Name: K V Sai Teja

Section:K1609

Roll No:14

Q.No:1,26.

Link:

Write a C program using the fork() system call that generates this sequence in the child process. The starting number will be provided from the user. For example, if 8 is passed as a parameter on the command line, the child process will output 8*,* 4*,* 2*,* 1. Because the parent and child processes have their own copies of the data, it will be necessary for the child to output the sequence. Have the parent invoke the wait() call to wait for the child process to complete before exiting the program. Perform necessary error checking to ensure that a positive integer is passed on the command line.

Code:

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

int main()

{

int n;

X:

printf("Enter the integer:");

scanf("%d", &n);

if(n<=0)

{

goto X;

}

pid\_t p;

p = fork();

if(p == 0)

{

int i=n;

while(i>0)

{

printf("%d",i);

i=i/2;

}

}

else

{

wait(NULL);

}

}

Algorithm:

int main()

{

int n;

X:

printf("Enter the integer:");

scanf("%d", &n);

if(n<=0)

{

goto X;

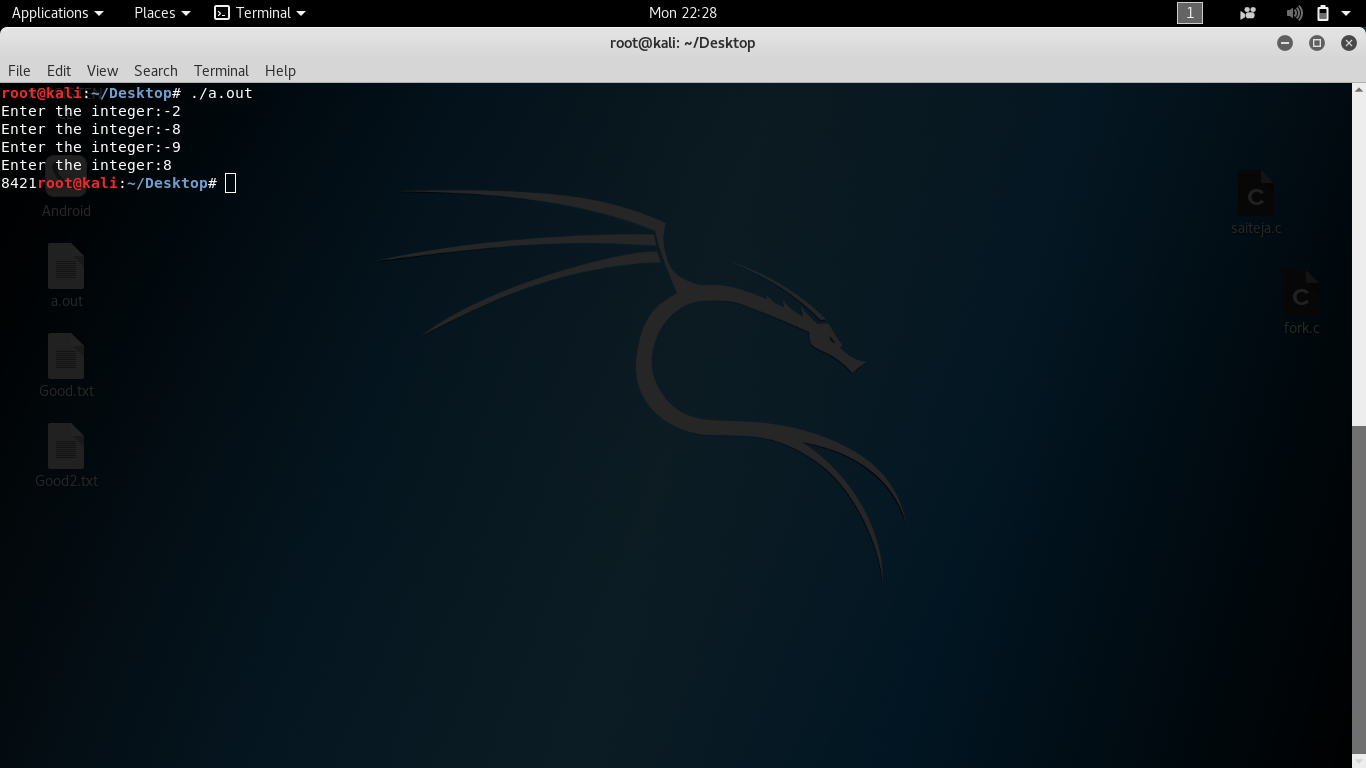
}

if number n is entered if n<=0 then it again goto back

if n>0 then child and parent process is created p==0 then went into the loop .

sequence is created. until child is excuted parent will be wait

Test cases:



Design a file-copying program named filecopy using ordinary pipes. This program will be passed two parameters: the name of the file to be copied and the name of the copied file. The program will then create an ordinary pipe and write the contents of the file to be copied to the pipe. The child process will read this file from the pipe and write it to the destination file. For example, if we invoke the program as follows:

filecopy input.txt copy.txt

The file input.txt will be written to the pipe. The child process will read the contents of this file and write it to the destination file copy.txt.

Code:

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<fcntl.h>

#include<sys/types.h>

int main()

{

int fd[2], n, fds;

char buffer[100];

pipe(fd);

fds = open("Good.txt", O\_CREAT | O\_RDWR, 0777);

if(fds == -1)

{

printf("Error while opening file:");

exit(1);

}

printf("Enter the message:");

n = read(0, buffer, 100);

write(fds, buffer, 100);

close(fds);

fds = open("Good.txt", O\_RDWR);

if(fds == -1)

{

printf("Error while opening the file.");

exit(1);

}

read(fds, buffer,n);

write(fd[1], buffer, n);

close(fds);

char buf[100];

fds = open("Good2.txt", O\_CREAT|O\_RDWR);

if(fds == -1)

{

printf("Error while creating file..");

exit(1);

}

read(fd[0], buf, n);

write(fds, buf, n);

close(fds);

}

Alogirthm:

pipe(fd)

pipe is

file discriptor is created

fds = open("Good.txt", O\_CREAT | O\_RDWR, 0777);

fds==-1

error while opening file

printf("Enter the message:");

enter the message

n = read(0, buffer, 100);

write(fds, buffer, 100);

read and write message

fds = open("Good.txt", O\_RDWR);

open the file

In this read the message in the file and store in the buffer

and write in the pipe

read(fds, buffer,n);

write(fd[1], buffer, n);

fds = open("Good2.txt", O\_CREAT|O\_RDWR);

another file is create in read and write mode

read(fd[0], buf, n);

write(fds, buf, n);

close(fds);

from pipe read the message and write it in the file.

