

Lab Tasks

By

Maryum Malik (45369)



Submitted to: Ma'am Kausar

Subject: Operating System

Date: 10/8/2024

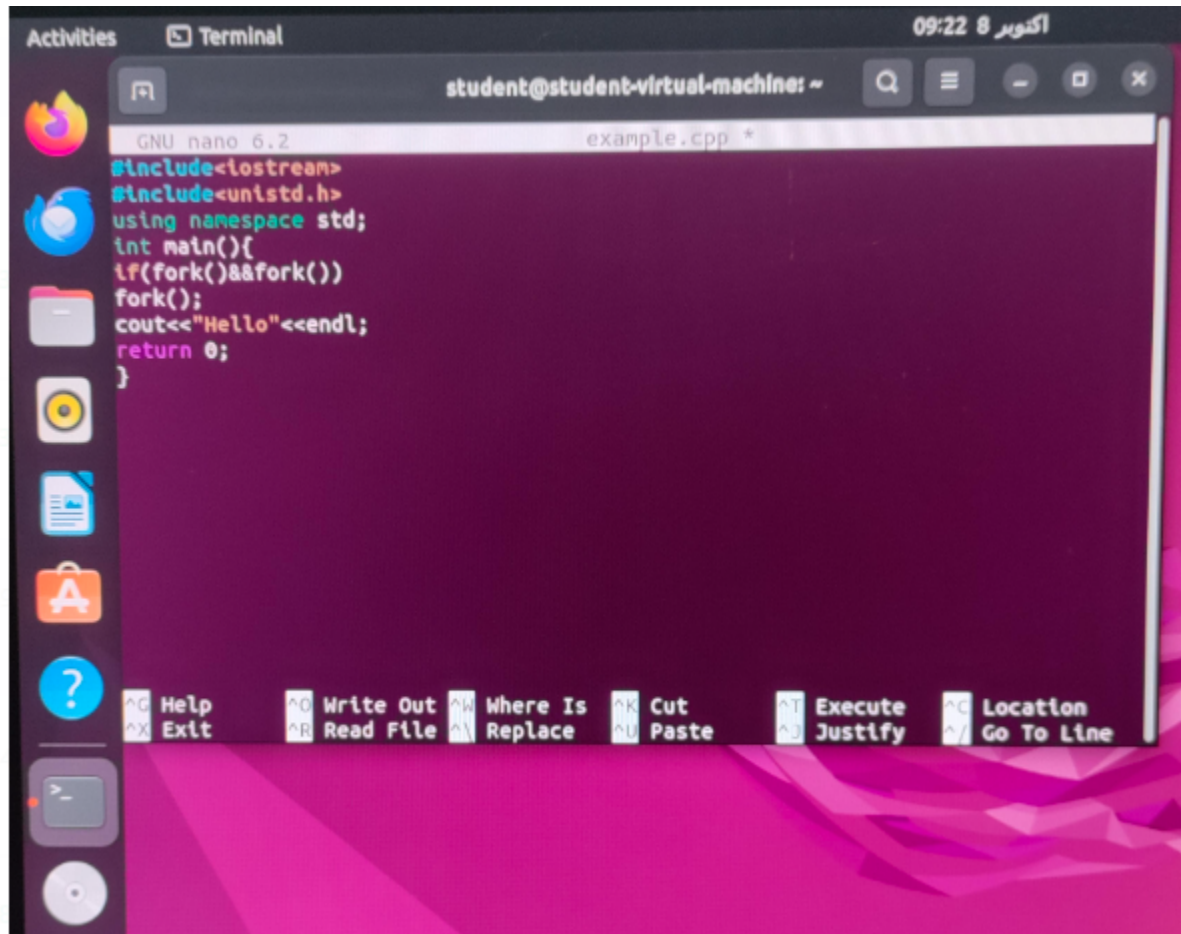
BSCS SEMESTER – 5

RIPHAH INTERNATIONAL UNIVERSITY

ISLAMABAD, PAKISTAN

Task

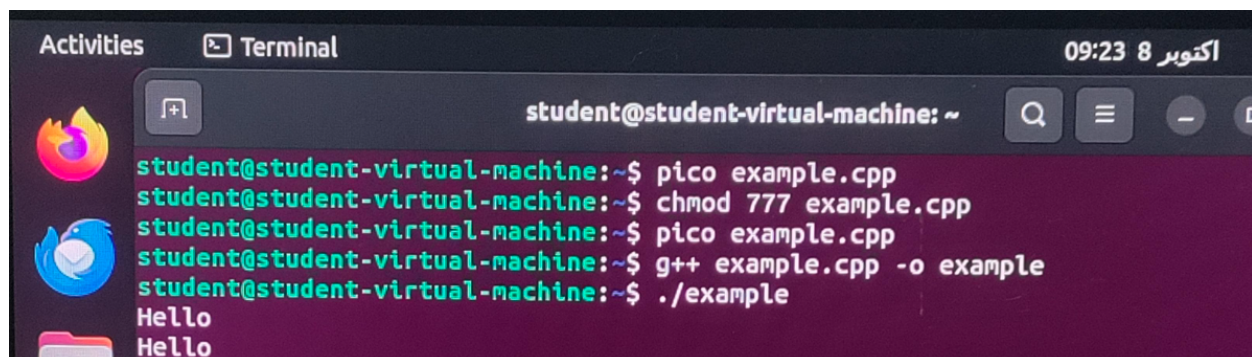
1. Write a C/C++ program that uses the `fork()` function and the logical AND (`&&`) operator.



A screenshot of a terminal window on a Linux system. The window title is "student@student-virtual-machine: ~". The terminal shows the GNU nano 6.2 editor editing a file named "example.cpp". The code in the file is as follows:

