Date: 31/11/2023

Academic Year: 2022-23

Course & Code: Operating System (OSY-22516)

Semester & Code: CM-5-I Class: TYCM-Lin, Win, Mac Sem: V(ODD)

Question Bank For Practical End Semester Examination Winter-2023

1. Write steps to install Linux Operating System.
2. How you will record all the activities performed by the user on terminal?

3. Give a command to display calendar for month of January 2025.
   Ans. cal 1 2025

4. Find your weekday of birth in the year 2030.
   Ans. date -d "2030-10-28" "+%A"

5. Give single statement command to display the calendar of previous current & next month.
   Ans. cal -3

6. Give the command to display full week day.
   Ans. date "+%A"

7. Give command for present working directory.
   Ans. pwd

8. List currently logged in users using command.
   Ans. who

9. Acquire the status of super user.
   Ans. Su

10. Write use and output for following commands
    i) $who; clear; who am i
    ii) $who; tty; date
11. What is name of your login shell?

12. How to take help on any command in Linux.
    $man command
13. Display
    a. 2 digit year
       Ans. date "+%y"
    b. full month name
       Ans. date "+%B"
    c. date in the form of dd/mm/yyyy
       Ans. date "+%d/%m/%y"
    d. current time in hh:mm:ss format
       Ans. date "+%T"
14. Display the statement "Todays century is _"
    Ans. date "+Todays century is %C"
15. What are various options of kill command? Execute and write output.
    Ans. Page 20
16. What are various options of ps commands? How to display PPID?
    Ans. Page 20
17. Explain about exit command.
    Ans. The exit command in Linux is used to terminate/quit a shell or a script. It is a built-in command that allows you to exit the current shell session or terminate a script at a specific point.

18. What are different options of ls command? Write down the command along with the option and write the output.
    Ans. Page 24 & 25
19. What is the use of mv command? What are different options of mv command?
    Ans. Page 25
20. What is use of split command? Create one file named "KKWP" with 10 lines and split it into 5 files with tag name "Nashik".
    Use: Page 26
    Cat > kkwp
    Abc
    Xyz
    Pqr
    .
    .
    .
    10 lines
    Split -5 kkwp "Nashik"
21. How to use join command? Create 2 tables and write the output after joining.
    Ans. Page 26

22. Write output of following commands;

i) Display all file names which starts with ‘p’ and ends with ‘a’.
Ans. ls p*a

ii) Enlist all the files beginning with ‘j’ and ending with any range of 1 to 5.
Ans. ls j*[1-5]

iii) Show the contents of the files whose file names contains exactly two characters.
cat ??

iv) Create a file ABCD.txt, create a copy with XXX.txt. Rename the original file with AACD.txt. Delete the file XXX.txt.

```
cat > ABCD.txt
cp ABCD.txt  XXX.txt
mv ABCD.txt  AACD.txt
rm XXX.txt
```

v) Display the inodes of any two files at the same time.
Ans. ls -i file1.txt file2.txt

23. List all file processing commands.
Ans. Page 24 25 26 27

24. How many lines will be displayed with head command if number is not specified.
    Ans. 10 Lines
25. Create two files chapter 1 and chapter 2 and perform the following operation
    i) Copy contents of chapter1 and chapter 2 and perform the following operations.
       Ans. cp chapter1.txt chapter2.txt
    ii) Rename the file 'chapter1' to 'Lesson 1'
       mv chapter1.txt lesson1.txt

26. How to shift from Root directory to User(Home) directory?
    Ans. cd /
27. How to see directory? Write command to go in and come out from specific directory
    See directory: ls
    Command to go in:
    Come out from specific directory: cd directory name

28. What is default set of permissions given by the system to the file and directory?
    Ans. Read,write and execute

29. Assign all the permissions to your directory for the users using symbolic and octal method.
    Ans. using symbolic method: chmod u+rwx abc
         Using octal method: chmod 700 abc
30. What is difference between comm and cmp command.
    Page 40

31. Write the command for performing the following tasks sequentially
    a. Display your current directory.
       pwd
    b. Create a directory 'subject' in the current directory.
       mkdir subject
    c. Create a file 'sample' in the directory 'subject'.
       cd subject
       touch subject
    d. Remove the write permission for the owner for 'sample' using symbolic method.
       Chmod o=rx sample
    e. Delete the file 'sample. What is an error message displayed?
       rm sample
       rm : remove write protected regular file 'sample'?

