

CI-3825

Enero-Marzo 2017

Proyecto 2

Objetivos

- Aprender a manejar concurrencia sencilla usando threads (POSIX).
- Aprender a manejar las llamadas para acceder al sistema de archivo

Problema

Es de gran utilidad en redes UNIX, donde los usuarios tienen asignadas cuotas de disco, contar con una aplicación que recorra recursivamente un árbol de directorios, generando un reporte con la información más importante para cada directorio en el árbol. De esta manera el usuario podrá saber con exactitud en qué directorios está invirtiendo mayor cantidad de espacio.

En esta asignación se pide desarrollar una aplicación llamada `tls` (thread ls) que debe cumplir con las siguientes características usando hilos

Sintaxis

```
tls [-h] | [-n i] [-d directorio] [ salida ]
```

-h	muestra por pantalla un mensaje de ayuda (sintaxis, descripción de parámetros, etc.) y termina.
-n i	nivel de concurrencia solicitado. Por defecto crea un solo hilo trabajador
-d directorio	especifica un directorio desde donde revisa el espacio utilizado. Por defecto hace el cálculo desde el directorio actual
salida	archivo que contendrá la lista de directorios y el espacio en bytes ocupado por los archivos regulares. Valor por defecto es la salida estándar.

Recuerde la semántica que tiene la especificación de sintaxis de comandos:

- [] Indica que lo que esté entre esos símbolos es opcional. Se debe especificar un valor por defecto.
- A | B indica que se espera que el comando se puede ejecutar ya sea con A o con B, pero no los dos al mismo tiempo. A y B puede ser una opción o secuencia de opciones como en `tls`.
- n i indica que cuando se vea en la lista de parámetros -n el siguiente debe ser interpretado como i. Esto se usa para no fijar el orden en que se pasan los parámetros al programa.

En resumen, esto implica que las siguientes operaciones son invocaciones validas:

```
./tls -h
./tls salida
./tls -n 4 salida
./tls -d /home/adiserio salida
./tls -n 6 -d /home/adiserio salida
./tls -d /tmp -n 8 salida
./tls
./tls -n 4
./tls -d prueba
./tls -n 6 -d otrodirectorio
./tls -d /tmp -n 8
```

- El hilo principal o *maestro* crea el número de hilos *trabajadores* que se especifica en la línea de comandos a través de la opción “-n i”
- El hilo maestro explora el directorio pasado en la línea de comando y para cada archivo *directorio* encontrado agrega el nombre del mismo a una lista de directorios a ser explorados.
- Un hilo trabajador disponible se encargará de revisar el siguiente directorio que se encuentra en la lista de directorios por revisar. Este hilo revisa el directorio asignado y devuelve el número de archivos regulares encontrados y la cantidad de bytes que ocupan. Si en la revisión aparecen archivos tipo directorio, estos serán incluidos en la lista de directorios por revisar. Una vez que finalice su trabajo en este directorio queda disponible para trabajar sobre el siguiente directorio.

- Cuando se terminan de explorar todos los directorios, el hilo maestro se encarga de imprimir la información relacionada con cada directorio encontrado en la jerarquía de directorios

La salida debe indicar:

- identificación del hilo que realizó el trabajo
- directorio examinado (camino completo)
- número de archivos regulares encontrados en el directorio
- cantidad de bytes ocupados por los archivos regulares en dicho directorio.

Aspectos importantes a tener en cuenta para la implementación

- Tener en cuenta que en el recorrido se pueden encontrar directorios sin permiso de lectura o ejecución.
- Verificar que el directorio donde se desee crear el archivo reporte puede no tener permisos de escritura, en este caso dar un mensaje de error y abortar el proceso.
- Recuerden tomar en cuenta aspectos de modularidad, eficiencia y buen uso de la memoria.

Entrega

El proyecto puede ser entregado el jueves de la semana 11 (23:55). La entrega consistirá de un archivo comprimido en formato *tar.gz* con los archivos fuente de su programa (.c y .h), makefile y README. No incluya el ejecutable ni los archivos objeto generados durante la compilación.