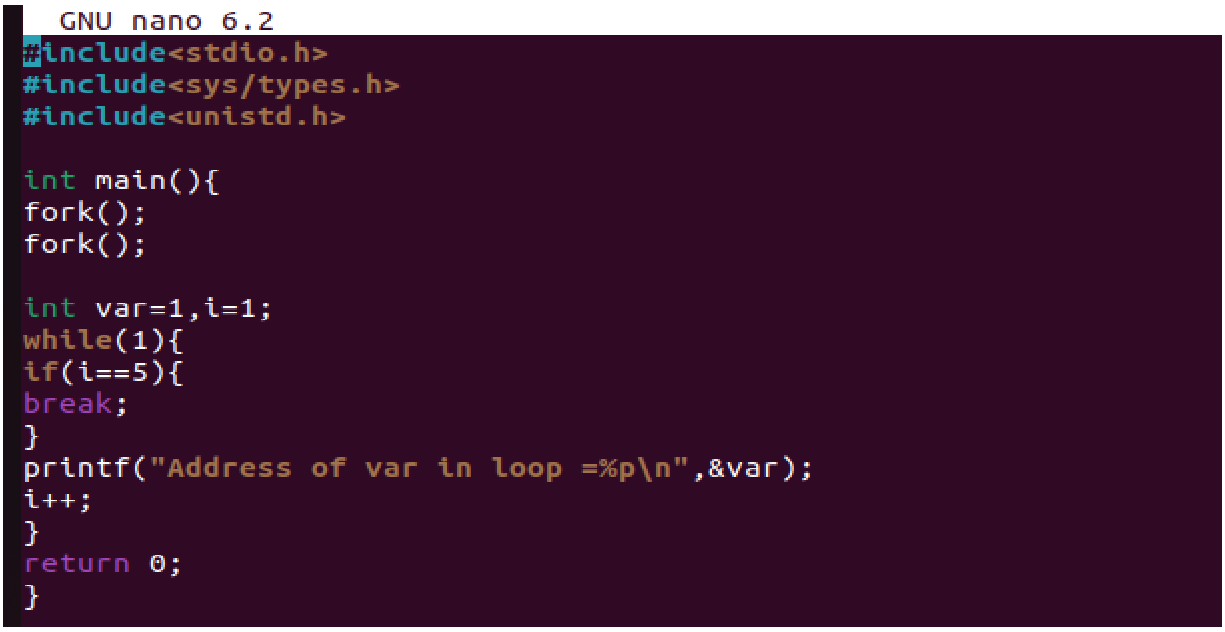
EXPERIMENT 10:

Title: 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
}
```

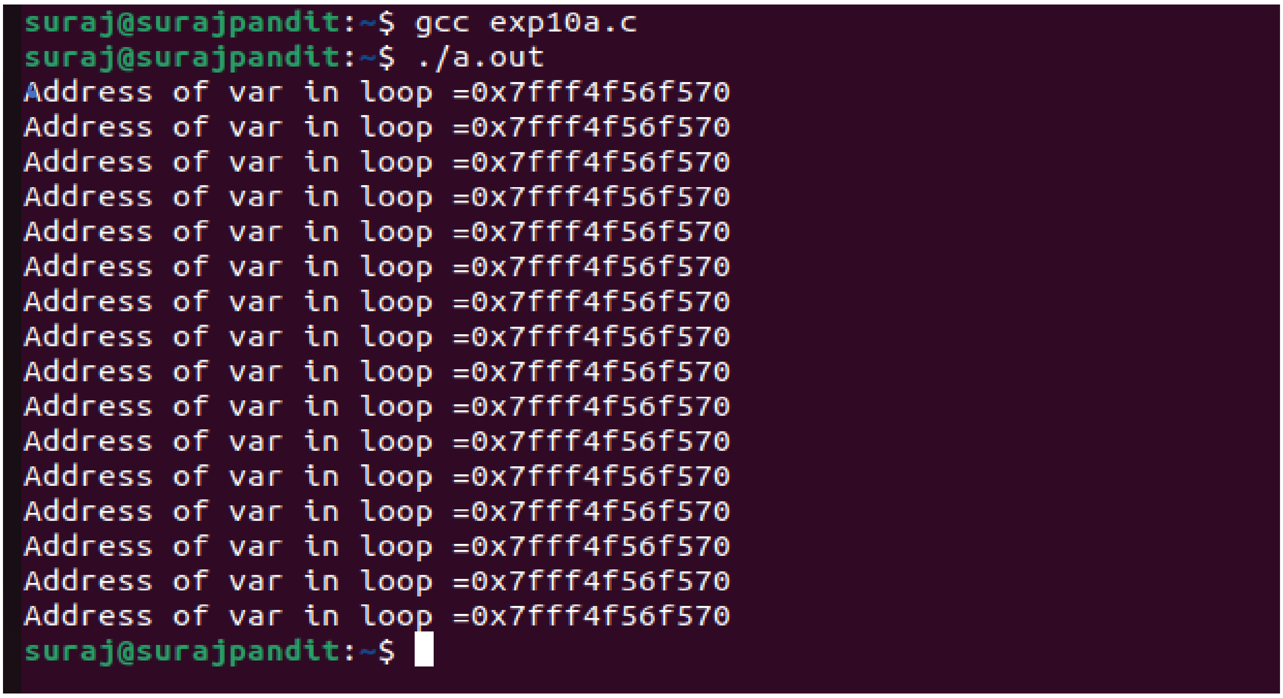


