**Operating System Project Report**

**Project Title:** Performance Comparison for IPC using Filing, Pipes, Signals and Semaphore

**Group Members:**

Abeer Zehra 16k-4068

Sara Shakil 16k-3949

**Section:** F

**Objective:**

Our main objective of this project is to implement inter process communication using filling, pipes and semaphores and the compare their performance.

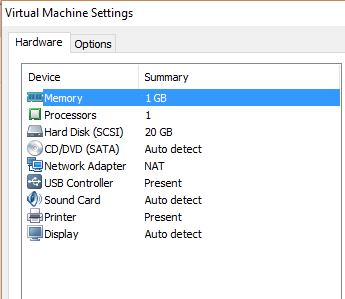
**Introduction: -**

Interprocess communication (IPC) refers specifically to the mechanics an operating system provides to allow processes to share data. IPC is very vital in any Embedded System. A program may have to feed another process for it to proceed. It is inherent in all the embedded systems. Typically, applications can use IPC, categorized as client and servers, where the client requests data and the server responds to client requests. This allows a program to handle many user requests at the same time. Each IPC method has its own advantages and limitations so it is not unusual for a single program to use all of the IPC methods.

**Programming Platform used: -**

Configuration of VMware

->C language is used as a programming platform



Ubuntu 16.4 was also used



**Methodology:**

We used linear searching on 50000 numbers as a task to implement interprocess communication in order to learn about the working and handling of large data by different methodologies of Inter process communications and to get clear idea of the performance of each type of IPC method so that we can compare them later on and do the analysis.

1. **­­­­Interprocess communication using pipes**

A pipe is very simple way of communicating between two processes. They are unidirectional i.e. data can be send in either of direction at a time if dual way communication is needed then we use two pipes.

Pipe can be created using pipe() system call. It will return two file descriptor one for read operation and other for write operation.

->Disadvantage of using unnamed pipes is that it can be used for only related processes. To overcome this issue we use named pipes.

Named pipes:

Named pipes are also called FIFO. By using named pipes we can communicate between two different processes.

Mkfifo command is used to create named pipes.

For implementing named pipes we uses two programs one for read() operation and other for write() operation. To communicate process A writes to a common file and process B can then read from that file. After read that file can be deleted.

1. **Interprocess communication using semaphores:**

Semaphores basically works by giving signals. We declare and initialize semaphores. When one process (I.e. child is done with searching) completes its task, it gives signal to the other (i.e. parent process to continue its task which is of coping data to shared memory) to start its operation.

1. **Inter process communication through filling**

We have implemented inter process communication using filling by program or you can say a processes. program reads the data from it number.txt and searches the data and then writes the data in shared.txt.

**(Time in seconds)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numbers (data)** | **FILING** | **PIPES (SIMPLE)** | **PIPES (MKFIFO)** | **Semaphores** |
| 10000 | 0.0028156 | 0.0003168 | 0.0004724 | 0.0041038 |
| 20000 | 0.0056312 | 0.0006336 | 0.0009448 | 0.0082076 |
| 30000 | 0.0084468 | 0.0009504 | 0.0014172 | 0.0123114 |
| 40000 | 0.0112624 | 0.0012672 | 0.0018896 | 0.0164152 |
| 50000 | 0.014078 | 0.001584 | 0.002362 | 0.020519 |

**Applications:**

To synchronize processes on one computer systems semaphores should be used instead of signals, as the overhead of signals is dramatic, caused by the context switch that has to be done.

Semaphores can be used to solve to solve producer consumer problem (i.e. race condition).

Filling is used when we have to work on big data.

Unlike other methodologies of inter process communications, pipes are use where one-way communication is required for example in implementation of producer consumer scenario. In which one process is producer whose duty is just to produce and other is consumer which is just consuming. It requires one-way communication.