 **BAHRIA UNIVERSITY (KARACHI CAMPUS)**

**ASSIGNMENT # 1 - FALL 2020**

# Operating Systems (CSC-320)

Class: **BSE 4 A&B** Submission Deadline: **17/18 March, 2020**

Course Instructor: **Dr. Osama Rehman**

Lab Instructor: **Engr. Fareeha Dilawar** Max Marks: **10**

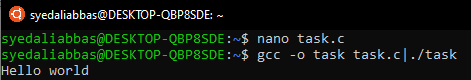
Student’s Name: **Syed Ali Abbas**

1. Study and implement pipes in Linux. List down three examples with outputs attached in the screenshots.

**Pipes in Linux:-**

* Pipe is a command in Linux that uses two or more commands such that output of one command serves as input for the next. The output of each process directly as input to the next one like a pipeline.
* Pipe is denoted by ‘|’.

**Example: 1**



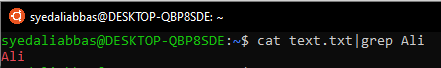
**Example: 2 The 'grep' command**

Scan the document for the desired information and gives output in the required format.

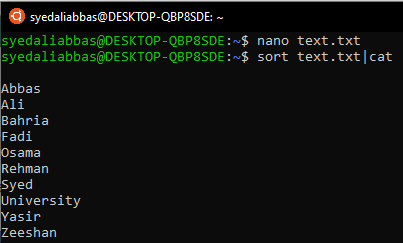
**Input**



**Output**

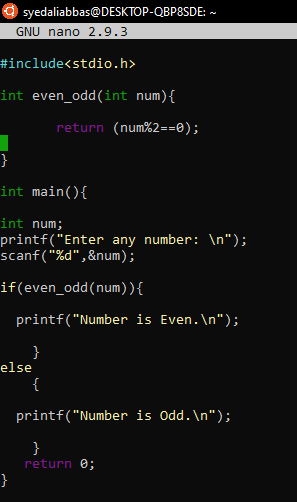


**Example: 3** Sort and cat command to print file data in a sorted manner.

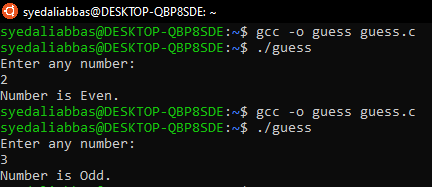


1. Write a C program that asks users to provide an integer input in the main() function. The program would call a function even\_odd() from the main() function, where the function even\_odd()accepts an integer as an argument, determine and display if the passed integer is either even or odd.

**Input:**



**Output:**



1. Study and implement system calls in Linux. List down three examples with output screenshot attached.

**System calls in Linux:-**

* A **system call** is a mechanism that provides the interface between a process and the operating system.
* Computer program requests a service from the kernel of the OS.

**Services Provided by System Calls:-**

1. Process creation and management
2. Main memory management
3. File Access, Directory and File system management
4. Device handling(I/O)
5. Protection
6. Networking, etc.

**Types of System Calls:-**

 There are 5 different categories of system calls.

1. **Process control:**

* End, abort, create, terminate, allocate and free memory.

1. **File management:**

* Create, open, close, delete, read file etc.

1. **Device management:**

* Request and release device, Logically attach/ detach devices, Get and Set device attributes

1. **Information maintenance:**

* Get or set time and date, Get process and device attributes

1. **Communication:**

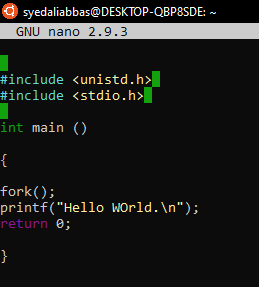
* Create, delete communications connections, send, receive message, help OS to transfer status information, attach or detach remote devices

**Example: 1**

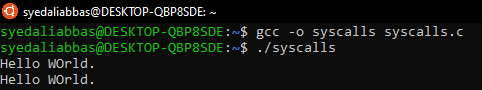
**fork ()**

Processes use this system call to create processes that are a copy of themselves. With the help of this system Call parent process creates a child process, and the execution of the parent process will be suspended till the child process executes.

