UNIX AND SHELL PROGRAMMING LABORATORY	
Course Code: CIL37	**Credits:** 0:0:1
Pre – requisites: Nil	**Contact Hours:** 14P
Course Coordinator: Shubha Vibhu Malige	

COURSE CONTENTS

List of Lab Programs

Vi/Vim editor operations

1. Introduction to Unix and Shelling Programming, Vi/Vim editor:

Consider the following content and edit using Vi/Vim editor:

"The simplest way to understand how AI and ML relate to each other is: AI is the broader concept of enabling a machine or system to sense, reason, act, or adapt like a human. ML is an application of AI that allows machines to extract knowledge from data and learn from it autonomously.

Artificial Learning is the capability of a computer system to mimic human cognitive functions such as learning and problem-solving. Through AI, a computer system uses math and logic to simulate the reasoning that people use to learn from new information and make decisions

Cyber security is primarily about people, processes, and technologies working together to encompass the full range of threat reduction, vulnerability reduction, deterrence, international engagement, incident response, resiliency, and recovery policies and activities, including computer network operations, information assurance, law enforcement, etc."

a. Execute the following operations:

i. Open the file in vi/vim editor and type the above given content and save the file by name test1.txt, continue working in the opened file.
ii. Navigate through the 10^{th} line and go to save mode and write the 10^{th} line to another file named test2.txt and continue working with test1.txt
iii. Find the Keyword AI and replace AI with Artificial Learning using interactive substitution and save the current file.
iv. Delete 5^{th} line to a buffer (a), save the current file. Now switch to the file test2.txt, move to a desired location and copy the text. Toggle between previous file and current file.
v. Navigate to the first line in test.txt file, and go to 15^{th} character towards right and move 3 lines down and 2 characters left and replace the character with M and save the current file and exit.

vi. In the test2.txt file, move to 3 words forward in the current line and take the cursor to 2 words back, delete the word in the cursor position and navigate yourself to move to the line extremes, save and escape to the shell.

vii. In the test1.txt, join first 3 lines (Join the current line with 2 lines).

viii. Delete the 7th line in current file and exit the vi editor by saving and quitting.

b. Correct the below given c program in Vi/Vim editor using vi commands and execute the corrected c program.

Sample.c(Before Correction—with errors)	Sample.c(After Correction--Right)

#include<stio.h>
#include errno.h
int test(int * message)
{
print("Errno is %8d',errno)
exit;
	#include<stdio.h>
	#include <errno.h>
	void test(char * message)
	{
	print("Errno is %8d\n",errno);
	exit(1);
	}

Shell Scripting

2. Consider the below scenarios and execute the given shell scripts.

"*Ramaiah College is having 10 departments (Say, CS, IS, AI, ML, Cyber Security, EC, Mechanical, EEE, DS, Civil) with UG and PG programs, and in each of the program student details, course details are maintained in 10 different files (such as Student Details, Course details, Curriculum, Exam, Marks, Research Activity, NBA, Placement Activities, Library Details, Extra Activities….)*."

a. Develop a shell script for the above scenario to create 10 levels of folders for the departments and inside each level(department) of folder, create 10 files in each department for maintaining student details. Display the entire hierarchy on the standard output by using tree command.

b. Develop a shell script that accepts above created filename as argument and display its creation time and permissions of the file, on the standard output.

3. a. Develop a shell script that takes a valid directory name as an argument and recursively descend all the sub-directories, finds the maximum length of any file in that hierarchy, and store the output in a file.

b. Create a shell script to find a file with particular name, (show separate outputs for both the conditions)

- if that file exists then rename the existing file and create an empty file with that name.
- if that file does not exist then create a new empty file.
- If both the conditions done together.

c. Set a cronjob for above developed scripts, that will be execute after every 30 minutes.

d. Illustrating the shell variables in a shell script.

