



## **EXPERIMENT 10**

**NAME: SUDANSHI SEHGAL**

**ROLL NO.: 2021A1R156**

**SUBJECT: OPERATING SYSTEM**

**SEMESTER: 3rd**

**COURSE CODE: COM-312**

**CO-ORDINATER:**

**Ms. PRAGTI JAMWAL**

**Mr. SAURABH SHARMA**

**Experiment 10:-Write a C program to print the address of a variable and enter a long loop (say using while(1)).**

**a) Start three to four processes of the same program and observe the printed address values.**

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

int main()
{
fork();
fork();

int var=1,i=1;

while(1)
{
if(i==5)
{
break;
}
printf("Address of var in loop =%p\n",&var);
i++;
}
return 0;
}
```

[illegible]

b) Show how two processes which are members of the relationship parent child are concurrent from execution point of view, initially the child is copy of the parent, but every process has its own data.

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

int main(void)
{ //declare variable
int var=1;
int* p = (int*)malloc(2);
pid_t PID = fork();
*p = 0;
if (PID >= 0)
{
if (PID ==0)
{
printf("\n\nChild process:\nInitial Value = %d", var);
var=5;
printf("\nNew Value of var = %d", var);
printf("\nAddress of Malloc in child= %p\n", p);
printf("\nAddress of var in child= %p\n", &var);
}
else
{
printf("\n\nParent process:\nInitial Value = %d", var);
var=10;
printf("\nNew Value = %d", var);
printf("\nAddress of Malloc in parent= %p",p);
printf("\nAddress of var in child= %p\n", &var);
}
}
return 0;
}
```

```
sudanshi@sudanshi:~/Desktop$ nano 10-1.c
sudanshi@sudanshi:~/Desktop$ gcc 10-1.c -o 10-1
sudanshi@sudanshi:~/Desktop$ ./10-1

Parent process:
Initial Value = 1
New Value = 10
Address of Malloc in parent= 0x5607d89972a0
Address of var in child= 0x7ffdda7df0b8

Child process:
Initial Value = 1
New Value of var = 5
Address of Malloc in child= 0x5607d89972a0
Address of var in child= 0x7ffdda7df0b8
sudanshi@sudanshi:~/Desktop$
```