**Exp 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.**

**C Program:**

#include<stdio.h>

#include<sys/types.h>

#include<unistd.h>

int main()

{

//create process

fork();

fork();

//create variables

int var=1,i=1;

//long loop but broke at i==5

while(1)

{

if(i==5)

{

break;

}

//print Address

printf("Address of var in loop=%p\n",&var);

i++;

}

return 0;

}

**Output:**



**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 cariable

int var=1;

int\* p= (int\*) malloc(2);

pid\_t PID = fork();

\*p = 0;

if (PID >=0)//if fork successfull

{

if(PID ==0)//if it is child process

{

printf("\n\nChild Process :\n Initial Value = %d",var);

var=5;

printf("\nNew Value of var =%p",p);

printf("\nAddress of var in child = %p\n",&var);

}

else //if it is parent process

{

printf("\n\nParent Process:\n Initial 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;

}

**Output:**