4. a. Build a shell script to display the system space used. If it is greater than 80%, display as Low system Space and list the files having size greater than 1GB. Set up a cron job for the above developed script to execute every Monday morning 10AM.

b. Write a shell program to count number of words, characters, white spaces, and special symbols in each text and display the output on standard output. Set a cronjob to execute above script every 3rd day of week morning 9 AM.

AWK

5. a. Develop an awkscript that accepts date argument in the form of dd-mm-yy and display it in the form month, day, and year. The script should check the validity of the argument and in the case of error, display a suitable message.

b. Develop an awkscript to delete duplicated line from a text file. The order of the original lines must remain unchanged.

c. Set up a cron job for the above developed scripts to execute every other day evening 4 PM.

Sed and Find

6. a. Type the below given text and save the file as 1.txt using Vi/Vim editor. Perform the below given operations.

"Python is a very popular language.
Python is easy to use. Python is easy to learn.
Python is a cross-platform language
HTML is a Markup Language

Python Programming Language
C Programming Language
Shell Programming
Perl Programming Language
Bash"

i. Replace all instances of a *Python* in a second line of 1.txt with *Perl*.
ii. Replace the last occurrence of *Programming* with *Scripting* only of a match, not other instances.
iii. Create a text file in the path /MSRIT/CSE/UG/Python.txt. Replace full path with just the filename no directory (such as Python.txt) and display it on standard output.
iv. Add string before and after the matching pattern using '\1'. In the above given text, navigate yourself to last line, you can find *Bash* keyword, Add *Learn* before *Bash* and *Programming* after *Bash* keyword.

b. Perform the following execution using find command

i. Find all the files in a current directory, whose permissions are 0777.
ii. Assign a sticky bit to all the files in a current directory.
iii. Find Directories with full permissions (777) and by using chmod command change the permissions by assigning read, write and execute permissions to owner and only read & execute to group and others.
iv. Find last 20 days modified files, accessed files.
v. Find all the files which are modified in last 1 hour.

C Programs by using UNIX File System Calls

7. a. Develop a C program to emulate the UNIX ls –li command, which lists all the attributes of the files in a specified directory.
   b. Write a C program to remove empty files from the given directory using system calls.

8. a. Write a C program to read n characters from a file and append them back to the same file using dup2 function.

   b. Write a C program to list all files in a current directory using systemcalls
.

9. a. Create a C program to simulate the copy command in Unix (cp command)

   b. Develop a C program to simulate the ls (list) Unix command using system calls.

10. a. Understanding File Descriptors and Building a C program to create two processes P1 and P2. P1 takes a string and passes it to P2. P2 concatenates the received string with another string without using string function and sends it back to P1 for printing, send the output to standard output.

b. Create a C program to simulate Grep Unix Command using system calls.

11. **XML Integration**

Consider the student details given below and create an XML file and save it as sample.xml.

Student Details: -

SI. No.	Student Name	USN	Department Name
1.	Alex	1RITCS001	CSE
2.	Smith	1RITDS040	DS
3.	Saliena	1RITCV051	Civil
4.	Elizabeth	1RITME011	Mechanical

Department Details: -

SI. No.	Department Name	DepartmentId	Total no of enrolled students
1.	CSE	01	170
2.	DS	02	188
3.	Civil	03	160
4.	Mechanical	04	150

a. From the above XML file, separate only student details and redirect the output to a file.
b. Replacing the tag name from name to Dept name and change globally in a shell script.

c. Read the tag value of USN and redirect the output to standard output as well as redirect to a file.

12. **Docker**

a. Docker installation and set up
b. Create a shell script to pass arguments and run in a container using Docker.
c. Creating docker file
d. Building docker image, creating a container and running the shell scripts.
e. Displaying and running docker image.

***Note: Practical sessions will be based on the contents.**

Suggested Learning Resources

Text Books:

1) Introduction to UNIX & SHELL programming, M.G. Venkatesh Murthy, Pearson Education.
2) Unix concepts and applications, Fourth Edition, Sumitabha Das, TMH.
3) Unix for programmers and users, 3rd edition, Gaham Glass & K. Ables, Pearson education.

