Flash

Date Page

1) Cal

October 2022

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

2) $date

Monday 31 october 2022 11:32:42 AM IST

3) $echo "hello"

hello

4) $man

what manual page do you want?

For example, try 'man man'

5) $ls

1BM19CS079 a.text Downloads gg.c mewl.sh public

a2.text Desktop first.txt hello.txt mewtext a3.text

Documents forloop1.sh Music pictures

6) $pwd

/home1 bmsce /[illegible] /rekha

7) $uname

Linux

Flash

Date Page

8) $who
bmsce :0 2022-10-31 11:26(:0)

9) mkdir : It is used to create new directories
mkdir 12
12 plish

10) cd- change directory
cd 12
bmse @ bmscee - HP - Pro - 330 - MT :- 112 $

11) cat - Concatenate file and print on the standard output

cat [OPTION]... [FILE]...

12) tty - print the file name of the terminal connected to standard input

13) Uname - Linux

14) echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
/sbin:/bin:/usr/games:/usr/local/games:/
snap/bin

15) echo $SHELL
/bin/bash

Flash

Date Page

16) wc - print newline, word, and byte counts for each file.

17) rmdir - removes empty directories

18) rm - remove files or directories

19) Ps - report a snapshot of the current process $ps

PID	TTY	TIME	CMD
1851	Pts/0	00:00:00	bash
8173	Pts/0	00:00:00	cat
2547	Pts/0	00:00:00	nano
8796	Pts/0	00:00:00	ps

20) echo "29+14" | bc
43

bc -> basic calculator

Flash

Date Page

Lab-2

stty - change and print terminal line settings

stty -a
-- all
print all current settings in human-readable form

tty - print the file name of the terminal connected to Standard input

Script - make typescript of terminal session

ls - list directory contents

ls -l → use a long listing format

passwd → Assigns a login password

① $nano first.sh
echo "Helloworld" -
$chmod a+x first.sh
→ $sh first.sh
helloworld
$./first.sh
hello world

Flash

Date Page

(2) #!/bin/bash
echo "printing with new line"
echo -n "printing without new line"
echo -e "\n Removing \t back slash \t characters\n"

O/P
printing with new line
printing with new line
removing back slash characters

---

(3) #!/bin/bash

((sum = 25+35))
echo $sum

output
60

---

(3) #bin/bash
((area = 5* 5))
echo $area

O/P : 25

Flash

Date Page

(4)

```
#!/bin/bash
echo "Enter your name"
read name
echo "welcome $name"
```

O/P:

Enter your name
Likitha
welcome Likitha

Flash

Date Page

Week-2

① echo "Enter the year"
Read y
if [`expr $y % 4` -eq 0]
then
echo "The year is leap"
else
echo "The year is not leap"
fi

O/P
Enter the year
2020
The year is leap

---

② echo "enter the radius"
read r
area = echo 3.14 * $r * $r | bc
echo "$area"

O/P:
Enter the radius
4
50.24

---

③ echo "enter the length"
read l
echo "enter the breadth"

Flash

Date Page

read b
area = echo 1\ * $l\ * $b\ bc'
echo " $area "

O/P
enter the length
2
enter the breadth
2
4

echo " Enter the height "
read b
echo " Enter the breadth "
read b
area = " echo 0.5 1"$h *$b
echo " $area "

O/P
enter the height
2
enter the breadth
2
2.0

echo " enter the side length
read s
area = echo 1 \ * $s 1 * $s 1 bc
echo " $area "

Flash

Date Page

o/p

Give the side length

4

16

positional parameter

```
echo " positional parameter "
echo " $0 = " $0
echo " $1 = " $1
echo " $2 = " $2
echo " $3 = " $3
```

$ sh positional.sh I am likitha

o/p

positional parameters

$0 = paspa.sh

$1 = I

$2 = am

$3 = likitha.

