



## **Lab # 08**

### **Operating System**

Name: Samreen Bibi

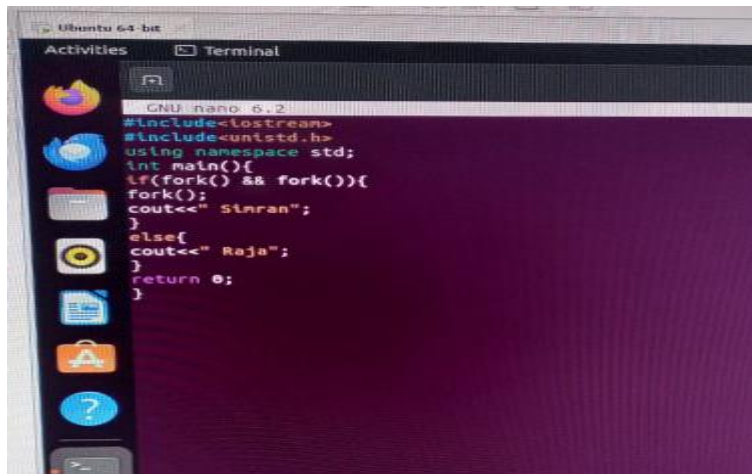
Sap Id:46484

Batch: BSCS-5<sup>th</sup> semester

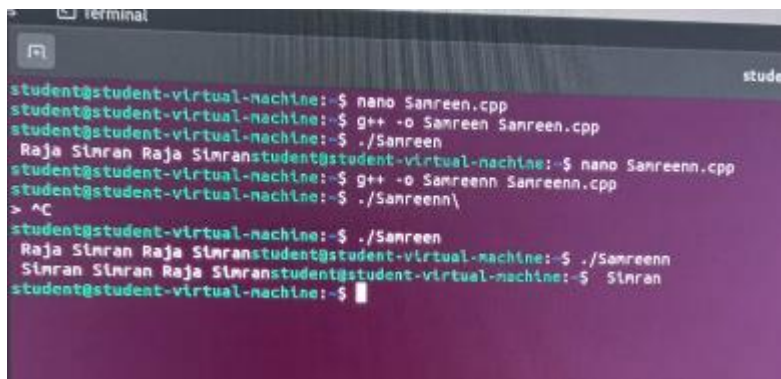
Submitted to: Kauser Nasreen

**Q1.**

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



```
GNU nano 6.2
#include<iostream>
#include<unistd.h>
using namespace std;
int main(){
    if(fork() && fork()){
        fork();
        cout<<" Sinran";
    }
    else{
        cout<<" Raja";
    }
    return 0;
}
```



```
student@student-virtual-machine:~$ nano Sanreen.cpp
student@student-virtual-machine:~$ g++ -o Sanreen Sanreen.cpp
student@student-virtual-machine:~$ ./Sanreen
Raja Sinran Raja Sinran
student@student-virtual-machine:~$ nano Sanreenn.cpp
student@student-virtual-machine:~$ g++ -o Sanreenn Sanreenn.cpp
student@student-virtual-machine:~$ ./Sanreenn\
> ^C
student@student-virtual-machine:~$ ./Sanreen
Raja Sinran Raja Sinran
student@student-virtual-machine:~$ ./Sanreenn
Sinran Sinran Raja Sinran
student@student-virtual-machine:~$
```

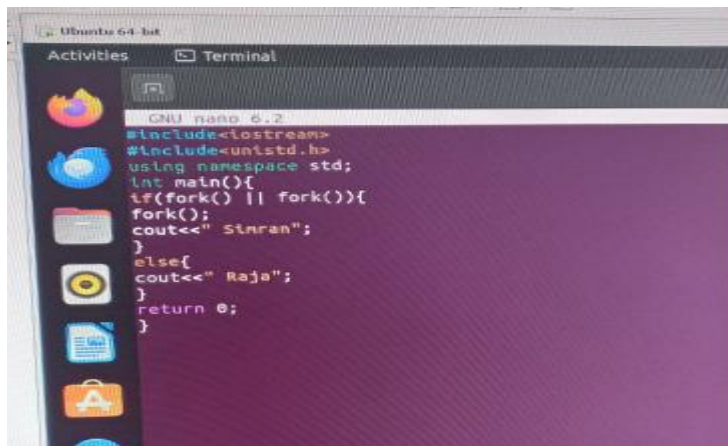
Fork () creates a new process (child process) that runs the same code as the parent process but independently. The || (And) operator means that if both of the fork () calls creates a child process (i.e., returns 0), the condition becomes true. This leads to running the code inside the if block. If the condition is true, a third fork () is called, creating another child process.

else block:

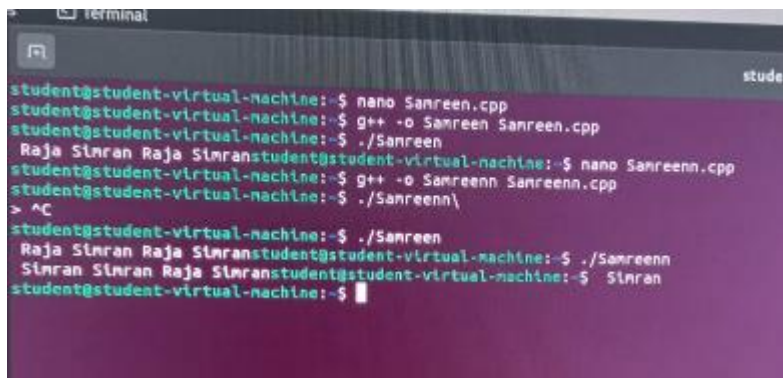
If the condition is false (both fork () calls return 0).

