

1) mkdir DOS_2241019588

cd DOS_2241019588

mkdir Dosass1

cd Dosass1

mkdir dire1

cd dire1

2) cd ..

mkdir dir2

cd dir2

3) rm -dir /home/Student/DOS_2241019588/Dosass1/dire2

4) cat > file1

I/P Name: AnsUMAN Swain

Regd no: 2241019588

Branch: CSE

Sem: 5

Sec : 36

Ctrl+d

cat file1

O/P Name: AnsUMAN Swain

Regd no: 2241019588

Branch: CSE

Sem: 5

Sec : 36

5) cat > file2

I/P: Sem1 : 9.5

Sem2 : 9.5

Sem3 : 9.8

Sem4 : 10

Ctrl+D

cat file2

O/P: Sem1 : 9.5

Sem2 : 9.5

Sem3 : 9.8

Sem4 : 10

- 6) cat file1 file2 > file3
- 7) mv file2 markinfo
- 8) cp file1 reginfo
- 9) ls -i file1 markinfo reginfo
O/P: 27135644 file1 27135643 markinfo 27135647 reginfo
- 10) rm file1
- 11) wc markinfo
O/P: 4 12 37
- 12) cat > personalinfo
Ansuman Swain
2241019588
Palei, Chandol, Kendrapara, Odisha, 754208
- 13) tac markinfo
- 14) cmp reginfo personalinfo
O/P: reginfo personalinfo differ : byte 5, line 1
diff reginfo personalinfo
3,5C3
- < Branch: CSE
- < Sem : 5
- < Sec : 36
- > Address - Palei, Chandol, Kendrapara, Odisha, 754208
- 15) ls | wc -l
- 16) echo * > filelist ~~or~~ ls * > filelist ~~or~~
ls | grep -v "filelist" > filelist
- 17) chmod a+w reginfo
- 18) chmod g-w reginfo
- 19) chmod a=rwx reginfo ~~or~~ chmod 777 reginfo
- 20) date ; pwd
O/P: Wednesday 18 September 2024 02:29:32 PM IST
home /student /DOS_2241019588 /Dosses /dir1

→ date ; pwd | wc -l

O/P:- Wednesday 18 September 2024 02:30:43 PM IST

1

→ (date;pwd) | wc -l

O/P:- 2

Q1) → echo *

O/P:- file3 filelist markinfo personalinfo reginfo

→ echo ***

O/P:- file3 filelist markinfo personalinfo reginfo

→ echo '***'

O/P:- ***

→ echo ***

O/P:- ***

→ echo /* *//*

O/P:- ***

→ echo /*/*

O/P:- /*/*

→ echo Don't do this

O/P:- >

→ echo "Don't do this."

O/P:- Don't do this

→ echo Hello #World

O/P:- Hello

→ echo "Hello#World"

O/P:- Hello#World

→ echo 'Hello#World'

O/P:- Hello#World

→ echo date

O/P:- date

→ echo 'date'

O/P:- date

→ echo "date"

O/P:- date

→ echo 'date'

O/P:- Wednesday 18 September 2024 02:51:12 PM IST

(Date)
21.9.24

1) `cat > a.txt`

5
6
7

[ctrl+d]

`cat > b.txt`

6
7
8

[ctrl+d]

`cat > c.txt`

1
2
3

[ctrl+d]

~~`cat > prog
sort a.txt b.txt c.txt > result; cat result;`~~

[ctrl+d]

`PATH = $PATH : $PWD`

`chmod 764 prog`

`prog`

O/P:

1
2
5
6
6
6
7
7
8

2) ~~`cat > systeminfo`~~
~~`echo "LOGIN NAME = $LOGINAME";`~~
~~`echo "Unix System Name = `uname`";`~~
~~`echo "Shell Name = $SHELL";`~~
~~`echo "Path = $PWD";`~~
~~`echo -e "Total File is :\n `ls`";`~~

[ctrl+d]

`chmod 764 systeminfo`

`systeminfo`

O/P:- LOGIN NAME = Linux Student

Unit System name = Linux

Shell Name = /bin/bash

Path = /home/student /DOS_2241019588 /DOSass2

Total File is :

a.txt

b.txt

c.txt

Prog

result

systeminfo

3) cat > dtcal
echo -e "Date\lt : `date`";
echo "Calender: `cal mar 2022`";
[Ctrl+d]
chmod 764 dtcal
dtcal