Flash

Date Page

week-3

1) check conditions for leap year:

```
echo "Enter year"
read leap
if [ `expr $leap % 400` -eq 0 ]
then
echo leap year
elif [ `expr $leap % 100` -eq 0 ]
then
echo not a leap year
elif [ `expr $leap % 4` -eq 0 ]
then
echo $leap is not a leap year
else
echo $leap is not a leapyear
fi
```

Output:

```
Enter year
2020
2020 is a leap year
```

2) check positive or negative or zero

```
echo "Enter a number"
read num
if [ $num -lt 0 ]
then echo "negative"
```

Flash

Date Page

elif [$num -gt 0]
then
echo "Positive"
else
echo "Neither positive nor negative"
fi

Output

Enter a number
-1
negative
Enter a number
2
positive
Enter a number
0
Neither positive nor negative

3] Find the largest of 3 numbers

echo "Enter 3 numbers"
read num1
read num2
read num3
if [$num1 -gt $num2] && [$num1 -gt $num3]
then
echo largest is $num1
elif [$num2 -gt $num1] && [$num2 -gt $num3]
then
echo largest is $num2
else

Flash

Date Page

echo "Largest is $num3"
fi

Output
Enter 3 number
23
34
56
Largest is 56

4) Largest no using command line arguments

if [$# -ne 3]
then
echo "3 number are not entered"
exit 1
fi
n1 = $1
n2 = $2
n3 = $3
if [$n1 -gt $n2] && [$n1 -gt $n3]
then
echo "$n1 is the biggest"
else
echo "$n3 is the biggest"
fi
$ sh greater and sh 10 12 34
34 is the biggest

Flash

Date Page

5) Compare 2 Strings

```
echo "Enter the Str1 & Str2"
read Str1
read Str2
if [$ Str1 = $Str2]
then
echo "True"
else
echo "False"
fi
```

Enter Str1 & Str2
abc
abc
True

6) To check if its a directory file

```
echo "Enter file name"
read file
if [-d $file]
then
echo "It is a directory file"
else
echo "It is not a directory file"
fi
```

Flash

Date _____ Page _____

8)

```
echo "Enter the number"
read n
a=`expr $n % 2`
if [$a -eq 0]
then
echo "even"
else
echo "odd"
```

Output

Enter the number

12

Even.

o/p for 1

Week-4

```
Echo "Enter a number"
read num
fact = 1
while [$num -gt 1]
do
fact = $(( fact * num ))
num = $(( num - 1 ))
done
echo $fact
```

O/P

Enter a number
3
6

```
#!/bin/bash
echo "Enter the basic salary"
read basal
gral = $(echo "$basal + ((20/100) * $basal) + ((10/100) * basal)" | bc -l)
echo "The gross salary is : $gross"
```

O/P:

Enter the basic salary
45000
The gross salary → 58500.000

Flash

Date Page

```
echo " enter fahrenheit temp"
read fahrenheit
celcius =' echo " scale=4; ($fahrenheit-32)/1.1|bc
echo " $fahrenheit celcius"
```

o/p

Enter fahrenheit temp

273

273 degree fahrenheit is equal to 133.888 degree celcius

```
Echo " enter limit"
read n
while [$i -lt $n]
do
sum = $((sum+i))
i= $((i+2))
done
echo "sum = " sum
```

o/p :-

Enter limit

9

sum : 20

Flash

Date Page

```
echo "Input number"
read no
echo "Input power"
read power
count=0
ans=1
while [ $power -ne $ counter]
do
ans=`expr $ans`"
counter=`expr $counter + 1`
done
echo "$no power of $power is $ans"
```

OP:

Input number

2

Input power

5

0 power 5 is 32

```
echo "enter limit"
read n
i=0
while[$i -le $n]
 do
Sum=$((Sum+i))
i=$((i+1))
done
echo $Sum="$Sum
```

Flash

Date Page

OP:

```
Enter limit
4
Sum: 10

echo "enter a"
read a
echo "Enter b"
read b
Echo "Enter the operation to be performed"
read op
case $op in
+) c=`expr $a + $b` ;;
-) c=`expr $a - $b` ;;
//) c=`expr $a / $b` ;;
*) c=`expr $a * $b` ;;
*) echo "no valid operation specified" ;;
esac
echo Result after performing operation on a and
  b is
echo $c
```