**Input:**



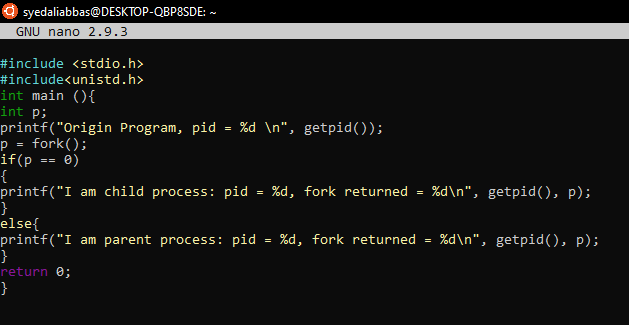
**Output:**



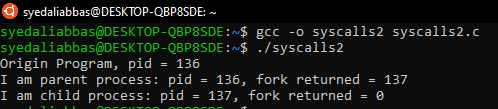
**Example: 2**

With fork(), the parent can learn the process id of the child, but the child doesn't know its own process id (or pid) after the fork nor does it know its parents process id. For that matter, the parent doesn't know its own process id either. There are two system calls to retrieve this information:

**Input:**



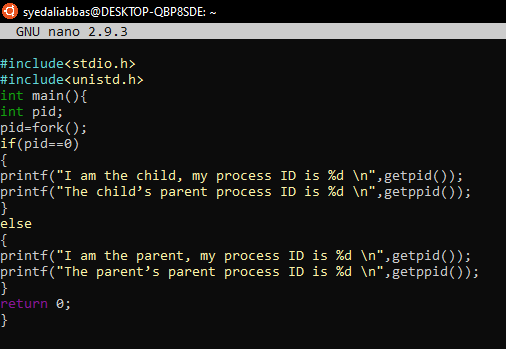
**Output:**



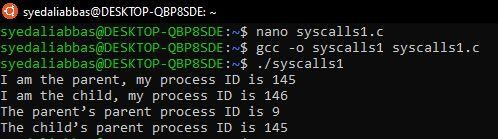
**Example: 3**

With fork(), the parent can learn the process id of the child, but the child doesn't know its own process id (or pid) after the fork nor does it know its parents process id. For that matter, the parent doesn't know its own process id either. There are two system calls to retrieve this information:

**Input:**



**Output:**



1. Think and propose two projects that can be delivered as the final lab projects in Operating Systems lab. The ideas must be valid and should be explained with research topics and examples.

* **USB Device Driver**

USB Device Driver project aims to develop a device driver for USB interface devices implementing all basic operations that can be performed with a USB such as reading, writing and transferring data between two Environments.

We will create and implement our own USB Device Driver in the Linux Environment using “Shell Scripting”. The USB client driver or the USB class-specific driver is loaded when the operating system first detects the corresponding class-specific device. It remains loaded until the last device of the corresponding class is removed from the system.

USB Device Driver will have the following functionalities:

1. **USB Functions :**

The USB - Driver adds its functions to the kernel device .It registers with the USB system driver (USBD) to determine whether the device is attached and configured .Other functionalities include reading, writing and transferring data from USB – Storage Device.

1. **Device Recognition :**

On attachment, the driver must be able to make the device ready to use.

* **Memory Management**

In this project, you need to devise a mechanism that would satisfy the memory management requirements such as allocation, protection, sharing, relocation, and its organization. These are essential for any modern operating system.

**Memory Management** is the process of controlling and coordinating computer memory, assigning portions known as blocks to various running programs to optimize the overall performance of the system. It is the most important function of an operating system that manages primary memory. It helps processes to move back and forward between the main memory and execution disk. It helps OS to keep track of every memory location, irrespective of whether it is allocated to some process or it remains free.