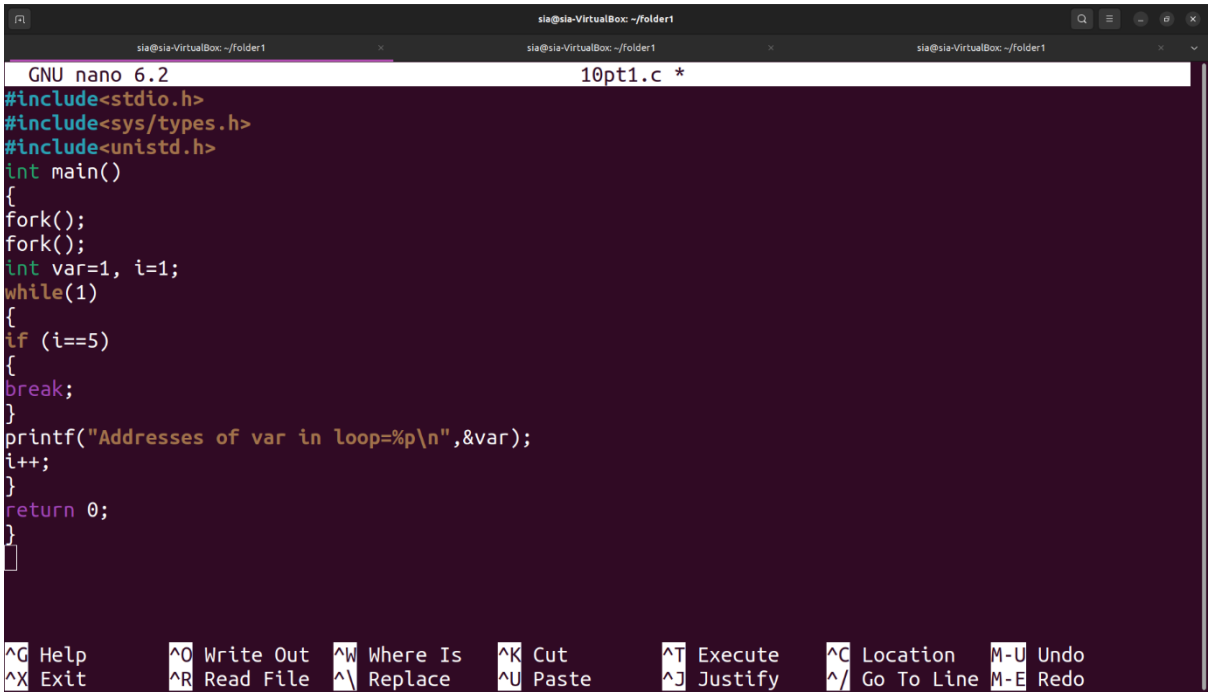
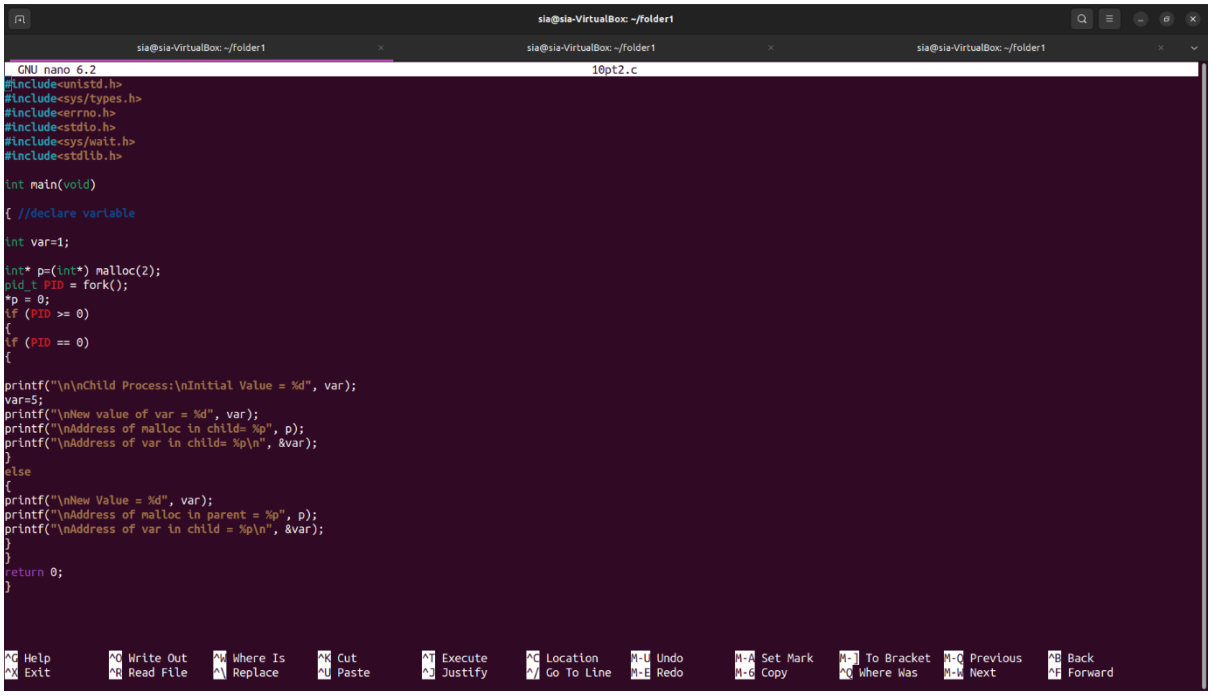
# Experiment – 10