32. What are the permissions assigned to the file after execution of following commands?
    a. chmod 700 abc
       rwx------
    b. chmod u+rwx, go-rwx file1 file 2
       Ans. rwx------
    c. chmod 536 xyz

Ans.

Create new files ‘cup’ and ‘fan’ and perform the commands
chmod ugo=r cup
chmod ugo+r fan

33. Assign read and write permission for the owner, write permission for the group and execute permission for the others using octal method for the mfile.
    Ans. chmod 621 mfile
34. Write commands to assign following permissions to the file OSY using octal method
    a. _ _ _ _ _ _ _ _ _ : chmod 000 OSY
    b. _rw_r_xr__ : chmod 645 OSY
    c. _r_xr_xr_x : chmod 555 OSY
35. Write commands to assign following permissions to the file OSY using symbolie method. a._rwxr_xr_____
    Ans. chmod u=rwx,g=r-x,o=r OSY

36. Try the commands and write output with its meaning
    i. tr “[a-f]” “[0-5]” < employee

Output Meaning: It will replace any occurrence of the characters 'a' through 'f' with the characters '0' through '5' in the file "employee".

    ii. tr –s “ ” < employee
        Output Meaning: It will remove extra spaces and leave only one space between consecutive words in the file "employee".
    iii. tr –d “f” <employee
        Output Meaning: It will remove all occurrences of the character 'f' in the file "employee".
37. Write the significance of following
    i. vi text.txt
    ii. Insert multiple lines
    iii. Delete contents using commands

i. vi text.txt

Significance: This command opens the vi text editor and creates or opens a file named text.txt.
Explanation: When you execute vi text.txt, you enter the vi `editor to view or edit the contents of the specified file (text.txt). If the file doesn't exist, vi will create an empty file with that name.
ii. Insert multiple lines

Significance: This refers to the action of inserting multiple lines of text into the file being edited in vi.
Explanation: Once inside the vi editor, you can switch to insert mode by pressing i and then type or paste the desired text. To insert multiple lines, press Enter at the end of each line. After inserting the text, press Esc to return to command mode.
iii. Delete contents using commands

Significance: In vi, there are commands to delete text or lines from the file being edited.

Explanation:

To delete characters:

In command mode, position the cursor over the character to be deleted and press x.

To delete entire lines:

In command mode, position the cursor on the line to be deleted and type dd.

To delete multiple lines, prefix dd with a number (e.g., 2dd to delete two lines).

To delete from the cursor position to the end of the line, type D.

To delete specific portions of text:

In command mode, position the cursor at the beginning of the text to be deleted, type d, then move the cursor to the end of the text, and press d again.

38. Write a shell script to print table of given number using FOR loop.

Ans.

```
echo "enter a number"
read num

echo "Table of $num"
for((i=1;i<=10;i++))
do
        result=$((num*i))
        echo "$num x $i = $result"
done
```

39. Execute shell script to print following pattern using FOR loop.

```
*****
****
***
**
*
```

Ans.

```
rows=5
for((i=rows;i>=1;i--))
do
for((j=1;j<=i;j++))
do
echo -n "*"
done
echo
done
```

40. Write a shell script to display Fibonacci series for n numbers. Input n from user.

Ans.

```
echo "Enter number"
read num

a=0
b=1
count=0
echo "Fibonacci series :"
while [ $count -lt $num ]
do
echo -n "$a"
temp=$((a + b))
a=$b
b=$temp
((count++))
done
```

echo

41. Write a shell script to display factorial of a given no.

Ans.

```
echo "Enter number"
read num
```

```
factorial=1
original_no=$num

while [ $num -gt 1 ]
do
factorial=$((factorial * num))
((num--))
done

    echo "Factorial of number= $factorial"
```

