**ASSIGNMENT 7**

**QUESTION a**

Write a Program to demonstrate the use of signal. The process will print a message infinitely until an interrupt signal occurs. It will handle the signal and will print a message along with the signal number that it has got.(Help: signal(SIGINT,SIG\_DFL))

**Source Code**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<fcntl.h>

#include<signal.h>

#include<sys/wait.h>

void custom\_handler(int);

void main()

{

    signal(SIGINT,custom\_handler);

    while(1)

    {

        printf("Hello world. I am Arkapratim Ghosh (13000121058)\n");

        sleep(1);

    }

}

void custom\_handler(int signum)

{

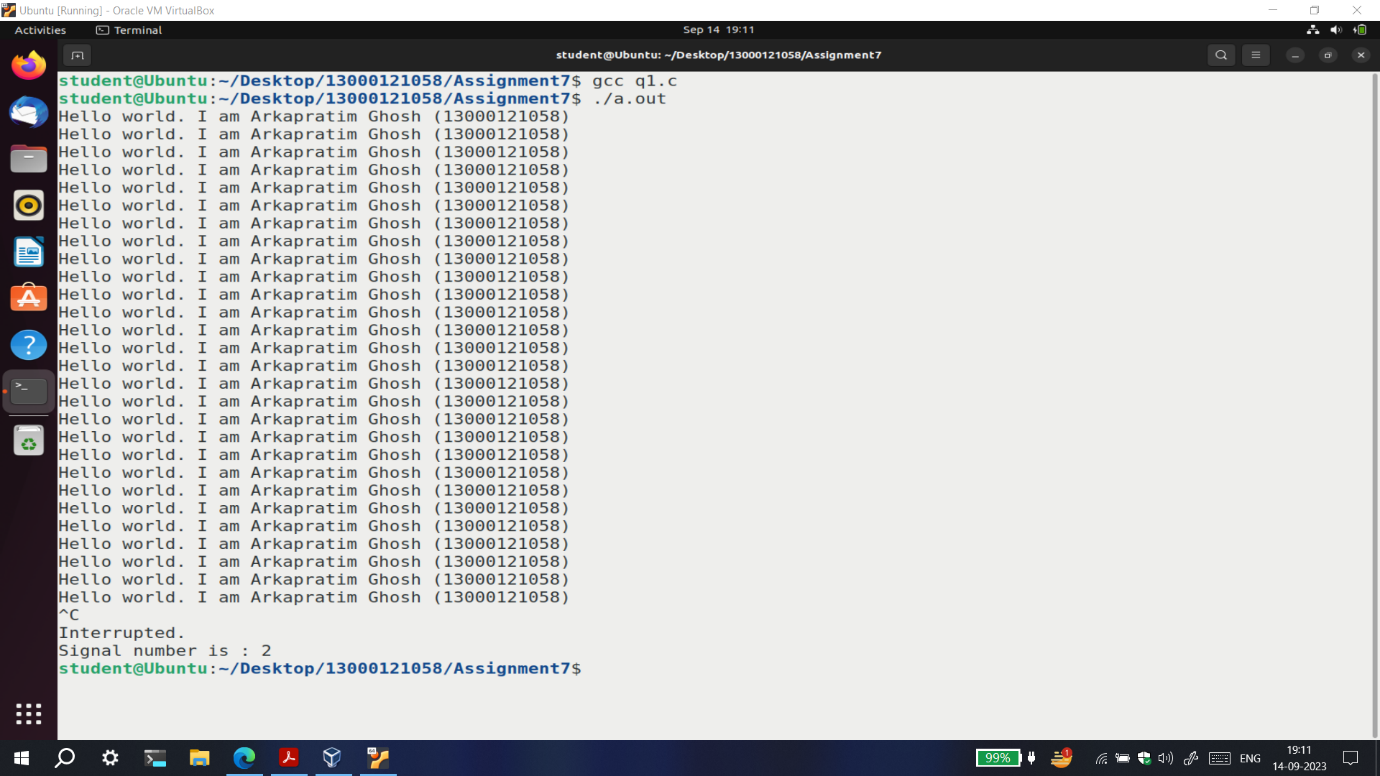
    printf("\nInterrupted.\n");

    printf("Signal number is : %d \n",signum);

    exit(0);

}

**Output**



**QUESTION b**

Write a program to demonstrate the use of signal. Parent process will stop until an alarm received from child process.