O/P:- Wednesday 25 September 2024 02:27:05 PM IST

Calender: March 2022

SU	MO	TU	We	Th	Fr	Sa
			1	2	3	4
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		

4) cat > nvwc
echo -e "Filename :\t dtcal";
echo -e "Line\lt count:\t`wc -L < dtcal`";
echo -e "Word\lt count:\t`wc -w < dtcal`";
echo -e "Charcount\lt:\t`wc -c < dtcal`";
[Ctrl+d]
chmod 764 nvwc
nvwc

O/P:-
Filename : dtcal
Line Count : 2
Word Count : 10
Charcount : 64

5) cat > nvwc2
echo -e "filename\t lineCount \t wordCount \t charCount";
echo -e "\\$1\t\t -wc-l<\\$1\t\t -wc-w<\\$1\t\t -wc-c<\\$1";
echo -e "[ctrl+d]"
chmod 764 nvwc2
nvwc2 darg
filename lineCount wordCount charCount
darg 4 19 119

6) cat > darg
echo "First argument = \$1";
echo "Second argument = \$2";
echo "Total no of arguments = \$#";
echo "All arguments are = \$*";
[ctrl+d]
chrom 764 darg
darg 10 15 12 14

OP2 First argument = 10
Second argument = 15
Total no of arguments = 4
All arguments are = 10 15 12 14

⇒ cat > n disp

$$n = \mathbb{S}^1;$$

$$m = \$2;$$

filename = \$3;

```
m=$2; filename=$3; echo -e "Total no of lines : `wc -l < $filename`"; echo -e "First $n lines of file is : `n `head -n $n $filename`"; echo -e "Last $m lines of file is : `n `tail -n $m $filename`";
```

Chord 764 mid

chmoa
ndisp 2 3 systeminfo
systeminfo

System info

O/P: Total no of lines : 5
First 2 lines of file is :
echo "LOGIN NAME = \$LOGNAME";
echo "Unix System Name = \$SHELL";

Last 3 lines of file is :
echo "Shell Name = \$SHELL";
echo "Path = \$PWD";
echo -e "Total File is : \n ~ls ~"

Chapt. 10. 24

Laboratory Assignment #No. 3

On

Design Principles of Operating System

CSE 3249)

Submitted by

Name : Ansuman Swain

Reg. No. : 2241019588

Semester : 5

Branch : CSE

Section : 36

Session : 2024-2025

Admission Batch : 2022



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY (ITER)
SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY
BHUBANESWAR, ODISHA – 751030

1) gedit iaop

```
echo "Enter value of a";
read a;
echo "Enter value of b";
read b;
echo "$a + $b = `expr $a + $b`";
echo "$a - $b = `expr $a - $b`";
echo "$a * $b = `expr $a * $b`";
echo "$a / $b = `expr $a / $b`";
echo "$a % $b = `expr $a % $b`";
```

[save & exit]

```
chmod a+x iaop
PATH=$PATH:$PWD
iaop
```

O/P: Enter value of a

10

Enter value of b

20

$10 + 20 = 30$

$10 - 20 = -10$

$10 * 20 = 200$

$10 / 20 = 0$

$10 \% 20 = 10$

2) gedit faop

```
echo "Enter value of a";
read a;
```

```
echo "Enter value of b";
read b;
```

```
echo "$a + $b = `echo $a + $b | bc`";
```

```
echo "$a - $b = `echo $a - $b | bc`";
```

```
echo "$a * $b = `echo $a * $b | bc`";
```

```
echo "$a / $b = `echo "scale=3; $a / $b" | bc -l`";
```

```
echo "$a % $b = `echo $a % $b | bc`";
```

```
echo "$a ^ $b = `echo $a ^ $b | bc`";
```

[save & exit]

```
chmod a+x faop
faop
```

O/P: Enter value of a

3.2

Enter value of b

5

$$3.2 + 5 = 8.2$$

$$3.2 - 5 = -1.8$$

$$3.2 * 5 = 16.0$$

~~$$3.2 / 5 = .640$$~~

~~$$3.2 \% 5 = 3.2$$~~

~~$$3.2 ^ 5 = 335.5$$~~

3) gedit q3

echo "Enter the base salary:";

read base;

da=' echo \$base * 0.4 | bc' ;

hra=' echo \$base * 0.2 | bc' ;

gross=' echo \$base + \$da + \$hra | bc' ;

echo "Base is : \$base" ;

echo "DA is : \$da" ;

echo "HRA is : \$hra" ;

echo "Gross is : \$gross" ;