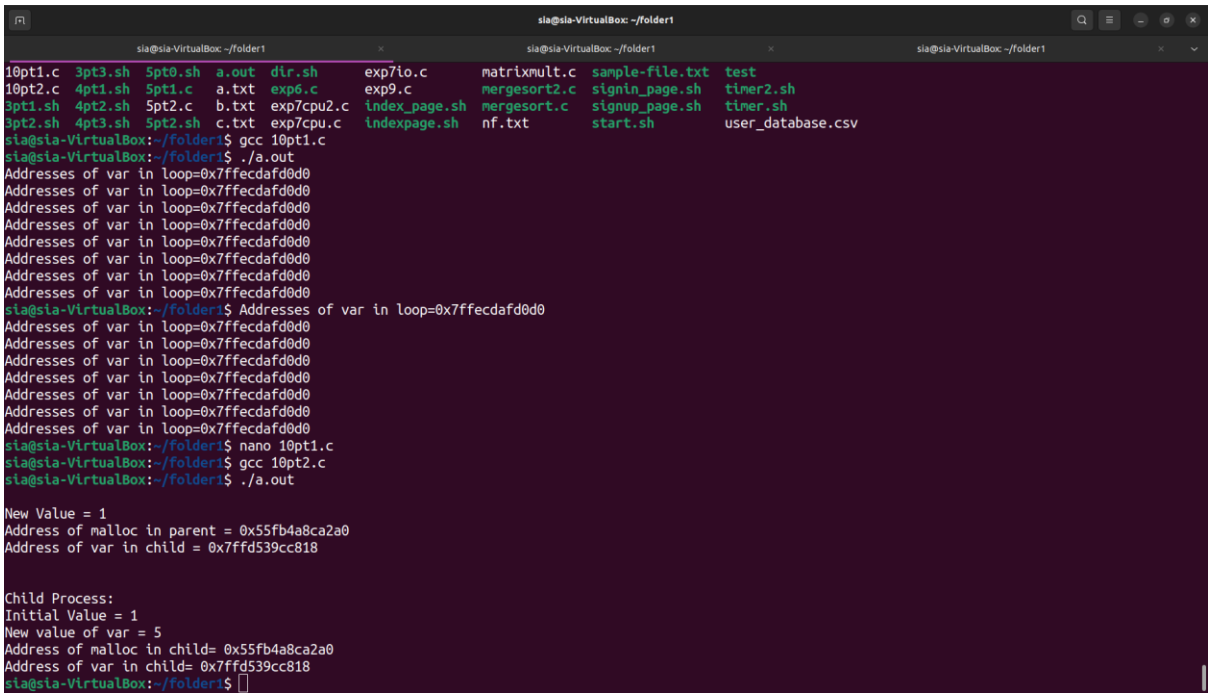
Write a C program to print the address of a variable and enter a long loop (say using while(1)).

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

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.




sia@sia-VirtualBox: ~/folder1
sia@sia-VirtualBox: ~/folder1
sia@sia-VirtualBox: ~/folder1
sia@sia-VirtualBox: ~/folder1
GNU nano 6.2
10pt2.c
#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;
}
^G Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut
^U Paste
^T Execute
^J Justify
^C Location
^/ Go To Line
M-U Undo
M-E Redo
M-A Set Mark
M-6 Copy
M-] To Bracket
^Q Where Was
M-Q Previous
M-W Next
^B Back
^F Forward


sia@sia-VirtualBox: ~/folder1
sia@sia-VirtualBox: ~/folder1
sia@sia-VirtualBox: ~/folder1
sia@sia-VirtualBox: ~/folder1
10pt1.c 3pt3.sh 5pt0.sh a.out dir.sh exp7io.c matrixmult.c sample-file.txt test
10pt2.c 4pt1.sh 5pt1.c a.txt exp6.c exp9.c mergesort2.c signin_page.sh timer2.sh
3pt1.sh 4pt2.sh 5pt2.c b.txt exp7cpu2.c index_page.sh mergesort.c signup_page.sh timer.sh
3pt2.sh 4pt3.sh 5pt2.sh c.txt exp7cpu.c indexpage.sh nf.txt start.sh user_database.csv
sia@sia-VirtualBox:~/folder1$ gcc 10pt1.c
sia@sia-VirtualBox:~/folder1$ ./a.out
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
sia@sia-VirtualBox:~/folder1$ Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
Addresses of var in loop=0x7ffecdafd0d0
sia@sia-VirtualBox:~/folder1$ nano 10pt1.c
sia@sia-VirtualBox:~/folder1$ gcc 10pt2.c
sia@sia-VirtualBox:~/folder1$ ./a.out
New Value = 1
Address of malloc in parent = 0x55fb4a8ca2a0
Address of var in child = 0x7ffd539cc818
Child Process:
Initial Value = 1
New value of var = 5
Address of malloc in child= 0x55fb4a8ca2a0
Address of var in child= 0x7ffd539cc818
sia@sia-VirtualBox:~/folder1$