```
#include<iostream>
#include<unistd.h>
using namespace std;
int main(){
    if(fork()&&fork())
        fork();
    cout<<"Hello"<<endl;
    return 0;
}
```

The terminal window has a sidebar on the left with icons for various applications. At the bottom, there is a menu bar with options: Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

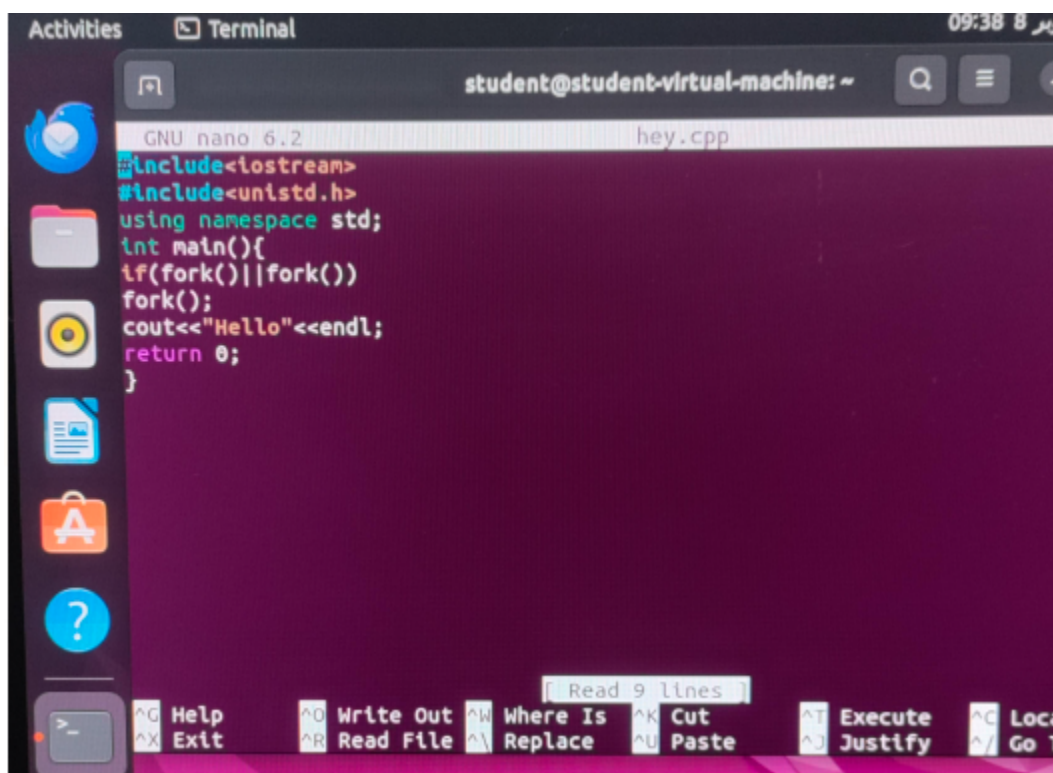


A screenshot of a terminal window on a Linux system. The window title is "student@student-virtual-machine: ~". The terminal shows the following commands and output:

