
Trabajo Práctico 1 - Posix Threads

Software a utilizar

Para realizar esta práctica utilizará el sistema operativo Linux. Puede utilizar las PCs que se encuentran en el laboratorio, utilizando un penndrive para extraer su progreso, o su notebook/netbook personal.

En caso de utilizar su computadora personal y no disponer del sistema operativo Linux, puede utilizar una máquina virtual, la cátedra provee una máquina basada en Xubuntu (32bits).

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

- Jupyter Notebook Se instala con el comando:

```
sudo apt-get install python-setuptools
sudo easy_install pip && sudo pip install jupyter
```

Luego, se invoca de la siguiente manera:

```
jupyter-notebook
```

- Terminal

Puede bien utilizar la integrada a Jupyter o cualquier terminal como la xterm, konsole, gnome-terminal, xfce4-termial, etc.

- Vim, Geany, Kate, Sublime Text, etc.

Editores de texto con resaltado de sintaxis. Jupyter provee funcionalidad básica para edición de textos.

- Herramientas de compilación básicas para linux y páginas de manual

```
sudo apt-get install build-essential
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
```

Complicación con Hilos

Los programas que utilicen `pthread` deberán ser compilados con la bandera `-pthread`.

Ej: `gcc -pthread ejemplo.c -o ejemplo`

1. Ejercicios

Nota Si bien los ejercicios se pueden realizar en la terminal, utilizaremos Jupyter Notebooks para la entrega de los trabajos. Estos deben evidenciar la solución. Cada punto deberá estar antecedido por una celda indicando el enunciado y comentarios del alumno. Ej:

Ejercicio 1

Crear programa C que genere imprima hola mundo

- 1) Cree un directorio con el comando `mkdir TP-1`. Acceda a dicha carpeta con el comando `%cd TP-1` y cree un archivo llamado `README` con su nombre. Ejecute `%writefile README.md` y agregue su nombre.

Apartir de ahora todos los ejercicios deberán estar contenidos en la carpeta `TP-1`.

Cambie el nombre del notebook

- 2) Cree un programa `ej2.c` (con `%writefile`) que recorra un arreglo de enteros inicializados aleatoriamente de manera secuencial. El tamaño del arreglo se definirá con una constante.
- 3) Modifique el programa para que se creen 2 hilos. Realice las modificaciones para que el programa termine cuando todos los hilos hallan completado el recorrido.

Puede ayudarse de la utilidad contar palabras `wc -l` para verificar que la salida es correcta.

- 4) Cree un programa `ej4.c` que utilice con 2 hilos dónde uno de ellos genere caracteres aleatorios y el otro hilo los muestre en pantalla.
¿Como evita que se repita el mismo caracter?
- 5) Cree `ej5.c`, basado en `ej4.c`, para que ahora el hilo productor pueda generar más de un elemento antes de quedar bloqueado. Haga que el consumidor pueda consumir tantos elementos.