```
  GNU nano 6.2
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>

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;
}
```

OUTPUT:

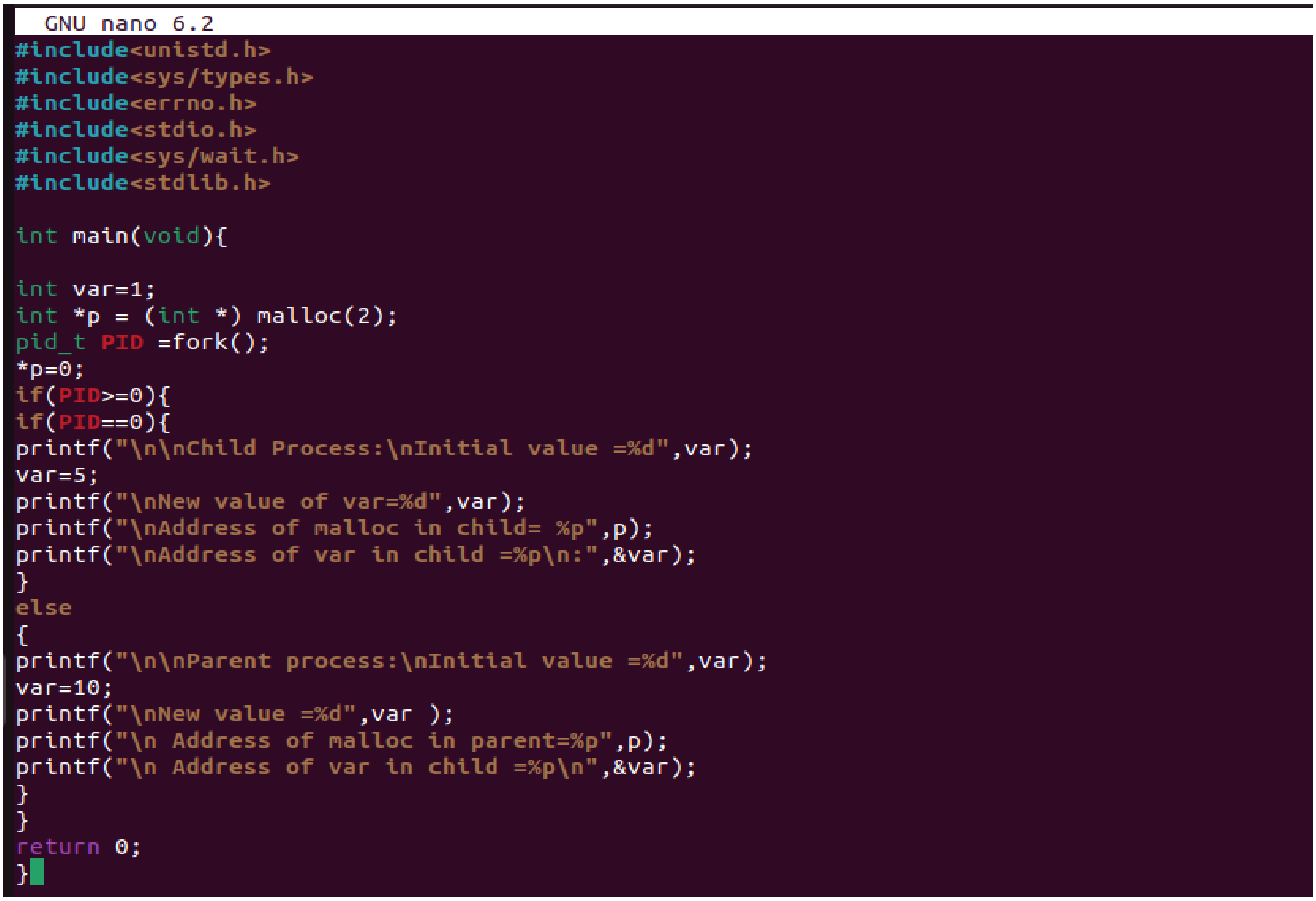