Course Outcomes:

At the end of the course students will be able to:

1. Understand the basics of Unix concepts and commands, dockers, containers and system calls of Unix.
2. Apply the commands to incorporate changes to the file system.
3. Implement the Shell scripts based on the real time scenarios.

Course Assessment and Evaluation:

Parameter	**Marks**
CIE Test	20
Lab Record Writing + Viva+ program execution	30
Total	**50**
Final Exam will be conducted for 50 marks (SEE)	

II. LAB EXERCISES

Prg:1. a.	Vi/Vim Editor Operations
	Performing the operations using Vi/Vim editor commands

Vi/Vim Editor Operations:

vi test1.txt

Type the below given text in the insert mode (by pressing i,I,a,A,eE,o,O,u,U)

The simplest way to understand how AI and ML relate to each other is: AI is the broader concept of enabling a machine or system to sense, reason, act, or adapt like a human. ML is an application of AI that allows machines to extract knowledge from data and learn from it autonomously.

Artificial Learning is the capability of a computer system to mimic human cognitive functions such as learning and problem-solving. Through AI, a computer system uses math and logic to simulate the reasoning that people use to learn from new information and make decisions

Cyber security is primarily about people, processes, and technologies working together to encompass the full range of threat reduction, vulnerability reduction, deterrence, international engagement, incident response, resiliency, and recovery policies and activities, including computer network operations, information assurance, law enforcement, etc

i. Press <Esc> and **:w** (Saves file and remains in editing mode)

ii. Press <Esc> and **:10w test2.txt**

iii. :1,$ s/AI/Artificial Learning/gc

It prompts **replace with fprintf (y/n/a/q/^E/^Y)?** to interactively substitute

iv. Go to the line 5--------**5G**

Delete the text into buffer a.-------**"add**(Current line will be deleted and stored in register a)
then ----- **:w**
: e test2.txt (To go the test2.txt file)
Navigate to the desired location in the file test2.txt then press <Esc>
" **ap**(Whatever is stored in register a, will be placed in the desired location)
To toggle between 2 files test1.txt and text2.txt press [ctrl ^] or **:e#**

v. Go the line number one-------**1G**
To move 15 characters right type **15l**
Then go to 3rd line ----**3j**
To Move left---**2h**
<Esc> **rM** (to replace)
:w (To save file)

vi. **:e test2.txt** (To go to file test2.txt)
3 words forward---- **3e**
2 words backward**---2b**
Delete the word**(X)** in the current cursor position.
Go to the end of the line ---**$**
and **: w**

vii. To go the file test1.txt--- **:e test1.txt**
Go the the line number 1---**1G**
Then **2J** joins 2 lines with current line.

viii. Press <Esc> **7G** and Press **dd,** and **:wq!** (to saving and exiting vi editor)

Prg:1. b.	**Vi/Vim Editor Operations**	
	Correct the below given c program in Vi/Vim editor using vi commands and execute the corrected c program.	
	Sample.c(Before Correction—with errors)	Sample.c(After Correction--Rigl

	#include<stio.h>
	#include errno.h
	int test(int * message)
	{
	print("Errno is %8d',errno)
	exit;
		#include<stdio.h>
		#include <errno.h>
		void test(char * message)
		{
		print("Errno is %8d\n",errno);
		exit(1);
		}

vi Sample.c

Type the corresponding program, which is mentioned at the left side (Before Correction—with errors)

Press <Esc> and :w

Perform below mentioned operations

Command	*Action*
1G	Moves to line 1
2w	Moves to s in stdio.h
i<*[Esc]*	Inserts <; shows <stdio.h
A>*[Esc]*	Appends >; shows <stdio.h>
2j	Move two lines below
3svoid*[Esc]*	Replaces int with void
j2e	Moves to first ' on next line
r"	Replaces ' with "
4w	Moves to 1 in 10
2x	Deletes 10 ; shows %d
a\n□*[Esc]*	Appends \n and a space to %d
l	Moves to closing '
r"	Replaces ' with "
A;*[Esc]*	Appends ; at end of line
j$	Moves to end of next line; at ; in exit;
i(1)*[Esc]*	Inserts (1) after t in exit
o}*[Esc]*	Opens line below and inserts }
:x	Saves and quits **vi**