OP

```
Enter a
3
Enter b
4
Enter operation
*
```

Result after performing the operation is 12

Flash

Date Page

week-5

WAP to generate all comb of 1, 2, 3

```
clear
for i in 1 2 3
do
for j in 1 2 3
do
for k in 1 2 3
do
echo $i $j $k
done
done
done
```

Output:-

111	231
112	232
113	233
121	311
122	313
123	321
131	322
132	323
133	331
211	332
212	333
213	
221	
222	
223	

Flash

Date Page

2] Fibonacci series

```
echo " Enter the value of "
read n
a=0
b=1
count=2
echo " Fibonacci series "
echo $a
echo $b
while [ $count -le $n ]
do
fib=`expr $a + $b`
a=$b
b=$fib
echo $fib
count=`expr $count + 1`
done
```

op

Enter the value of n=5

0

1

2

3

0/p ✓ 5/12/22

1 pending

Flash

Date Page

3) echo "Enter Marks"
read marks

if test $marks -lt 40
then echo "fail"

elif test $marks -ge 40 -a $marks -lt 50
then echo "third class"

elif test $marks -gr 50 -a marks -lt 60
then echo "Second class"

elif test $marks 60 -a $marks -lt 70
then echo "first class"

else
echo "destinction"
fi

O/P

Enter marks
58
Second class

Flash

Date Page

12/12/2022 Week-6

1] Shell script to count the number of vowels

```
echo "Enter the string"
read str
l = expr length $str
vowel=0
while [$l -gt 0]
do
temp='expr $str | cut -c $l'
case $temp
in
a|A) vowel='expr $vowel+1';;
e|A) vowel='expr $vowel+1';;
i|A) vowel='expr $vowel+1';;
o|A) vowel='expr $vowel+1';;
u|A) vowel='expr $vowel+1';;
esac
l='expr $l-1'
done
echo "The string has $vowel vowels"
```

O/P

```
$ sh vowel.sh
Enter the string
mango
The string has 2 vowels
```

Flash

Date Page

2) Shell script to check the number of lines, words and characters

```
echo "Enter file name"
read file
if [ -f $file ]
then
echo "File exists"
echo "no. of lines"
wc -l $file
echo "no. of characters"
wc -c $file
echo "no. of words"
wc -w $file
else echo "file does not exist"
fi
```

O/P

```
sh checkfile.sh
Enter file name
wwbc
file does not exist
Enter file name
vowel.sh
file exists
no of lines
19 vowel.sh
no of characters
362 vowel.sh
```

Flash

Date Page

no of inputs

73 name.sh

3) Shell script to do below commands

Set +

Set value = 7

value = 7

echo $value

echo $1

echo $2

echo $3

O/P:

7

value = 7

Shift +

Set value = 7

shift 1

echo $1

echo $2

echo $3

O/P

27

O/P seen 12/12/22

Flash

Date Page

trap

$ nc
$ nc
$ trap - echo hello world " SIGINT
$ ^C hello world // now for ctrl+c, hello world is printed
$ trap - SIGINT // remove trap

4) chmod command

$ trap.sh // create file
$ chmod +x trap.sh // make it
$ touch trap.sh // create file
$ chmod +x trap.sh // make it executable

5) chown (changes owner of file)
chown bmsce area.sh

6) chgrp

Flash

Date Page

19/12/2022

Week-7

1] Write a shell script for GCD/LCM

```
echo "Enter two integers"
read m n
echo "Find GCD & LCM"
temp=`expr $m \* $n`
while [$m != $n]
do
if [$m -gt $n]
 then
m=`expr $m - $n`
else
 n=`expr $n - $m`
fi
done
echo GCD = $n
LCM = `expr $temp / $n`
echo LCM $LCM
```

2] Print the pattern

```
num=1
num=5
for ((i=1; i<=rows; i++))
do
For ((j=1; j<=i; j++))
do
```

Flash

Date Page

```
echo -n " $number"
number = $((num+1))
done
numb = 1
echo
done
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Q) write shell script for creating hard & soft links file.

Creating hardlink