```
suraj@surajpandit:~$ gcc exp10a.c
suraj@surajpandit:~$ ./a.out
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
Address of var in loop =0x7fff4f56f570
suraj@surajpandit:~$ 
```

b) **Show how two processes which are members of the relationship parentchild 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){
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("\n\nParent process:\nInitial value =%d",var);
var=10;
printf("\nNew value =%d",var );
printf("\n Address of malloc in parent=%p",p);
printf("\n Address of var in child =%p\n",&var);
}
}
return 0;
}
```

CODE:

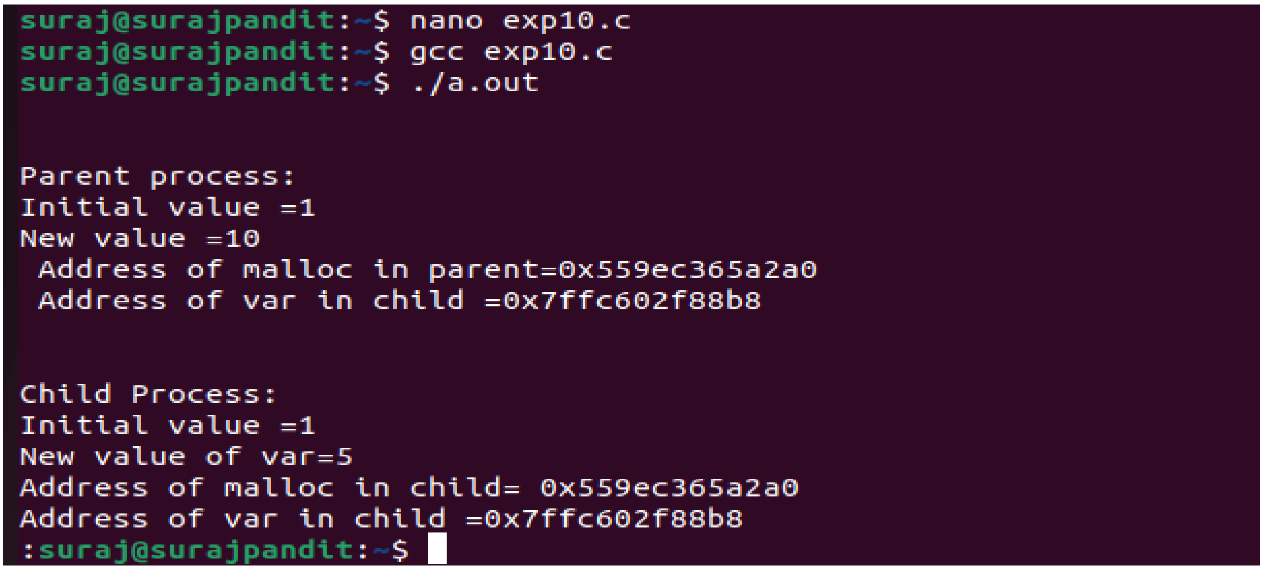


```
  GNU nano 6.2
#include<unistd.h>
#include<sys/types.h>
#include<errno.h>
#include<stdio.h>
#include<sys/wait.h>
#include<stdlib.h>

int main(void){

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("\n\nParent process:\nInitial value =%d",var);
var=10;
printf("\nNew value =%d",var );
printf("\n Address of malloc in parent=%p",p);
printf("\n Address of var in child =%p\n",&var);
}
}
return 0;
}
```

OUTPUT:



```
suraj@surajpandit:~$ nano exp10.c
suraj@surajpandit:~$ gcc exp10.c
suraj@surajpandit:~$ ./a.out


Parent process:
Initial value =1
New value =10
 Address of malloc in parent=0x559ec365a2a0
 Address of var in child =0x7ffc602f88b8


Child Process:
Initial value =1
New value of var=5
Address of malloc in child= 0x559ec365a2a0
Address of var in child =0x7ffc602f88b8
:suraj@surajpandit:~$
```