[Save & exit]

chmod a+r q3

q3

O/P: Enter the base salary

1000

Base is : 1000

DA is : 400

HRA is : 200

Gross is : 1600

4) gedit q4

echo "Enter the five digit numbers :" ;

read n;

a='expr \$n \% 10' ;

n='expr \$n / 10' ;

b='expr \$n \% 10' ;

n='expr \$n / 10' ;

c='expr \$n \% 10' ;

```
n = `expr $n / 10 `;
d = `expr $n % 10 `;
n = `expr $n / 10 `;
e = `expr $n % 10 `;
echo "Sum of digit is `expr $a + $b + $c + $d + $e `";
[Save & exit]
chmod a+x 94
```

94

O/P: Enter the five digit number

12345

Sum of digit is 15

5) gedit 95
echo "Enter the cost price :";
read CP;
echo "Enter the sell price :";
read SP;
if [\$SP -ge \$CP]
then
 if [\$SP -eq \$CP]
 then
 echo "No profit no loss";
 else
 echo "profit made `echo \$SP - \$CP | bc`";
 fi
 else
 echo "loss made `echo \$CP - \$SP | bc`";
 fi
[Save & exit]
chmod a+x 95
95

O/P: Enter the cost price :

500

Enter the sell price :

500

No profit no loss

ex:2 Enter the cost price:

500

Enter the sell price :

600

profit made 100

6) gedit leapyear

```
if [ $# -eq 0 ]
then
year=`date +%Y`;
else
year=$1;
fi
a=`expr $year \% 4`;
b=`expr $year \% 100`;
c=`expr $year \% 400`;
if [ $a -eq 0 -a $b -ne 0 -o $c -eq 0 ]
then
echo "$year is a leap year";
else
echo "$year is not a leap year";
fi
```

[save & exit]

```
chmod a+x leapyear
```

O/P : leapyear

```
2024 is a leap year
→ leapyear 2000
2000 is a leap year
→ leapyear 2003
2003 is not a leap year
```

7) gedit allow

```
echo "Enter the internal mark : ";
read m;
echo "Enter the attendance : ";
read P;
if [ $m -ge 20 -a $P -ge 75 ]
then
echo "Allowed";
else
echo "Not allowed";
fi
```

[save & exit]

```
chmod a+x leap year
allow
```

O/P

```
Enter the internal mark :
25
Enter the attendance :
80
Allowed
```

8) gedit small3

```
a=$1;  
b=$2;  
c=$3;  
if [ $a -lt $b -a $b -lt $c ]  
then  
echo "lowest number is : $a ";  
elif [ $b -lt $c -a $b -lt $a ]  
then  
echo "lowest number is : $b ";  
else  
echo "lowest number is : $c ";  
fi  
[save & exit]  
chmod a+r small3
```

O/P
→ small3 10 15 20
lowest number is: 10
→ small3 15 10 20
lowest number is: 10
→ small3 15 20 10
lowest number is: 10

9) gedit check-char
echo "Enter a character :";
read n;
case \$n in

[a-z])
echo "You entered a lower case alphabet";
;;

[A-Z])
echo "You entered an upper case alphabet";
;;

[0-9])
echo "You entered a digit.";

?)
echo "You have entered a special symbol.";

*)
echo "You have entered more than one character.";

;;
 esac
chmod a+r check-char
check-char

D/P: check_char

Enter a character:

C

You entered an upper case alphabet.

10) getit class_time

echo "Enter a day: "

read n;

case \$n in

Sunday)

echo "Holiday";

;;

Monday)

Echo "D&S class 9-10am, room C-024";

;;

Tuesday)

echo "D&S class 12-1pm, room C-024";

;;

Saturday)

Echo "D&S class at 2-4pm, room C-008";

;;

*)

echo "No class on \$n";

;;

esac

chroot a2z class_time

class_time

D/P: Enter a day:

Tuesday

D&S class at 12-1pm, room C-024

11) getit filechk

m=\$1;

n=\$2;

if cmp \$m \$n

then echo "Files \$m and \$n have same content";

rm \$n

echo "So \$n deleted"

else echo "Files \$m and \$n have same content,";

fi

chroot a2z filechk

O/P: filecheck a c
Files a and c have same content.
So c deleted.