```
Sh hardlink.sh
ln -v hardlink.sh hellolink
'hellolink' => 'hardlink.sh'
ls -i hardlink.sh hellolink
3409795 --- rw-rw-r-- 4 bmsce bmsce hellolink
3409795 --- rw-rw-r-- 4 bmsce bmsce hardlink.sh
Sh hellolink
This is the hardlink program
Deleting
rm hellolink
```

Flash

Date Page

Creating Softlink --

ln -s hardlink.sh softlink
'softlink 1' —> 'hardlink.sh'
ln -li hardlink.sh softlink
34090795 -rw-rw-r--- 4 bmsce hardlink.sh
34080836 lrwxrwxrwx 1 bmsce softlink -> hardlink.sh
sh softlink 1
Hardlink file program

Delete,
rm softlink 1

8/8/8
14/12/22

4) Shell script to locate and print from your home directory tree

#!/bin/bash
echo "enter directory on which you want to find"
read dir
echo "enter type of file you want to find"
echo "1. html files"
echo "2. 666 permission files"
read type
case $type in
(i) find $dir -name "*.html"
;;
(ii) find $dir -perm 666
;;
*) echo "incorrect type"

O/P

Flash

Date Page

esac

enter type of file you want to search

1. html files

2. bbb permission files

1

./a.html

./b.html

2.

./con.sh

./per.sh

Flash

Date Page

Week-8

3b(i)

```
echo " Menu driven ss "
echo " Enter the choice "
read choice
case $choice in
1) echo " list of files 'ls' " ;;
2) echo " machine char 'stty' ;;
3) echo " todays date 'date' " ;;
4) echo " name of terminal $SHELL " ;;
*) echo 'exit' ;;
esac
```

O/P

```
Menu driven shell script
Enter the choice
1
list of files 3b2.sh
inf
aetav.sh
;
Enter the choice
3
todays date Monday 26 December 2022 11:52:21
Enter the choice
4
Name of terminal /bin/bash
Enter the choice
*
$
```

Flash

Date Page

3(ii)

```
echo " To generate electricity bill"
echo " Enter cname"
read cname
echo " Enter ccd"
read ccd
Echo "Enter the units consumed"
read units ; charge =0
unit = $[ [$unit -b -le 199]
then
charge = $ (expr $unit \* 1.55)
else if [$ units -eq 200] && [$units -ge 400]
then
charge = ($units * 1.50)
else if [$units -eq 400] && [$units -le 600]
then
charge = ($units * 1.80)
then
bill $((($bill * 0.13 + bill))
if [$bill -lt 100]
then
bill = 100
fi
```

o/p

