? (CO2)

P7 Run `$ who am i` and then interpret the output. (CO2)

P8 Find out whether the following commands are internal or external: `echo`, `date`, `pwd`, and `ls`. (CO2)

P9 Display the current date in the form `dd/mm/yyyy`. (CO2)

P10 Both of the following commands try to open the file `mca`, but the error messages are a little different. What could be the reason? (CO2)

```
$ cat mca
cat: mca: No such file or directory
$ cat < mca
bash: mca: No such file or directory
```

P11 Run the following commands, and discuss their output? (CO2)

(a) `$ uname`

(b) `$ passwd`

(c) `$ echo $SHELL`

(d) `$ man man`

(e) `$ which echo`

(f) `$ type echo`

(g) `$ whereis ls`

(h) `$ cd`

(i) `$ cd $HOME`

(j) `$ cd ~`

P12 Frame `ls` command to (i) mark directories and executables separately, and (ii) also display hidden files. (CO2)

P13 Find out the result of following: (CO2)

```
$ cat mca mca mca
```

P14 Run the following and determine which commands will work? Explain with reasons. (CO2)

(a) `$ mkdir a/b/`

(b) `$ mkdir a a/b`

(c) `$ rmdir a/b/c`

(d) `$ rmdir a a/b`

(e) `$ mkdir /bin/mca`

P15 How does the command `mv mca1 mca2` behave, where both `mca1` and `mca2` are directories, when (i) `mca2` exists, (ii) `mca2` doesn't exist? (CO2)

P16 Assuming that you are positioned in the directory `/home/bvicam`, what are these commands presumed to do, and explain whether they will work at all: (CO2)

(a) `$ cd ../..`

(b) `$ mkdir ../bin`

(c) `$ rmdir ..`

(d) `$ ls ..`

P17 Apply Peterson algorithm for solving the critical section problem with C/Java multi-threaded programming. Assume appropriate code snippet for critical section. (CO3)

P18 Apply Bakery algorithm for synchronization of processes/threads in a C/Java program. Assume appropriate code snippet for critical section. (CO3)

P19 Write C/Java program to simulate and solve the Producer-Consumer problem. (CO3)

P20 Implement Semaphore(s) in a C/Java-multithreaded program to simulate the working and solution of Reader-Writer problem. Assume multiple readers and writers. (CO3)

P21 Create a zombie process and an orphan process in a 'C' program with appropriate system calls. (CO3)

P22 Write a 'C' program which creates a new process and allows both, child and parent, to report their identification numbers (ids). The parent process should wait for the termination of the child process. (CO3)

P23 Write two 'C' programs (A.c and B.c) where one program (A.c) creates a child process and then that child process executes the code of other program (B.c). The logic of program 'B.c' is to generate all the prime numbers within the specified limit. (CO3)

P24 Write an appropriate 'C' program which implements the concept of dynamic memory allocation (use of malloc(), calloc(), realloc(), and free() system call). (CO4)

P25 Create a text file, named as 'courses.txt' that contains the following four lines:

Java Programming

Operating System

Discrete Structure

Write a 'C' program that forks three other processes. After forking, the parent process goes into wait state and waits for the children to finish their

execution. Each child process reads a line from the 'course.txt' file (Child 1 Reads Line 1, Child 2 Reads Line 2, and Child 3 Reads Line 3) and each prints the respective line. The lines can be printed in any order. (CO4)

P26 Write a 'C' program (using appropriate system calls of Linux) that generates 'n' integers and stores them in a text file, named as 'All.txt'. Then, retrieve the stored integers from this file and copy to "Odd.txt" and 'Even.txt' based upon the type of number, i.e. if the retrieved integer if odd number then store in 'Odd.txt' file or if the retrieved integer is even then store in 'Even.txt' file. Finally, display the contents of all three files on the screen. (CO4)

P27 Write a program in 'C' which accepts the file or directory name and permission (access rights) from the user and then changes the access rights accordingly. Use appropriate system call(s) of Linux. (CO4)

P28 Write a 'C' program (using appropriate system calls of Linux) which generates and stores the characters from 'a' to 'z'. Then, display the stored characters in alternative manner, like: a, c, e, g, …, etc. (CO4)

P29 Write a 'C' program (using appropriate system calls of Linux) which receives roll number and names of 'n' students, from the user one-by-one and then stores them in a text file, named as 'Student.txt'. After inserting all 'n' roll numbers and names, display the contents of file. Also, display the access rights of the file 'Student.txt'. (CO4)

P30 Demonstrate the use of following system calls by writing an appropriate 'C' program. (CO4)

(a) lseek()
(b) chmod()
(c) umask()
(d) access()
(e) utime()

Appendix – A: Index of Lab File

Week No.	Lab. Ex. No.	Detailed Description of the Lab Exercise	Outcome Mapping		Page No./Link of Online Document	Signature of Teacher with Date
			CO	BTL		
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

Note: *The students should use header and footer, mentioning their roll number & name in header and page number in footer.*