Prog 2. a.	Consider the below scenarios and execute the given shell scripts. "*Ramaiah College is having 10 departments (Say, CS, IS, AI, ML, Cyber Security, EC, Mechanical, EEE, DS, Civil) with UG and PG programs, and in each of the program student details, course details are maintained in 10 different files (such as Student Details, Course details, Curriculum, Exam, Marks, Research Activity, NBA, Placement Activities, Library Details, Extra Activities….)*."
	Develop a shell script for the above scenario to create 10 levels of folders for the departments and inside each level(department) of folder, create 10 files in each department for maintaining student details. Display the entire hierarchy on the standard output by using tree command.

Shell Script:

```
#!/bin/bash
echo " 10 levels of folders  are created for the departments and  10 levels of files
created for student details"
i=1
while [ $i -le 10 ]
do
mkdir MSRITDept$i
cd MSRITDept$i
j=1
while [ $j -le 20 ]
do
touch MSRITStudentDetails$j
j=$(($j+1))
done
cd ..
i=$(($i+1))

done
```

Output:

```
 # sh 2a.sh
```

Note: Then install tree command:
#sudo apt install tree

Once the tree command is installed.. Then use tree to display above hierarchical structure of created directories and files.

tree

Prog 2. b.	**Shell Scripting**
	Develop a shell script that accepts above created filename as argument & display its creation time and permissions of the file, on the standard output

Shell Script

```
! /bin/bash
if [ $# -eq 0 ]
then
echo "display does not exit"
else
ls -l $1> t1
x=`cut -d ' ' -f  1,6,7,8,9  t1`
echo $x
fi
```

Output:-

```
# sh 2b.sh 2aa.sh
-rw-rw-r-- Aug 12 10:55
```

	Shell Scripting
Prog 3. a.	Develop a shell script that takes a valid directory name as an argument and recursively descend all the sub-directories, finds the maximum length of any file in that hierarchy, and store the output in a file.

Shell Script:

```
#! /bin/bash
 for i in $*
   do
     if [ -d $i ]
       then
          echo "large filename size is"
          echo `ls -lR $1 | grep "^-" | tr -s ' ' | cut -d' ' -f 5 | sort -n| tail -1`
      else
          echo "not directory"
      fi
     done
```

Output:

```
  #sh 3a.sh dir1
```

Now the output will be displayed on the monitor(Standard Output).

Note:- In order to store the output in a file

```
echo `ls -lR $1 | grep "^-" | tr -s ' ' | cut -d' ' -f 5 | sort -n| tail -1` >out
```

To be modified.

If we open the filename **out ,** then we can able to see the largest size of file.

Prog 3. b.	**Shell Scripting**
	Create a shell script to find a file with particular name, (show separate outputs for both the conditions) • if that file exists then rename the existing file and create an empty file with that name. • if that file does not exist then create a new empty file. • If both the conditions done together.

Create a file named rit.txt with below contents

Hai
Hello
Welcome to MSRIT

Shell Script

```
#! /bin/bash
cd /home/ritadmin/sample
filename="rit.txt"
if [ -e $filename ]
then
echo "moving the contents of rit.txt to rit.txt_old"
mv -f  $filename $filename"_old"
touch $filename
fi
```

Output:-

```
#sh 3b.sh
moving the contents of rit.txt to ritdup.txt
ritadmin@ThinkCentre-M70t:~/sample$ ls
```

```
3b.sh  rit.txt_old  rit.txt
ritadmin@ThinkCentre-M70t:~/sample$ cat rit.txt_old
Hai

Hello
Welcome to MSRIT
ritadmin@ThinkCentre-M70t:~/sample$ cat rit.txt
ritadmin@ThinkCentre-M70t:~/sample$
```

