Assignment 7

7A:

#include<stdio.h>

#include<signal.h>

typedef void (\*signalhandler\_t)(int);

signalhandler\_t signal(int signal\_num, signalhandler\_t handler);

void fun1(int signum)

{

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

}

//whatever you want to do to handle the signal}

int main ()

{

signal (SIGINT, fun1);

while (1)

{

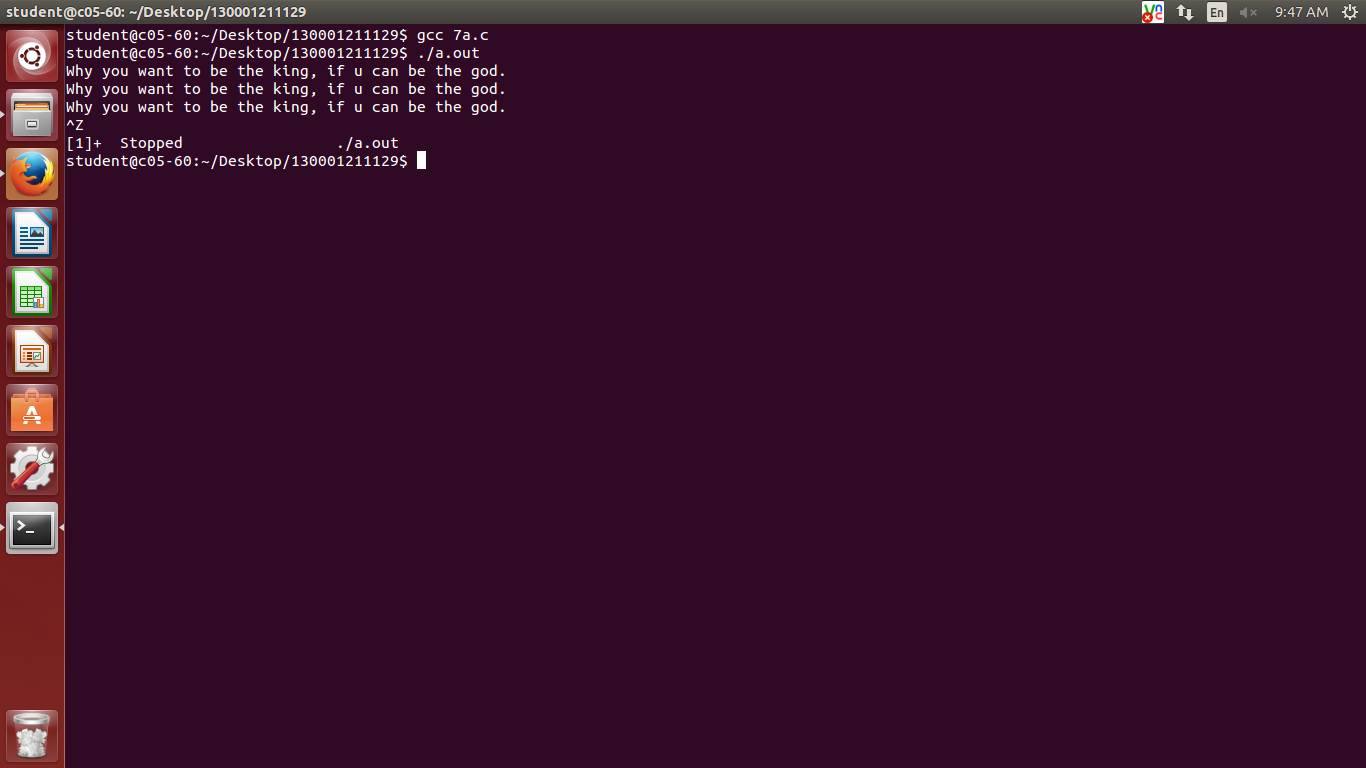
printf("Why you want to be the king, if u can be the god.\n");

sleep(1);//to sync the output}} Output:

}

}

Output:



7b:

#include<stdio.h>

#include<stdlib.h>

#include<signal.h>

void fun1(int signum)

{

// whatever you want to do to handle the signal

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

}

int main()

{

int pid = fork();

// child thread create

if(pid == 0)

{

// child

sleep(5);

kill(getppid(), SIGALRM); // sending signal to parent

exit(0);

}

else {signal(SIGALRM, fun1);

while (1)

