

## Trabajo Práctico de Laboratorio N°1: IPCs y Threads

### 1. Message Queues:

A. Generar 2 programas que accedan a la misma Message Queue (y la creen en caso de que la misma no exista). Uno de los programas deberá pedir un texto por teclado y se lo enviará al otro, que lo imprimirá por pantalla. Los programas finalizan cuando se recibe el texto FIN. Cuando el programa que escribe en la Message Queue finalice su ejecución, se debe liberar el recurso.

B. Realizar un programa que lea de un archivo (cuyo path se recibe por línea de comandos) los valores de 3 sensores. Los mismos se encuentran grabados en el archivo con el siguiente formato:

```
struct datos {  
    char tipo_sensor;  
    long data;  
}
```

En donde los tipos de sensor pueden ser:

- a. Tipo 1: Sensor de temperatura
- b. Tipo 2: Sensor de presión
- c. Tipo 3: Sensor de humedad

Por cada dato leído, se deberá mandar por una Message Queue el dato del sensor, con el campo mtype que se corresponda con el tipo de sensor al cual se hace referencia.

Realizar luego 3 programas similares (LectorTemperatura, LectorPresion, LectorHumedad) que lean los datos de la Message Queue que les correspondan (solo los que tengan el campo mtype igual al tipo de sensor del programa) y los impriman por pantalla.

### 2. Threads:

A. Dados 2 archivos de datos numéricos flotantes, hacer un programa que abra dichos archivos y genere 2 threads, enviándole a cada uno el file descriptor del archivo correspondiente. El primero deberá hacer el producto de todos los números del archivo, y el segundo la sumatoria de los mismos, imprimiendo el resultado en dos variables globales compartidas por todos los threads. El thread principal deberá tomar el resultado de estas dos operaciones y realizará el cociente de los mismos, imprimiendo finalmente el resultado por pantalla.

B. Ampliar el programa anterior, utilizando semáforos para acceder a las variables globales que se comparten entre los threads.

### **3. Shared Memory y semáforos:**

A. Realizar un programa que reciba un texto ingresado por consola y lo cargue en una Shared Memory.

B. Realizar 2 programas que lean la shared memory anterior (Señalizados por semáforos para asegurarse que no se está leyendo la información cuando todavía no se terminó de cargar). Uno de ellos deberá indicar la cantidad de palabras que contiene el texto, y el otro indicará la cantidad de letras del mismo.