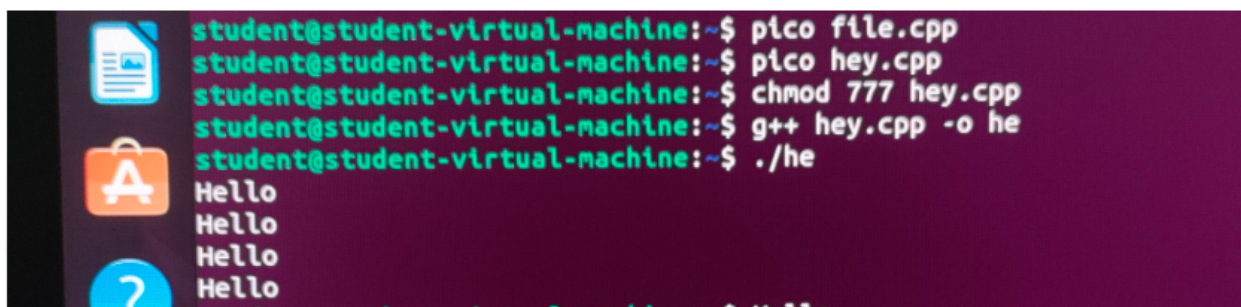
```
student@student-virtual-machine:~$ pico example.cpp
student@student-virtual-machine:~$ chmod 777 example.cpp
student@student-virtual-machine:~$ pico example.cpp
student@student-virtual-machine:~$ g++ example.cpp -o example
student@student-virtual-machine:~$ ./example
Hello
Hello
```

In AND operator if both conditions are true then it will proceed same in this program fork means create child. First fork creates child (0) and parent (1). The second fork is called for Parent (1) child (0) will be terminated. Then the parent (1) will be divided into Child (0) and Parent (1). Again, last fork is only called for parent (1+1 = true) Child is terminated. Parent is divided into Child (0) and Parent (1). In this way it will give in output 2 hello.

2. Write a C/C++ program that uses the fork() function and the logical OR (II) operator.



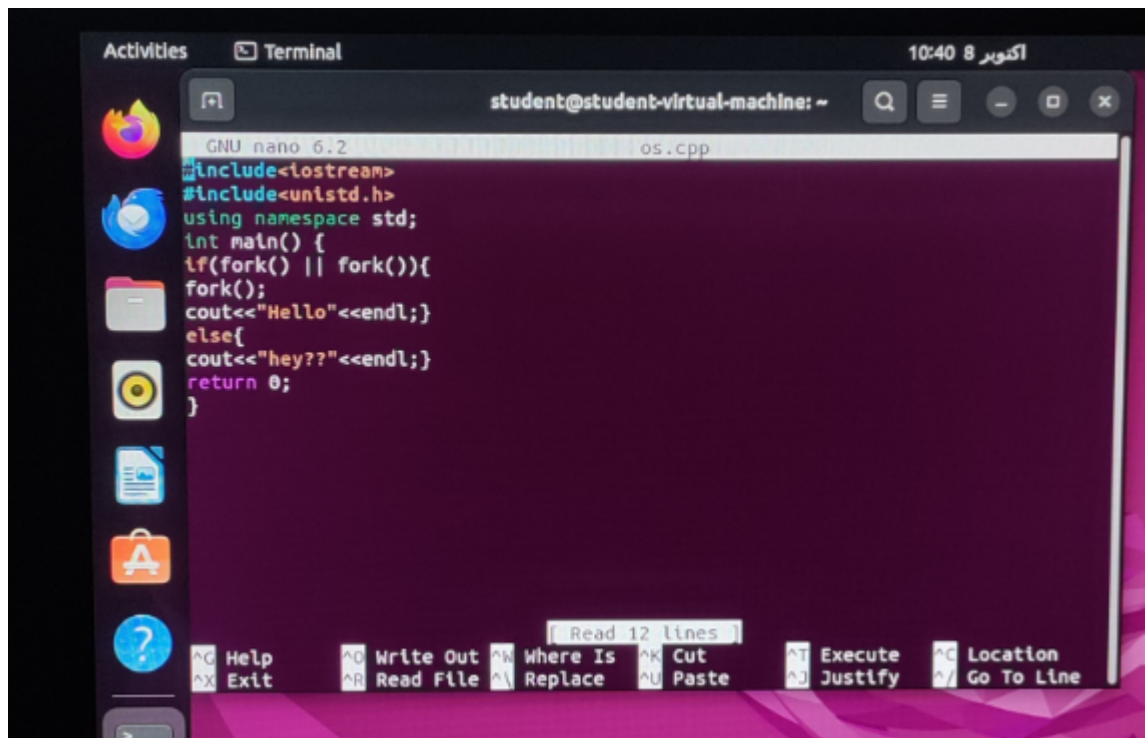
```
GNU nano 6.2 hey.cpp
#include<iostream>
#include<unistd.h>
using namespace std;
int main(){
    if(fork()||fork())
        fork();
    cout<<"Hello"<<endl;
    return 0;
}
```



```
student@student-virtual-machine:~$ pico file.cpp
student@student-virtual-machine:~$ pico hey.cpp
student@student-virtual-machine:~$ chmod 777 hey.cpp
student@student-virtual-machine:~$ g++ hey.cpp -o he
student@student-virtual-machine:~$ ./he
Hello
Hello
Hello
Hello
```