```
Enter units = 200
bill = 300
```

Flash

Date Page

4b)
```
echo "enter m and n"
read mn
for a in $(seq $m $n)
do
k=0
for i in $(seq 2 $(expr $a - 1))
do
if [$(expr $a % $i) -eq 0]
then
k=1
break
fi
done
if [$k -eq 0]
then
done
fi echo $a
```

4b)
```
Enter m and n
3 10
3
5
7
```

4a i)
```
echo "Enter the decimal number"
read dn
b=$(echo "obase=2; $dn" | bc)
echo binary $b
```

o/p)
```
Enter the decimal no
12
binary 1100
4
binary 100
```

Flash

Week 9

Date Page

$ Cat > emp.lst

2233 | a.k Shukla | g.m | Sales | 12 | 12 | 52 | 6000

9576 | jai Sharma | director | production | 03 | 12 | 50

5678 | Sumit chakrabarty | d.g.m | marketing | 04 | 19 | 21 | 6000

2365 | barun sengupta | director | personnel | 05 | 11 | 47 | 7800

⋮

2345 | j.b Saxena | g.m | marketing | 03 | 12 | 45 | 8000

0110 | v.k agarwal | g.m | marketing | 12 | 31 | 40 | 9000

1) to display header and footer of a file

$ head emp.lst

2233 | ak Shukla | g.m | Sales | 12 | 12 | 52 | 6000

9876 | jai Sharma | director | production | 03 | 12 | 50 | 7000

⋮

$ head -n 3 emp.lst

2233 | a.k Shukla | g.m | sales | 12 | 12 | 52 | 6000

9876 | jai sharma | director | production | 03 | 12 | 50 | 7000

5678 | Sumit chakrabarty | dgm | marketing | 04 | 19 | 23 | 6000

$ tail -n 3 emp.lst

2345 | j.b Saxena | g.m | marketing | 03 | 12 | 45 | 8000

0110 | v.k agarwal | g.m | marketing | 12 | 31 | 40 | 9000

$ tail emp.lst

6213 | kamuna ganguly | g.m | accounts | 06 | 05 | 62 | 6800

⋮

0110 | v.k agarwal | g.m | marketing | 12 | 31 | 40 | 9000

Flash

Date Page

2) b display first 5 lines & last 5 lines of file $head -n 5 emp.lst

2237|a.k Shukla|g.m|Sales|12|12|52|6000
9876|Jai Sharma|director|production|03|12|50|7000
5678|Sumit chakraborty|dgm|marketing|04|19|43|6000
2365|barun Sengupta|director|personnel|05|11|47|7800
5423|nk gupta|chairman|admin|08|30|56|5400

$ tail -n 5 emp.lst
3212|shyam saksena|dgm|accounts|12|12|55|6000
!
0110|V.K agarwal|g.m|marketing|12|31|40|90000

3) grep and with any 5 basic regular expressions operations on input file

=) a) pattern matching

$ grep "Sales" emp.lst
2233|a.k Shukla|g.m|Sales|12|12|52|6000
1006|chanchal Singhvi|director|Sales|09|03|38|6700
1265|sn dasgupta|manager|Sales|09|12|63|5600
2476|anil aggarwal|manager|Sales|05|01|54|500000

$ grep -i "Agarwal" emp.lst
3564|sudhir Agarwal|executive|personnel|07|06|47|8000

$ grep -i 'aggarwal' emp.lst
2476|anil aggarwal|manager|Sales|05|01|54|500000

E-Engineers days

Flash

Date Page

=> grep -v '' emp.lst - 'v' displays except ''
$ grep -v 'Sales' emp.lst
9876 | jai Sharma | director | production | 03|12|50 | 7000

grep -v 'production' emp.lst
2233 | a.k Shukla | g.m | Sales | 12|12|52 | 6000
grep -v 'Sharma' emp.lst
2233 | a.k Shukla | g.m Sales | 12|12|52 | 6000

=> grep -n 'pattern' emp.lst
-n displays the line no's containing the pattern, along with the lines.

$ grep -n 'marketing' emp.lst
3: 5678 | Sumit chakraborty | d.g.m | 01|19|43| 6000
11: 652 | alok | dir | marketing | 09|20|15 | 8200
14: 2345 | j.b saxena | g.m | marketing | 03|12|45| 8000
15: 0110 | v.k agarwal | g.m | marketing | 15|31|40| 9000

$ grep -n 'Sales' emp.lst
$ grep -n 'Sumit' emp.lst
3: 5678 | Sumit chakraborty | d.g.m | marketing | 04
19|43| 6000

=> grep -c 'pattern' emp.lst
-c counts the pattern with only that line
$ grep -c 'Sumit' emp.lst
1
$ grep -c 'marketing' emp.lst
4

Flash

Date Page

4

=) grep -l ' ' *.lst :- displays the patterns which matches with in the multiple files

cat Y dev.lst
2233| a.k Shukla | g.m | sales | 12 | 12 | 52 | 6000

$ grep -l 'sales' *.lst
dev.lst
emp.lst

=) grep -e :- matches with multiple patterns

$ grep -e 'agarwal' -e 'aggarwal'
-e 'Agarwal' emp.lst

2476 | amit aggarwal | manager | —
3564 | Sudhir Agarwal | executive | —

4) demo of cut and columns and field wise both

$ cut -c 6-22, 24-32 emp.lst
a.k Shukla | g.m | sales | 12 | 1
jai Sharma | director | product

$ cut -d'|' -f 2,3 emp.lst
a.k Shukla | g.m
jai Sharma | director

Flash

Date Page

$ cut -d "|" -f 1,4 emp.lst > empstore.lst

$ cat empstore.lst

9233|Sahu

:

0110|marketing

0/p
6/1/23

Flash

Date Page

Week 10

1) File 1:

Hi

Hello

How are you

File 2:

Hello

Hi

I am good

$paste file1 file2

Hi Hello

Hello Hi

How are you? I'm good.

2) File 3:

ghi 20000

mno 30000

abc 98456

jkl 25678

def 78990

$sort file3

abc 98456

def 78990

ghi 90000

jkl 25678

mno 30000

Flash

Date Page

$ sort -r file3

mno 30000
jkl 25678
ghi 20000
def 78990
abc 98456

$ sort -k 2n file3
ghi 20000
jkl 25678
mno 30000
def 78990
abc 98456

3) i) file1:
hi
hi
hi
hello
hello
how are you?
$ sort file1 | uniq -c
2 Hello
3 Hi
1 how are you?

(ii) echo "ans is a demonstration of tr command" | tr [:lower:] [:upper:]
THIS IS A DEMONSTRATION OF TR COMMAND

Flash

Date Page

4)
```
#include <stdio.h>
int main(int argc, char arg[])
{
int c
char ptr;
extern char **environ;
for (ptr = environ; ptr; = 0; ptr) {
  printf("%s\n", ptr);
}
return 0;
}
```

o/p seen
9/1/23

o/p
$ gcc a.c
$ ./a.out

Flash

Date Page

Week-7a

1) int main (int argc, char *argv[])
{
if (argc < 3 || argc > 4 || (argc == 4 && strcmp(argv[1], "-s")))
{
printf ("Usage: ./a.out [-s] <org_file> <new_link>\n");
return 1;
}
if (argc >= 4)
{
if (CreateSymLink(argv[2], argv[3]) == -1)
printf ("Cannot create Symbolic link\n");
else
printf ("Symbolic link created\n");
}
else
{
if (CreateHardLink(argv[1], argv[2]) == -1)
printf ("Cannot create Hard link\n");
else
printf ("Hard link created\n");
}
return 0;
}

O/P

i/p to given
$ ./a.out 1 2
3 4
o/p for hard
link2 for existing

Flash

Date________Page________

$./a.out $.c

1

i/p for

Softmit for execution

$./a.out -S 14,22.

2) #define

_POSIX_SOURCE

#define

_POSIX_C_SOURCE 199309L

#include <stdio.h>

#include <unistd.h>

int main()

{

#ifdef

_POSIX_JOB_CONTROL

printf ("System supports job control \n")

#endif

#ifdef

_POSIX_SAVED_IDS

printf ("System supports saved set-UID and saved set-GID\n");

#else

printf ("System does not support saved set-UID and saved set-GID \n");

#endif

#ifdef

_POSIX_NO_TRUNC

printf (" Pathname trunc option is:%d\n", _POSIX_NO_TRUNC)

Flash

Date Page