**Source Code**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<signal.h>

#include<fcntl.h>

int alarm\_raised=0;

int sig=0;

void raised\_alarm\_handler(int signum)

{

    alarm\_raised=1;

    sig=signum;

}

void main()

{

    signal(SIGALRM,raised\_alarm\_handler);

    int pid=fork();

    if(pid==-1)

    {

        perror("fork");

        exit(1);

    }

    if(pid==0)

    {

        printf("Child process with PID : %d and PPID : %d is running..\n",getpid(),getppid());

        sleep(5);

        kill(getppid(),SIGALRM);

        printf("Child process sent alarm to parent\n");

    }

    else

    {

        printf("Parent process with PID : %d is waiting..\n",getpid());

        while(alarm\_raised==0)

        {

            printf("Hello I am Arka from parent process..\n");

            sleep(1);

        }

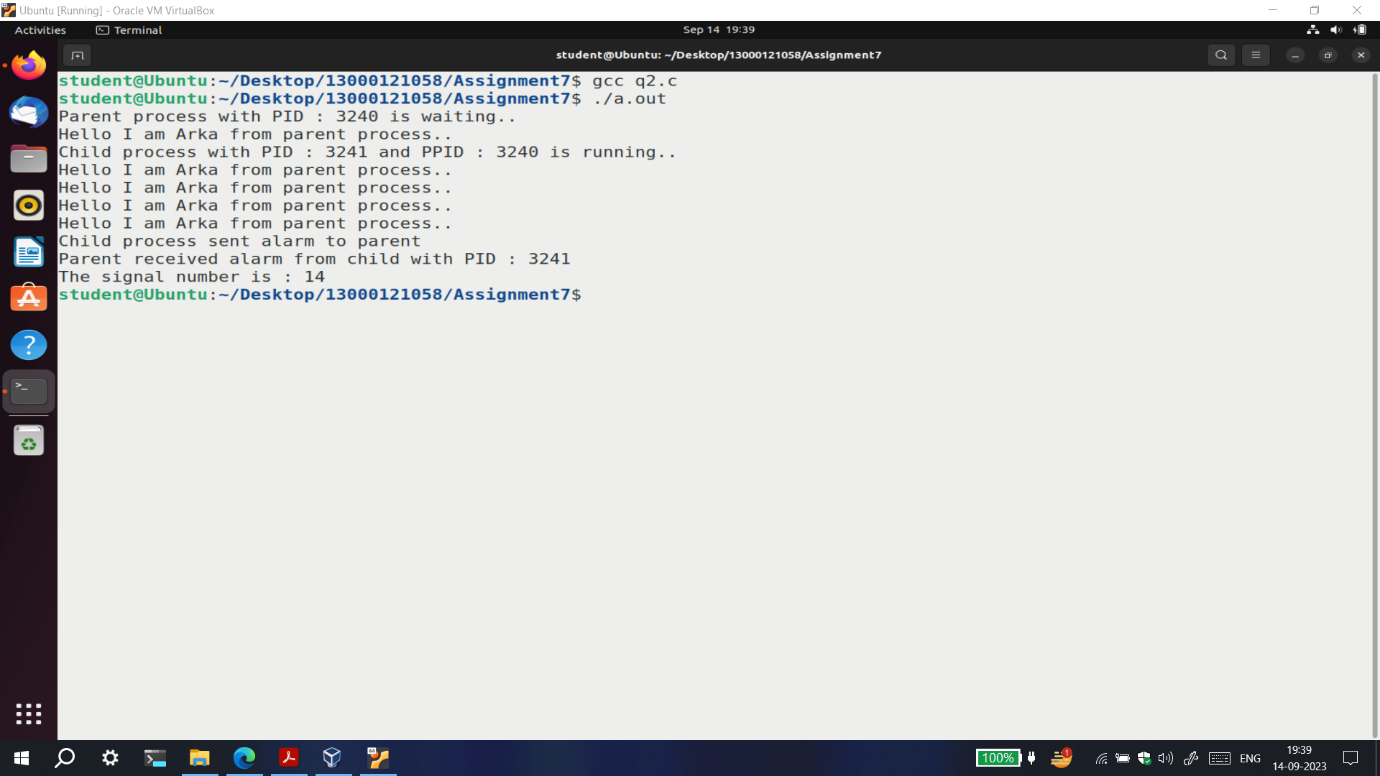
        printf("Parent received alarm from child with PID : %d\n",pid);

        printf("The signal number is : %d\n",sig);

    }

}

**Output**



**QUESTION c**

Write a Program to create a child process. The parent will send a signal to the child every 5 seconds and the child will handle the signal and check if an input number is a leap year or not.

**Source Code**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<signal.h>

#include<fcntl.h>

int received=0;

void leap(int y)

{

    if(y%400==0) printf("Leap Year\n");

    else if(y%100!=0 && y%4==0) printf("Leap Year\n");

    else printf("Not Leap Year\n");

}

void custom\_handler(int signum)

{

    received=1;

}

void main()

{

    int pid,n,i=1;

    signal(SIGCHLD,custom\_handler);

    pid=fork();

    if(pid==-1)

    {

        perror("fork");

        exit(1);

    }

    if(pid==0)

    {

        printf("Child process with PID : %d and PPID : %d started..\n",getpid(),getppid());

        while(1)

        {

            pause();

            if(received==1)

            {

                printf("Child Process %d : Enter year to check leap year or not : ",getpid());

                scanf("%d",&n);

                leap(n);

                received=0;

            }

        }

    }

    else

    {

        printf("Parent process with PID : %d started..\n",getpid());

        while(i==1)

        {

            sleep(5);

            printf("Enter 1 to continue : ");

            scanf("%d",&i);

            if(i==0) continue;

            printf("Parent sending signal to child with PID %d ..\n",pid);

            kill(pid,SIGCHLD);

        }

    }

}

**Output**