**Q2.**

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



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



```
student@student-virtual-machine:~$ nano Sanreen.cpp
student@student-virtual-machine:~$ g++ -o Sanreen Sanreen.cpp
student@student-virtual-machine:~$ ./Sanreen
Raja Sinran Raja Sinran
student@student-virtual-machine:~$ nano Sanreenn.cpp
student@student-virtual-machine:~$ g++ -o Sanreenn Sanreenn.cpp
student@student-virtual-machine:~$ ./Sanreenn\
> ^C
student@student-virtual-machine:~$ ./Sanreen
Raja Sinran Raja Sinran
student@student-virtual-machine:~$ ./Sanreenn
Sinran Sinran Raja Sinran
student@student-virtual-machine:~$
```

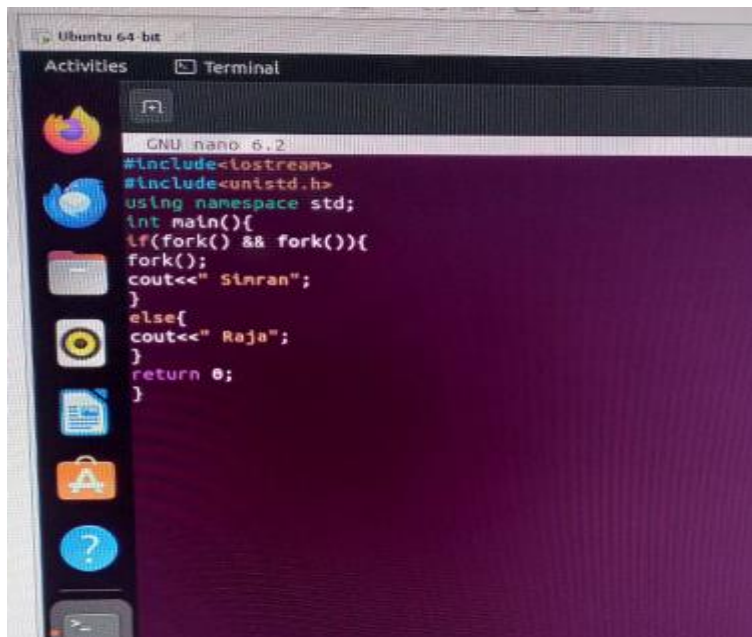
Fork () creates a new process (child process) that runs the same code as the parent process but independently. The || (OR) operator means that if either of the fork () calls creates a child process (i.e., returns 0), the condition becomes true. This leads to running the code inside the if block. If the condition is true, a third fork () is called, creating another child process.

else block:

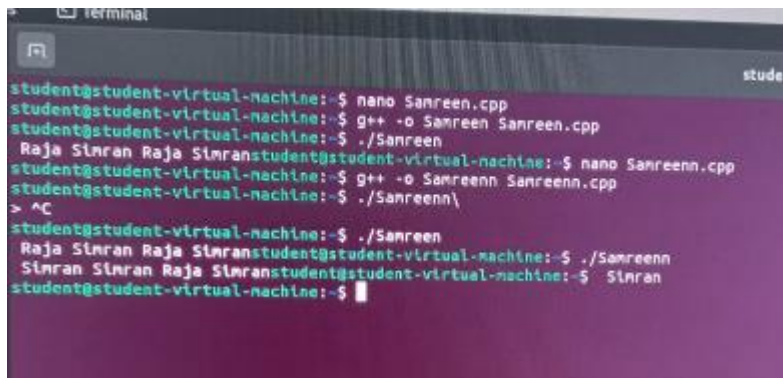
If the condition is false (both fork () calls return 0).

Q3.

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



```
GNU nano 6.2
#include<iostream>
#include<unistd.h>
using namespace std;
int main(){
    if(fork() && fork()){
        fork();
        cout<<" Sinran";
    }
    else{
        cout<<" Raja";
    }
    return 0;
}
```



```
student@student-virtual-machine:~$ nano Sanreen.cpp
student@student-virtual-machine:~$ g++ -o Sanreen Sanreen.cpp
student@student-virtual-machine:~$ ./Sanreen
Raja Sinran Raja Sinran
student@student-virtual-machine:~$ nano Sanreenn.cpp
student@student-virtual-machine:~$ g++ -o Sanreenn Sanreenn.cpp
student@student-virtual-machine:~$ ./Sanreenn\
> ^C
student@student-virtual-machine:~$ ./Sanreen
Raja Sinran Raja Sinran
student@student-virtual-machine:~$ ./Sanreenn
Sinran Sinran Raja Sinran
student@student-virtual-machine:~$
```

Fork () creates a new process (child process) that runs the same code as the parent process but independently. The || (OR) operator means that if either of the fork () calls creates a child process (i.e., returns 0), the condition becomes true. This leads to running the code inside the if block. If the condition is true, a third fork () is called, creating another child process.

else block:

If the condition is false (both fork () calls return 0).