```
#else
printf("System does not support wide pathname truncoption\n");
#endif
#ifdef
-POSIX_VDISABLE
printf("Disable character for terminal files is:%d\n",_POSIX_VDISABLE);
#else
printf("System does not support _POSIX_VDISABLE\n");
#endif
return 0;
}
```

O/P

System supports job control
System supports saved set-UID and saved GID
chown-restricted
option is 1
Pathname :
trunc option is 1
Disable
character for terminal files is 0.

O/P for 6.11 in 3:

Flash

Date _____ Page _____

3)

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <string.h>
#include <errno.h>
#include <stdio.h>
int main(int argc, char *argv[])
{
int fd;
char buf[256];
if(argc != 2 && argc != 3)
{
printf("USAGE %s <file> [<arg>]\n", argv[0]);
return 0;
}
mkfifo(argv[1], S_IFIFO
| S_IRWXU | S_IRWXG | S_IRWXO);
if(argc == 2)
{
fd = open(argv[1], O_RDONLY | O_NONBLOCK);
while(read(fd, buf, sizeof(buf)) > 0)
printf("%s", buf);
}
else
{
fd = open(argv[1], O_WRONLY);
write(fd, argv[2], strlen(argv[2]));
}
close(fd);
}
```

o/p:- $ gcc ex2.c
$ ./a.out

System supports job control
System supports saved set UID and
saved set GID
down - restricted option, ESC
pathname basic option esc
Disable character per terminal

IPC Pending