**Operating Systems Lab**

**Fall 2024**

**Lab Task 08:**

****

**Lab Instructor:**

**Kausar Nasreen Khattak**

**Name: Faareha Raza**

**Sap id:47431**

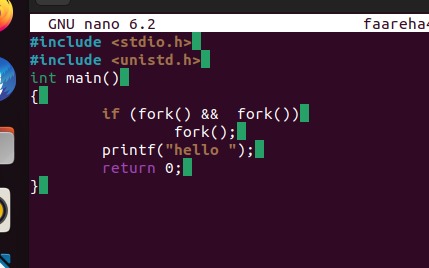
**Email:**

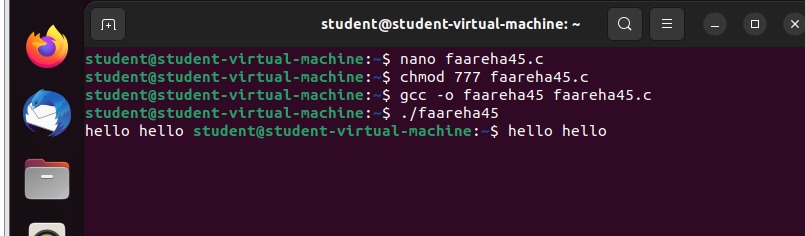
**kausar.nasreen@riphah.edu.pk**

**Lab Task**

**Note:** Include screenshots, required to illustrate your explanation for all Questions.

Q1: Write a C/C++ program that uses the fork() function and the logical AND (&&) operator.



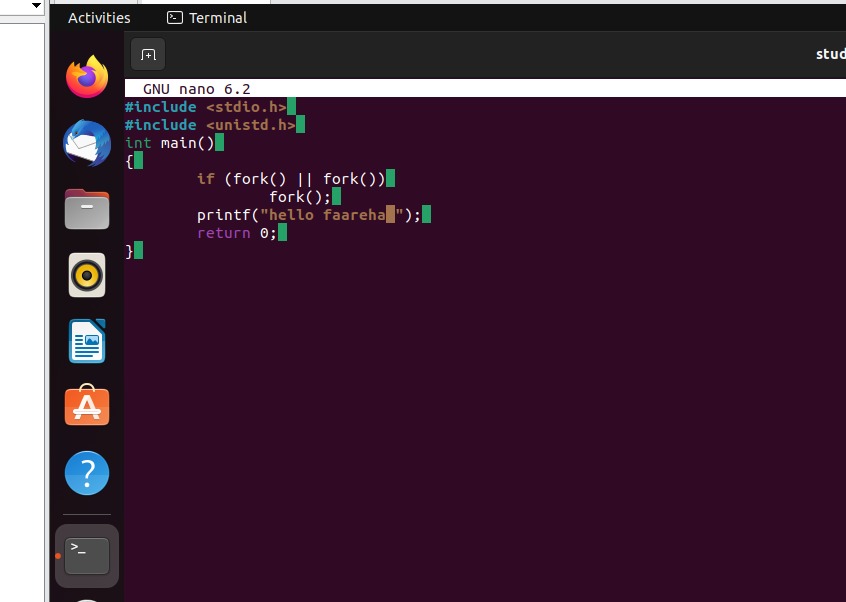


1. **fork() system call**: Each time fork() is called, it creates a new child process. Both the parent and child processes continue executing the next lines of code.
2. **Logical condition (if (fork() && fork()))**:
   * The first fork() creates a child process.
   * The second fork() in the && condition creates another child process.
   * If both fork() return a non-zero value (i.e., in the parent process), it enters the if block and performs another fork().
3. **printf("hello ");**: This line prints "hello " every time it is executed by any process (parent or child).

**Output:**

* The commands shown are:
  + nano faareha45.c: Opens the file faareha45.c in the nano editor.
  + chmod 777 faareha45.c: Gives read, write, and execute permissions to the file.
  + gcc -o faareha45 faareha45.c: Compiles the C program into an executable named faareha45.
  + ./faareha45: Runs the compiled program.
* The output hello hello shows up because multiple processes are created due to fork(), each printing "hello ".

Q2: Write a C/C++ program that uses the fork() function and the logical OR (II) operator.





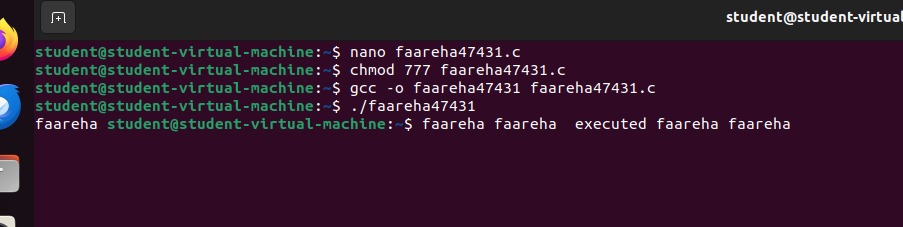
1. **fork() system call**: Each time fork() is called, it creates a new child process. Both the parent and child processes continue executing the next lines of code.
2. **Logical condition (if (fork() && fork()))**:
   * The first fork() creates a child process.
   * The second fork() in the && condition creates another child process.
   * If both fork() return a non-zero value (i.e., in the parent process), it enters the if block and performs another fork().
3. **printf("hello ");**: This line prints "hello faareha " every time it is executed by any process (parent or child).

### Output:

* The commands shown are:
  + nano raza.c: Opens the file faareha45.c in the nano editor.
  + chmod 777 raza.c: Gives read, write, and execute permissions to the file.
  + gcc -o raza raza.c: Compiles the C program into an executable named raza.
  + ./raza: Runs the compiled program.
* The output hello faareha hello faareha shows up because multiple processes are created due to fork(), each printing "hello faareha ".

Q3: Write a C++ program that uses fork() to create a child process. Use an if-else statement.





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