2) gedit calculator

```
a=$1;  
b=$2;  
c=$3;  
case $0 in  
+) echo "$a + $b = `expr $a + $b`";  
;;  
-) echo "$a - $b = `expr $a - $b`";  
;;  
*) echo "$a * $b = `expr $a * $b`";  
;;  
/) echo "$a / $b = `expr $a / $b`";  
;;  
%) echo "$a % $b = `expr $a % $b`";  
;;  
^) echo "$a ^ $b = `echo $a ^ $b | bc`";  
;;  
*) echo "Invalid input";
```

esac

[save & exit]
chmod a+x calculator

O/P: calculator 5 + 10
 $5 + 10 = 15$
calculator 5 x 10
 $5 \times 10 = 50$
calculator 15 % 10
 $15 \% 10 = 5$

Ranjit
13.11.24

Laboratory Assignment #No. 4

On

Design Principles of Operating System

CSE 3249)

Submitted by

Name : Ansuman Swain
Reg. No. : 2241019588
Semester : 5th
Branch : CSE
Section : 36
Session : 2024-2025
Admission Batch : 2022



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY (ITER)
SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY

① @

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t c1 = fork();
    if (c1 == 0)
    {
        fprintf(stderr, "Child Process id : %d", getpid());
        while(1);
    }
    else
    {
        fprintf(stderr, "Parent Process id : %d", getpid());
        while(1);
    }
}
```

\$: gcc q1.c

\$: ./a.out

Output → Parent process id : 110 Child process id : 111
(Open another terminal)

\$: ps -al

S	PID	PPID
R	110	10
R	111	110

② \$: kill -9 111

\$: ps -al

S	PID	PPID
R	110	10
Z	111	110

i.e. child will become zombie until parent is terminated.
Parent : Running
Child : EXIT-ZOMBIE

①③ \$: gcc q1.c

\$: ./a.out

Parent Process id : 137 Child process id : 138

S	PID	PPID
R	137	10
R	138	137

\$: kill -9 137

\$: ps -al

S	PID	PPID
R	138	1

e.g. Parent is deleted but child will remain as orphan.
parentid of child process will be changed.

①② #include <stdio.h>

#include <unistd.h>

#include <sys/wait.h>

int main()

pid_t c1 = fork();

if (c1 == 0)

{ printf(stderr, "Child process id : %d ", getpid());

while (1);

}

else

{ printf(stderr, "Parent process id : %d ", getpid());

wait(NULL);

while (1);

3

3

\$: gcc q1.c

\$: ./a.out

Parent process id : 147 child process id : 148

\$: ps -al

S	PID	PPID
S	147	10
R	148	147

i.e., child is Running and Parent is in TASK_INTERRUPTIBLE

② \$: kill -9 148

\$: ps -al

S	PID	PPID
R	147	10

i.e., Parent is Running
child process is Exit dead / terminated

② a) 1 2 3 3

or 2 1 3 3

or 1 3 2 3

or 2 3 1 3

② b) 1 2 3

② c) child : 6 Parent : 6

d) child : 6 Parent : 7

e) child : 6 Parent : 5

f) child : 6 Parent : 6

g) child : 5 Parent : 5

or Parent : 5 child : 5

h) child : 5 Parent : 5

i) 1 2 3

j) 1 2 3

k) Child

$$n = 20$$

Parent

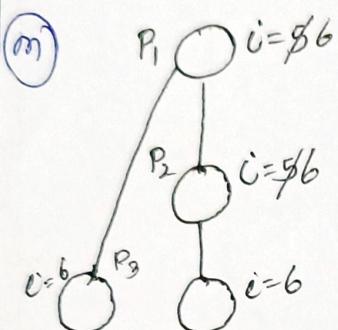
$$n = 10$$

l) Child

$$n = 20$$

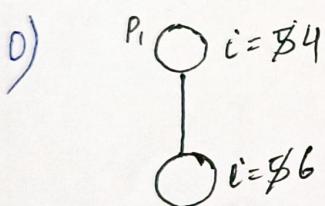
Parent

$$n = 20$$



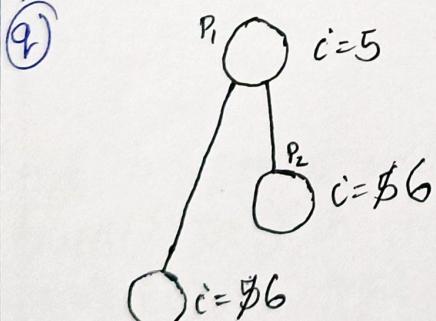
O/P: 6666

n) Child: 5 Parent: 6



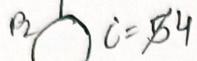
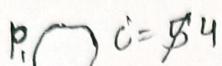
O/P: 64 or 46

p) 5



O/P: 665

(7)



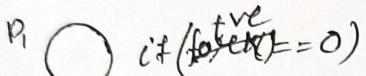
$C = \$4$

$C = \$4$

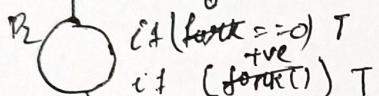
$C = 5$

O/P : 445

(8)

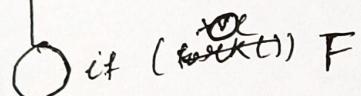


$\text{if } (\text{fork}^{\text{tve}} = 0)$



$\text{if } (\text{fork}^{\circ} = 0) \ T$

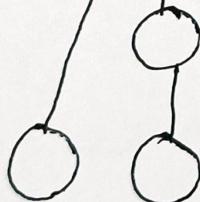
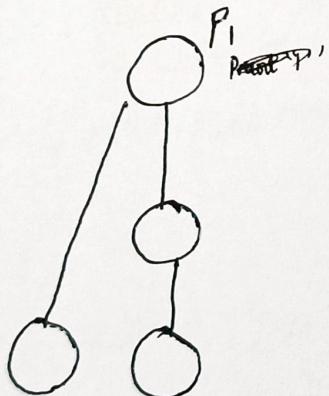
$\text{if } (\text{fork}^{\text{tve}}) \ T$



$\text{if } (\text{fork}^{\times} = 0) \ F$

O/P:- 1

(9)



O/P:-

|

|

|

|

|

|

|

all 4 processes will execute the
`printf("1\n")` in `fun1()` and
`main()`. So total 8 no of 1's. (8 no of 1's)

