RIPHAH INTERNATIONAL UNIVERSITY, ISLAMABAD

Lab # 8

Bachelors of Computer Science – 6th Semester

Subject: Operating System

Submitted to: Ms. Kausar

Submitted by: Ayesha Noor _ 41379

Date of Submission: 08- Oct -2024

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

```
student@student-virtual-machine:~$ nano programm.cpp
```

```
 GNU nano 6.2                                programm.c
#include<iostream>
#include<unistd.h>
using namespace std;
int main(){

if(fork() && fork()){
fork();
cout<<"Hello";
}

return 0;
}
```

```
student@student-virtual-machine:~$ chmod 777 programm.cpp
student@student-virtual-machine:~$ g++ -o programm programm.cpp
student@student-virtual-machine:~$ ./programm
HelloHellostudent@student-virtual-machine:~$ nano programm.cpp
```

Explanation: With the logical AND **(&&)**, both **fork()** calls must return non-zero (parent processes) for the code inside the if block to execute. Only if both **fork()** calls create child processes will the block be skipped. If the block is entered, multiple processes print "Hello".

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

```
student@student-virtual-machine:~$ nano programm.cpp
```

```
 GNU nano 6.2                                programm.cpp
#include<iostream>
#include<unistd.h>
using namespace std;
int main(){

if(fork() || fork()){
fork();
cout<<"Hello";
}

return 0;
}
```

```
HelloHellostudent@student-virtual-machine:~$ g++ -o programm programm.cpp
student@student-virtual-machine:~$ ./programm
HelloHelloHelloHellostudent@student-virtual-machine:~$ nano programm.cpp
```

Explanation: The logical **OR (||)** operator returns true if at least one of its operands is true. The code uses fork() to create multiple processes, and the logical OR (||) determines if the second fork() is executed. Each process that enters the if block prints "Hello", resulting in multiple outputs from different processes.

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

```
student@student-virtual-machine:~$ nano programm.cpp
```

```
#include<iostream>
#include<unistd.h>
using namespace std;
int main(){
int num , fd;
cout<<"Enter any number ";
cin>>num;
fd=fork();
fd=num;
if(fd > 0 && fd < 10){
fork();
fork();
cout<<"Hello ";
}
else{
cout<<"Number is greater than 10 ";
}
return 0;
}
```

```
student@student-virtual-machine:~$ g++ -o programm programm.cpp
student@student-virtual-machine:~$ ./programm
Enter any number 7
Hello Hello Hello Hello Hello Hello Hello Hello student@student-virtual-machine:
~$
```

Explanation:

In this program:

- The **fork()** call creates a child process.
- The **if-else** block distinguishes between the parent and child processes based on the returned.