42. Write a shell script to check whether given no. is ODD or EVEN.

    Ans.

```
echo "Enter number"
read num

if (($num % 2 == 0))
then
echo "$num is an even number"
else
echo "$num is an odd number"
    fi
```

43. Write a shell script to display tables of n using While loop. Input n from user.

```
echo "Enter number"
read num

i=1
echo "Multiplication of $num"
while [ $i -le 10 ]
do
result=$((num * i))
echo "$num x $i= $result"
((i++))
    done
```

44. Write a shell script to accept four digit number and perform addition of all digits.

```
echo "Enter four digit number"
read num
digit1=$((num/1000))
digit2=$(((num/100) % 10))
digit3=$(((num/10) % 10  ))
digit4=$((num % 10))
```

```
sum=$((digit1 + digit2 + digit3 + digit4))
    echo "sum of all digits=$sum"
```

45. Execute the script for the following
    The for loop using day of week list.
    Ans.

```
for i in Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
do
  echo “$i”
done
```

46. Execute the script for the following
    The while loop to print different * patterns

47. Execute the script for the following

The case statement for performing various mathematical operations

```
echo "enter number one"
read num
echo "enter number two"
read numm

echo "choose option"
echo "1.addition"
echo "2.subtraction"
echo "3.multiplication"
echo "4.division"
echo "enter your choice"
read ch
case $ch in
1) result=$((num + numm))
   echo "addition=$result";;
2) result=$((num - numm))
   echo "subtraction=$result";;
3) result=$((num * numm))
   echo "multiplication=$result";;
4) result=$((num / numm))
   echo "division=$result";;
*) echo "invalid choice"
     esac
```

48. Write a shell script to copy source file into the destination file.
    Ans.

```
echo “Enter source filename”
```

```
read srcfile
echo “Enter destination filename”
read destfile
if [ -a $srcfile ]
then
cp $srcfile $destfile
else
echo “$srcfile doesn’t exist”
fi
```

49. Write a shell script to find smallest/greatest no among two/three no.s.

Ans.

Smallest no among two nos.

```
echo “Enter no1 ”
read no1
echo “Enter no2 ”
read no2
if [ $no1 -lt  $no2 ]
then
   smallest=$no1
else
  smallest=$no2
fi
echo “The smallest number:$smallest”
```

Smallest no among three nos.

```
echo "enter number 1"
read num
echo "enter number 2"
read numm
echo "enter number 3"
read nummm
if [ $num -lt $numm -a $num -lt $nummm ];
then
smallest=$num
elif [ $numm -lt $num -a $numm -lt $nummm ];
then
smallest=$numm
else
smallest=$nummm
fi

echo "smallest=$smallest"
```

Greatest number among two nos.

```
echo "enter number 1"
read num
echo "enter number 2"
read numm

if [ $num -gt $numm ];
then
greatest=$num
else
greatest=$numm
fi

    echo "greatest=$greatest"
```

Greatest number among three nos.

```
echo "enter number 1"
read num
echo "enter number 2"
read numm
echo "enter number 3"
read nummm
if [ $num -gt $numm -a $num -gt $nummm ];
then
greatest=$num
elif [ $numm -gt $num -a $numm -gt $nummm ];
then
greatest=$numm
else
greatest=$nummm
fi

    echo "greatest=$greatest"
```

50. Write a shell script which displays list of all directories in your home directories.

    Ans.

```
    For directories in ‘ls’
    Do
    If [ -d $directories ]
    Then
    echo “$directories”
    else
    echo “Directories not found”
    fi
```

done

51. Write a shell script which displays list of all files in your home directory.

Ans.

```
for files in ‘ls’
do
if [ -f $files ]
then
echo  “$files”
else
echo  “files not found”
fi
done
```