```

③ #include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
int main() {
    if (vfork() == 0)
    {
        fprintf(stderr, "\n Child process id : %d Parent
process id : %d \n", getPid(), getPPid());
        execl("/usr/bin/CP", "CP", "file1", "file2", NULL);
    }
    sleep(1);
    if (vfork() == 0)
    {
        fprintf(stderr, "\n Child process id : %d Parent
process id : %d \n", getPid(), getPPid());
        execl("/usr/bin/cat", "cat", "file 2", NULL);
    }
    sleep(1);
    if (vfork() == 0)
    {
        fprintf(stderr, "\n Child process id : %d Parent
process id : %d \n", getPid(), getPPid());
        execl("/USR/bin/sort", "sort", "-R", "file2", NULL);
    }
    sleep(1);
    fprintf(stderr, "\n Parent process id : %d ", getPid());
}

```

\$: gcc 93.c

\$: ./a.out

Output:-

Child process id : 165 Parent process id : 164

Child process id : 166 Parent process id : 164

1

2

3

4

5

Child process id : 167 Parent process id : 164

5

4

3

2

1

Parent process id : 164

④ #include < stdio.h >

#include < unistd.h >

int main()

{ int length;

printf ("Enter the length : ");

scanf ("%d", &length);

int fibArray [length];

if (vfork () == 0)

fibArray [0] = 0;

fibArray [1] = 1;

for (int i = 2; i < length; i++)

fibArray [i] = fibArray [i-1] + fibArray [i-2];

exit (0);

}

printf ("Fibonacci number : \n ");

for (int i = 0; i < length; i++)

printf ("%d", fibArray [i]);

```

printf("Prime Fibonacci Series\n");
for (int i=3; i<length; i++) {
    int isPrime = 1;
    for (int j=2; j*j <= fibArray[i]; j++) {
        if (fibArray[i] % j == 0)
            isPrime = 0;
    }
    if (isPrime)
        printf("%d at Index : %d \n", fibArray[i], i);
}
return 0;

```

Output:

Enter the length : 10

Fibonacci Series :

0 1 1 2 3 5 8 13 21 24

Prime Fibonacci numbers :

2 at index : 3

3 at index : 4

5 at index : 5

13 at index : 7