INTRODUCCIÓN A LA CONCURRENCIA

TRABAJO PRÁCTICO N° 1

“Posix Threads”

Software a utilizar

Para realizar esta práctica utilizará el sistema operativo Linux.

En caso de no disponer del sistema operativo Linux, puede utilizar una máquina virtual, la cual será proveída por la cátedra y tendrá todo el software necesario instalado.

Tenga en cuenta que algunas de las instrucciones son específicas para sistemas Linux basados en Debian (Ubuntu, Kubuntu, Mint, etc.).

Una vez que cuente con el sistema operativo Linux funcional, deberá contar con el siguiente software (tenga en cuenta que existe más de una alternativa, seleccione la que más se ajuste a su comodidad o entorno).

Terminal

Se puede usar cualquier terminal (línea de comandos) como por ejemplo: xterm, konsole, gnome-terminal, xfce4-terminal, etc.

Editor de texto

Se necesita un editor de texto para realizar los prácticos. Desde la cátedra recomendamos Visual Studio Code, pero se pueden usar el que deseen.

Herramientas de compilación básicas para linux y páginas de manual

```
usuario@host $ sudo apt-get install build-essential
usuario@host $ sudo apt-get install manpages-es
```

Debugger

Permite depurar programas. gdb es el de línea de comandos, existiendo algunas interfaces gráficas como ddd, kdbg, nemiver y algunas integraciones a IDEs como Eclipse o CodeBlcks.

Compilación y ejecución de un programa C

Para compilar un programa escrito en C se debe usar el comando gcc:

usuario@host $ gcc ejemplo.c -o ejemplo

Luego, para ejecutarlo, se debe anteponer la ruta. A diferencia de Windows, no se ejecutan los programas que se encuentran en el directorio actual:

usuario@host $ ./ejemplo

Compilación con Hilos

Los programas que utilicen pthread deberán ser compilados con la bandera -pthread, por ejemplo:

usuario@host $ gcc ejemplo.c -o ejemplo -pthread

Entrega de trabajos

Los ejercicios se pueden realizar en la terminal junto con algún editor de texto. Se deben entregar solamente los archivos fuentes (extensión ".c") junto con su Makefile correspondiente. Estos deben evidenciar la solución de cada ejercicio. Cada punto deberá estar antecedido por una celda indicando el enunciado y comentarios del alumno, ejemplo:

Ejercicio 1
Crear programa C que imprima hola mundo

Código de la solución
…

Ejercicios

1. Escriba un programa llamado ej1.c que tenga un único contador y tres hilos para modificar su valor. Uno de esos hilos, incrementa el contador en 2 en cada iteración y muestra su valor en pantalla, mientras que los dos hilos restantes, lo decrementa en 1 y muestra su valor en pantalla. Ejemplo "Hilo 1 incremento el contador al valor de 25".

2. Cree un programa llamado ej2.c que defina un arreglo de enteros. Genere un hilo que inicialice el arreglo con valores aleatorios. Al finalizar, el proceso principal debe mostrar por consola los valores del arreglo de forma secuencial. El tamaño del arreglo será suministrado por medio de un argumento al programa.

3. Escriba un programa llamado ej3.c que genere N cantidad de hilos, siendo N una constante (al menos mayor a 10). Cree un arreglo de enteros de N posiciones, el cual se va a inicializar con valores aleatorios. El proceso principal dispone de una variable global. Cada hilo tiene que sumarle a esa variable global, el valor correspondiente de su posición en el arreglo. Por ejemplo si se está ejecutando el hilo número tres, usara el valor del arreglo que este en la posición 3. Tenga en cuenta que los arreglos se inician en la posición cero. El proceso debe esperar la finalización de todos los hilos y mostrar el valor resultante en la variable global.

4. Cree un programa llamado ej4.c el cual genere dos hilos. Uno será el hilo productor y el otro será el consumidor. EL hilo productor debe generar caracteres aleatorios y

guardarlos en un buffer. El hilo consumidor tiene que leer los valores del buffer y mostrarlos por consola. Cuando se habla de buffer, se puede representar con un arreglo. Como detalle a tener en cuenta:

a. Cuando el productor quiere generar un carácter y el buffer está lleno.

b. Cuando el consumidor quiere leer un carácter del buffer y el mismo está vacío.

5. Cree la versión 3 del programa mm.c (multiplicación de matrices), basado en la versión 2 disponible en el Aula, pero mejorado. Esta versión debe repartir el trabajo en H hilos. H debería dividir exactamente a N, siendo N el tamaño de la matriz. El valor H se pasará como parámetro en la invocación del programa. Utilice un tamaño de matriz de 1024x1024 o mayor, si la memoria disponible lo permite.

   Para procesar una matriz con un tamaño mayor a 381 en Linux con 32 bits, es necesario correr el siguiente comando:

   ```
   usuario@host $ ulimit -s 1024
   ```

   Realice pruebas de tiempos con diferentes valores de H. Un ejemplo de invocación podría ser:

   ```
   usuario@host $  ./mm 2
   ```

   En este caso se repartirá el trabajo en 2 hilos. Al primero le tocarán las filas 0 a 511 de la matriz A (y de la matriz resultante, C). Al segundo le tocarán las filas 512 a 1023.

   Otro ejemplo:

   ```
   usuario@host $  ./mm 512
   ```

   En este caso se crean 512 hilos. A cada uno le tocan dos filas.