52. Write a file handling program . First check whether it is file or directory, then if it is file the program shall ask user for choices of copying, removing and renaming files. Use case statement.

```
echo “Enter file or directory name: ”
read file
if[ -f $file ]
then
echo “$file is a file”
echo “choose an option”
echo “1. Copy file”
echo “2. Rename file”
echo “3. Remove file”
echo “Enter your choice”
read ch
case $ch in
1)
    echo “Enter filename to be copied”
    read cpf
    cp $file $cpf
    echo “file copied”

2)
    echo “Enter name to rename”
    read name
    mv $file $name
    echo “$file renamed as $name”
```

3)

```
rm $file
echo “removed”

esac
else
if [ -d $file ]
then
echo “$file is a directory”
else
echo “$file is neither a file or directory”
fi
fi
```

53. Write a shell script which displays the list of all executable files in the current working directory.

Ans.

```
for file in ‘ls’
do
if[-f $file -a -x $file]
then
  echo “$file”
fi
done
```

54. 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.

Ans.

```
for file in ‘ls’
do
if [ -f $file ]
then
 if [ -r $file -a -w $file -a -x $file ]
then
echo “$file”
fi
fi
done
```

55. Write a shell script which accepts a filename and assigns it all the permissions.

Ans.

```
echo “Enter a filename:”
Read file
chmod u+rwx $file
chmod o+rwx $file
chmod g+rwx $file
echo “All permissions assigned”
```

56. Implement a C program to calculate Average Waiting time using FCFS Scheduling Algorithm.

```
#include <stdio.h>

void calculateWaitingTime(int processes[], int n, int bt[], int wt[]) {
    // Waiting time for the first process is always 0
    wt[0] = 0;

    // Calculate waiting time for each process
    for (int i = 1; i < n; i++) {
        wt[i] = wt[i - 1] + bt[i - 1];
    }
}

void calculateAverageWaitingTime(int processes[], int n, int bt[]) {
    int wt[n]; // Array to store waiting time for each process
    float avgWaitingTime;

    // Calculate waiting time for each process
    calculateWaitingTime(processes, n, bt, wt);

    // Calculate average waiting time
    avgWaitingTime = 0;
    for (int i = 0; i < n; i++) {
        avgWaitingTime += wt[i];
    }
    avgWaitingTime /= n;

    printf("Average Waiting Time: %.2f\n", avgWaitingTime);
}

int main() {
    // Example:
    int processes[] = {1, 2, 3};
    int n = sizeof(processes) / sizeof(processes[0]);
```

```
        // Burst time for each process
        int burstTime[] = {6, 8, 10};

        calculateAverageWaitingTime(processes, n, burstTime);

        return 0;
         }
```

57. Implement FIFO Page Replacement Algorithm.

```
#include <studio.h>
#define max_frames 3
int main()
{
  int i=0; j=0, hit=0;
   int reference_string []= {1,2,3,2,1,5,2,1,6,2,5,6,3,1,3,1,2,4,3}
   int n=size of (reference string)/ size of (reference_string[0])
  int pageframes [Max_frames];
int pagefaults=0;
int frameindex=0;

 for(i=0;i<max_frame;i++)
{
  pageframe[i]=-1;
}

for(i=0;i<n;i++)
{
   int currentpage=reference_string[i];
   boolean pagefound=false;
   for(j=0;j<max_frames; j++)
   {
       If(pageframes[j]==currentpage)
       {
          pagefound=true;
         printf("page %d has page hit)
         hit++;
          break;
       }
   }
 if(!pagefound)
  {
     printf("page %d  caused a page fault currentpage");
     pageframes[frameindex]-currentpage;
      frameindex-(frameinndex+1)%max_frame;
         pagefault++

  }
}

printf("Total page faults :%d \n", pagefault);
printf("Total page hit :%d \n", hit);
```

```
printf(“Total:%d \n”, n);
return 0;

}
```