In OR operator if any of one is true, the result will be true then it will proceed same in this program fork means create child. First fork creates child (0) and parent (1). The second fork is called for both Parent (1) child (1). Then the parent (1) will be divided into Child (0) and Parent (1) and Child (1) will be divided into Child (0), Child (0). One condition in child (1) is true ($0+1=\text{true}$). It will further divide into Child (0) and Child (0) this is the last call of fork. In this way 4 times hello is printed.

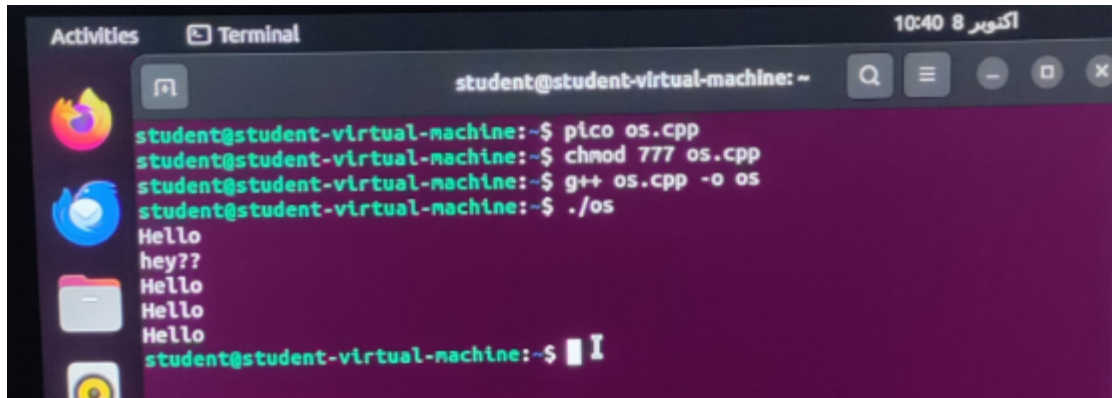
3. Write a C++ program that uses `fork()` to create a child process. Use an if-else statement.



The screenshot shows a terminal window titled "student@student-virtual-machine: ~" with a clock showing 10:40. The terminal is running the GNU nano 6.2 editor, editing a file named "os.cpp". The code in the editor is as follows:

```
#include<iostream>
#include<unistd.h>
using namespace std;
int main() {
    if(fork() || fork()){
        fork();
        cout<<"Hello"<<endl;
    }
    else{
        cout<<"hey??"<<endl;
    }
    return 0;
}
```

At the bottom of the terminal, there is a status bar with the text "[Read 12 lines]" and a row of keyboard shortcuts: ^G Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^\ Replace, ^K Cut, ^U Paste, ^T Execute, ^J Justify, ^_ Location, and ^_ Go To Line.



```
student@student-virtual-machine: ~  
student@student-virtual-machine:~$ pico os.cpp  
student@student-virtual-machine:~$ chmod 777 os.cpp  
student@student-virtual-machine:~$ g++ os.cpp -o os  
student@student-virtual-machine:~$ ./os  
Hello  
hey??  
Hello  
Hello  
Hello  
Hello  
student@student-virtual-machine:~$
```

In this program we have used AND operator with if else case means if case fork fails then else case will run. AND the operator will work if one of the conditions is false it will false.