**Name : Keyur Patel**

**Roll no : 16010421073**

**Batch : A3**

**Exp no : 3**

**Process Management**

**Code:-**

#include<stdio.h>

#include<unistd.h>

#include<sys/types.h>

void main()

{

int i=fork();

if(fork())

{

printf("Inchild");

printf("%d %d ", getpid(),getppid());

for(i=0;i<100;i++)

printf("%d",i+2);

printf("\n");

}

else

{

printf(" Inparent");

printf("%d", getpid());

for(i=0;i<100;i++)

printf("%d",i+5);

printf("\n");

}

}

**Command:-**

keyur@Ubuntu:~/Desktop/kp\_73$ gcc menu.c -o menu

keyur@Ubuntu:~/Desktop/kp\_73$ ./menu

**Output:-**

Inchild18637 4750 23456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101

Inparent1863956789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104

keyur@Ubuntu:~/Desktop/kp\_73$ Inchild18638 1253 23456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101

Inparent1864056789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104

**File Management:-**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<fcntl.h>

int main()

{

int fd,option,i;

char a[50];

while(option!=5)

{

printf("\n 1-create\n2-Write\n3-Read\n4-Close File\n5-Exit");

printf("\nEnter the choice:");

scanf("%d",&option);

switch(option)

{

case 1:

fd= creat("e.txt",0777);

printf("\nFile is created successfully.\n");

break;

case 2:

printf("\nEnter the string:\n");

scanf("%s",&a[50]);

fd=open("e.txt",O\_WRONLY,0777);

write(fd,a,10);

close(fd);

break;

case 3:

printf("\nThe entered string is:\n");

i=0;

fd=open("e.txt",O\_RDONLY);

read(fd,a,10);

for(i=0;a[i]!='\0';i++)

{

printf("%c",a[i]);

}

break;

case 4:

printf("\nFile is closed.\n");

close(fd);

break;

case 5:

printf("\nExited.\n");

break;

}

}

return 0;

}

**Command:-**

keyur@Ubuntu:~/Desktop/kp\_73$ gcc hello.c -o hello

keyur@Ubuntu:~/Desktop/kp\_73$ ./hello

**Output:-**

1-create

2-Write

3-Read

4-Close File

5-Exit

Enter the choice:1

File is created successfully.

1-create

2-Write

3-Read

4-Close File

5-Exit

Enter the choice:2

Enter the string:

messi vs ronaldo

1-create

2-Write

3-Read

4-Close File

5-Exit

Enter the choice:3

The entered string is:

messi vs ronaldo

1-create

2-Write

3-Read

4-Close File

5-Exit

Enter the choice:4

File is closed.

1-create

2-Write

3-Read

4-Close File

5-Exit

Enter the choice:5

Exited.

\*\*\* stack smashing detected \*\*\*: terminated

Aborted (core dumped)

**Outcomes:**

**CO1:** Understand basic structure of modern operating system

**Conclusion:**

Understanding the results how system calls are used for file and process  handling.