

Experiment 10:

Write a program to print the address of a variable and enter a long loop using while(1)

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

C program :

```
#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("Addresses of var in loop=%p\n",&var);
i++;
}
return 0;
}
```

Output:

[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

C program:

```
#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", p);
printf("\nAddress of var in child= %p\n", &var);
}
else
{
printf("\nNew Value = %d", var);
printf("\nAddress of malloc in parent = %p", p);
printf("\nAddress of var in child = %p\n", &var);
}
}
return 0;
}
```

Output:

```
haripriya@haripriya-VirtualBox:~$ nano exp10b.c
haripriya@haripriya-VirtualBox:~$ gcc exp10b.c
haripriya@haripriya-VirtualBox:~$ ./a.out
```

```
parent process :
  initial value = 1
new value = 10
address of malloc in parent = 0x55fb2accc2a0
adress of var in child = 0x7ffda5363ce8
haripriya@haripriya-VirtualBox:~$
```

```
child process :
  initial value = 1
  new value of var = 5
adress of malloc in child = 0x55fb2accc2a0
address of var in child = 0x7ffda5363ce8
```