{

printf("Knock Knock\n");

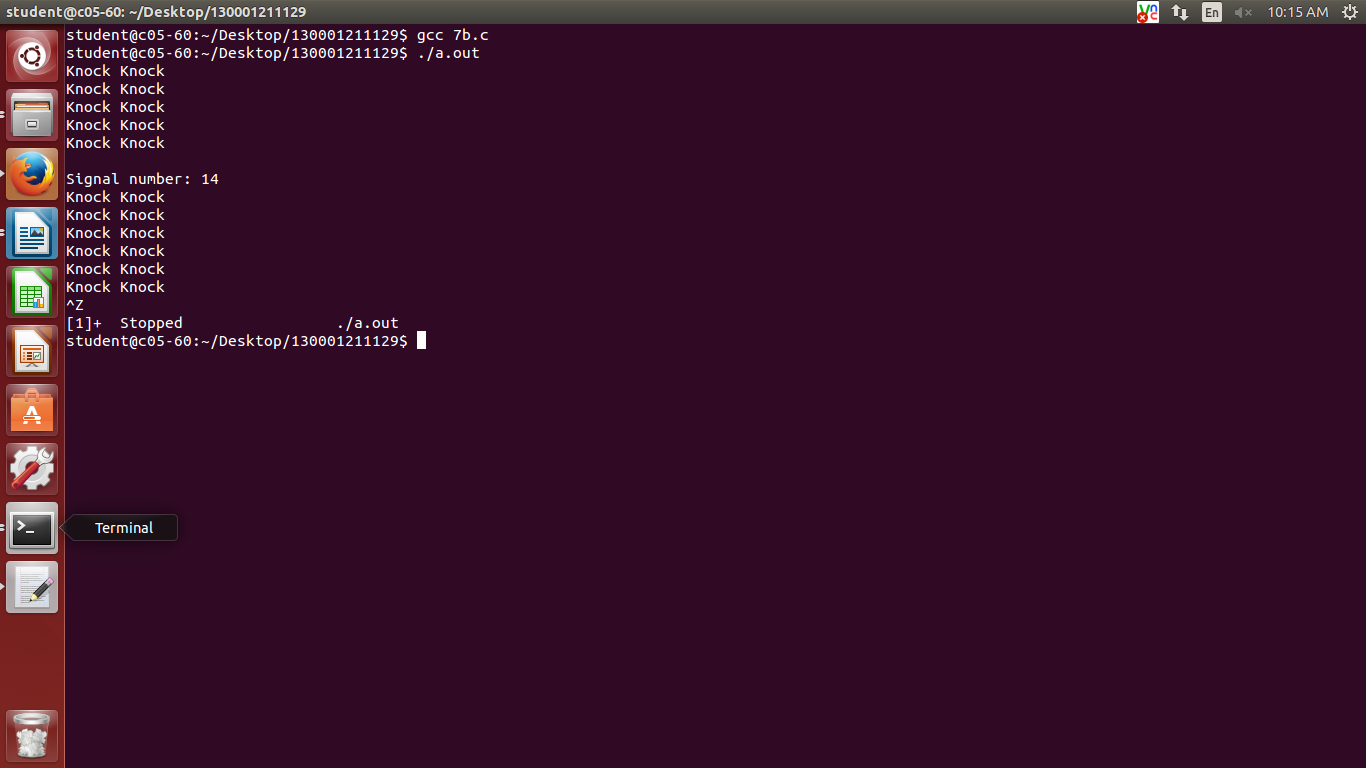
sleep(1); // to sync the output

}

}

}

Output:



7c:

#include<stdio.h>

#include<stdlib.h>

#include<signal.h>

void fun1(int signum)

{

// here give the code for leap year check

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

}

int main()

{

int a;

printf("Enter the year: ");

scanf("%d",&a);

if (a%4==0 || a%400==0 )

{

printf("%d is leap year\n",a);

}

else if(a%100==0)

{

printf("%d is not a leap\n",a);

}

else

{

printf("Not a leap\n");

}

int pid = fork(); // child thread create

if(pid == 0)

{

// child

while(1)

{

signal(SIGALRM, fun1); // handling alarm

}

}

else {

while (1)

{

printf("A Message.\n");

sleep(5); // to sync the output

kill(pid, SIGALRM);

}

}

}

Output:

