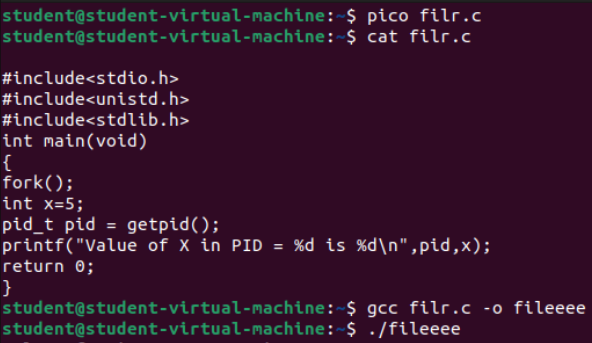


**OPERATING SYSTEM LAB**

**TAYYABA REHMAN 49690**

**LAB 8**

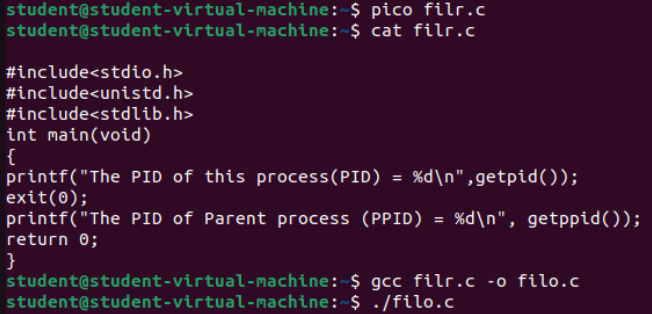
**CODE 1:**



**OUTPUT:**



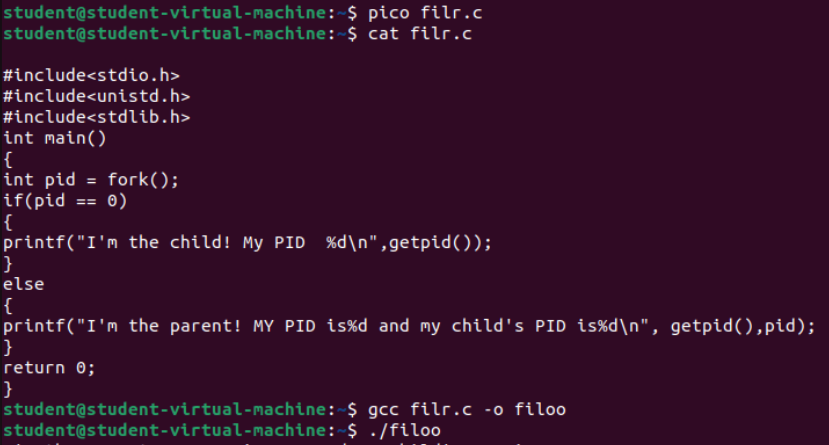
**CODE 2:**



**OUTPUT:**



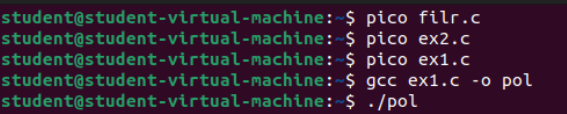
**Code 3:**



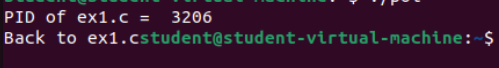
**OUTPUT:**



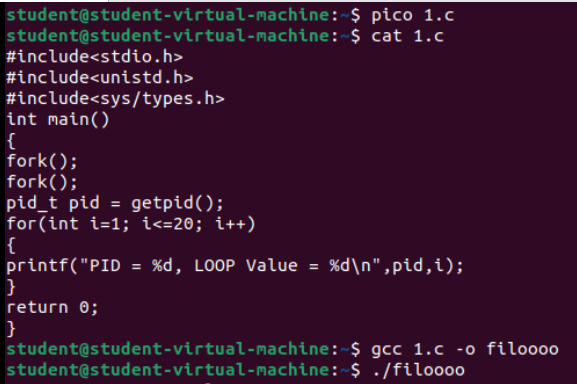
**Code 4:**



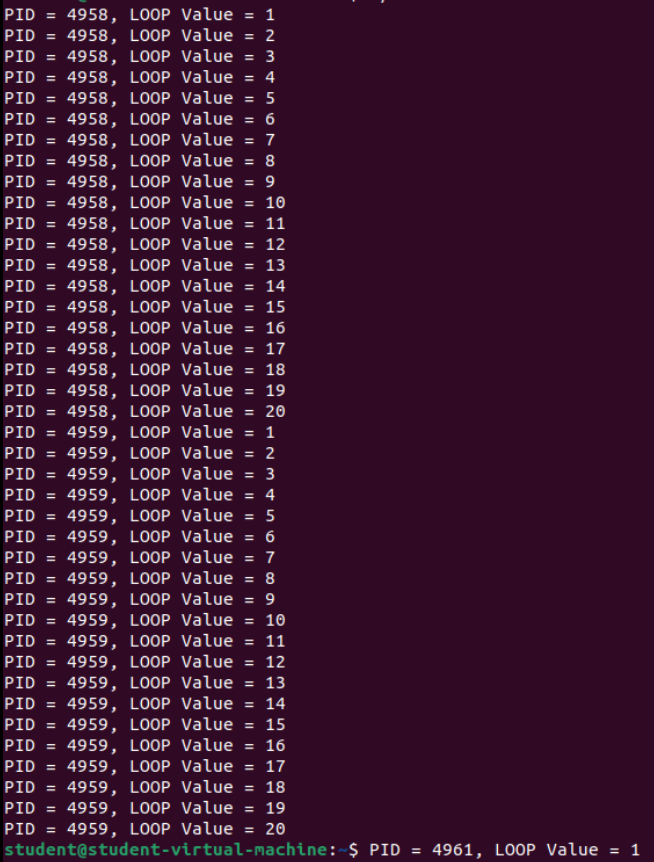
**OUTPUT:**

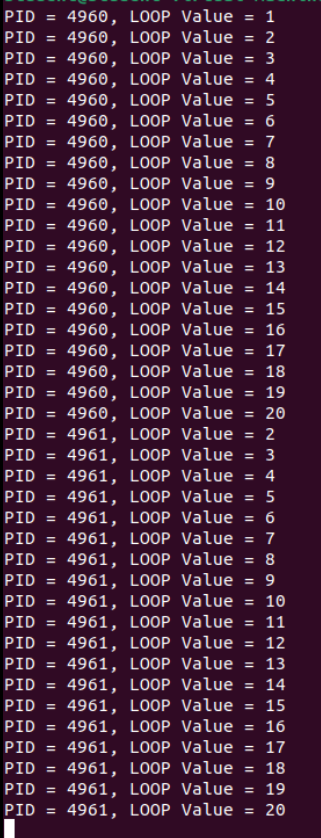


**Task 1:**

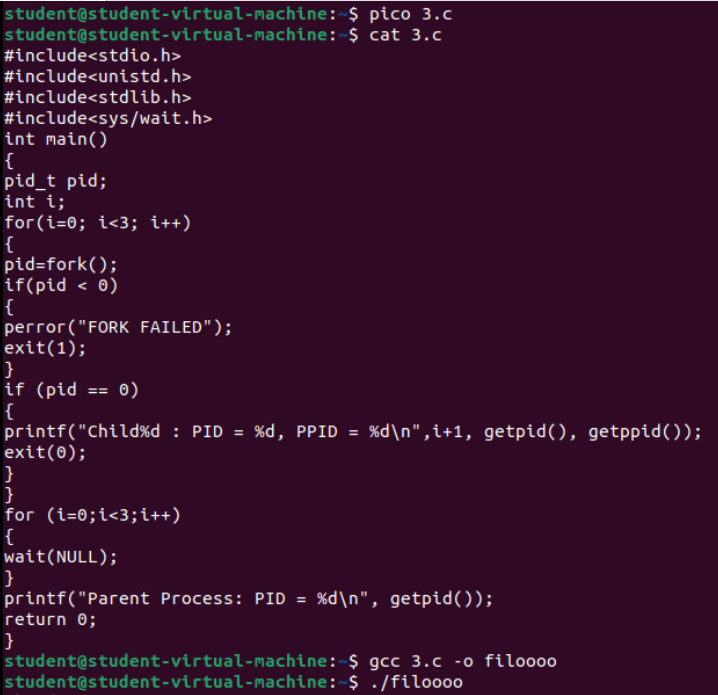


**OUTPUT:**

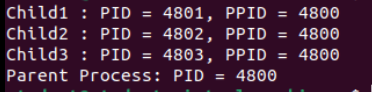




**Task 2:**



**OUTPUT:**



**Task 3:**

**Common System Calls:**

* **Process Management**
* **getpid()**  
  Returns the **Process ID (PID)** of the calling process.
* **getuid()**  
  Returns the **User ID (UID)** of the user who owns the process.
* **getgid()**  
  Returns the **Group ID (GID)** of the group that owns the process.
* **File I/O**
* **create()**  
  Creates a new file or rewrites an existing file. (In some systems, this is a wrapper over open() with specific flags.)
* **open()**  
  Opens a file for reading, writing, or both. Returns a **file descriptor** used to access the file.
* **close()**  
  Closes an opened file descriptor, freeing the resource for reuse.

### ****Process Control****

### ****fork()**** Creates a ****new process**** by duplicating the calling process. The new process is called the ****child process****.

### ****exec()**** Replaces the current process image with a ****new** **program****. It is used to run a new program within the same process.

### ****wait()**** Makes the parent process ****wait until one of its child processes terminates****. It collects the child's exit status.

### ****Signal Handling****

### ****kill()**** Sends a ****signal**** to a process or group of processes. Often used to terminate a process.

### ****alarm()**** Schedules a ****SIGALRM**** signal to be sent to the process after a specified number of seconds. Useful for timers.