Shell Script

```
#! /bin/bash
cd /home/ritadmin/sample
filename="rit.txt"
if [ ! -e $filename ]
then
        echo " File doesnot exist..so creating a empty file"
        touch $filename
fi
```

Output:

```
#sh 3bb.sh
Filename doesnot exist creating an empty file”
tadmin@ThinkCentre-M70t:~/sample$ ls
3b.sh   3bb.sh rit.txt
```

Shell Script

```
#! /bin/bash
cd /home/ritadmin/sample
filename="rit.txt"
if [ -e $filename ]
then
mv -f $filename $filename"_old"
touch $filename
else
echo “File does not exist ---Creating”
touch $filename
fi
```

Output:-

```
#sh 3bb2.sh
Filename does exist, so moving filename with other name”
tadmin@ThinkCentre-M70t:~/sample$ ls
3b.sh   3bb.sh rit.txt 3bb2.sh rit.txt_old
```

Prog 3. c.	**Shell Scripting**
	Set up a cron job for the above developed scripts, , that will be execute after every 30 minutes

The Cron software utility is a time-based job scheduler in Unix-like operating systems. Cron allows Linux and Unix users to run commands or scripts at a given time and date.

Installation Steps:

1. #apt-get update && #apt-get upgrade
2. #apt-get install cron
3. #systemctl status cron
4. #nano /etc/crontab
5. #crontab -e //install your cron job by running this command.

crontab -l : List the all your cron jobs.
crontab -r : Delete the current cron jobs.

```
0, 30  *  *  *  *   3b.sh
0, 30  *  *  *  *   3bb.sh
0, 30  *  *  *  *   3bb2.sh
```

Output

Prog 3. d.	**Shell Scripting**
	Illustrating shell variables in a shell script

Shell Variables:

Variables that are set by shell itself and help shell to work with functions correctly. It contains both, which means it has both, some variables are Environment variable, and some are Local Variables.

For example:
`$PWD` = Stores working directory
`$HOME` = Stores user's home directory
`$SHELL` = Stores the path to the shell program that is being used.

Shell Variable is used in shell scripts for many functionalities like storing data and information, taking input from users, printing values that are stored. They are also used for storing data temporarily and storing output of commands.

a. Accessing Variable

```
#!/bin/bash

VAR_1="Devil"
VAR_2="OWL"

echo "$VAR_1$VAR_2"
```

b. Storing user data in a variable.

```
#!/bin/bash
echo "Enter the length of the rectangle"
read length
echo "Enter the width of the rectangle"
read width
area=$((length * width))
echo "The are of the rectangle is: $area"
```

Special Variables:

$# ----------This parameter represents the number of arguments passed to the shell script.

$0---------- This parameter represents the script name.

$i---------- This parameter represents the ith argument passed to the shell script like $1, $2

$*---------- This parameter gives all arguments passed to the shell script separated by the space.

$!---------- This parameter gives PID of the last background running process

$?---------- This parameter represents the exit status of the last command that executed. The 0 code represents success and 1 represents failure.

$$---------- This parameter gives the PID of the current shell.

$@---------- This parameter holds all argument passed to the script and treat them as an array. It is similar to the $* parameter

Shell Script:

```
#! /bin/bash
# To understand special variables

echo " '$*' output is $*"
echo "'$#' output is $#"
echo "'$1  & $2' output is $1 and $2"
echo "'$@' output is $@"
echo "'$?' output is $?"
echo "'$$' output is $$"
sleep 400 &
echo "'$!' output is $!"

echo "'$0' your current program name is $0"
```

Output:

#sh 3d.sh execute unix and shell programs
‘execute unix and shell programs’ output is execute unix and shell programs
‘5’ output is 5
‘execute & unix’ output execute and unix
‘execute unix and shell programs ‘output of execute unix and shell programs
‘0’ output is 0
‘2018’ output is 2018
‘2019’ output 2019
‘3d.sh’ your current program is 3d.sh