Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-3825 - Sistemas de Operación I
Trimestre Septiembre-Diciembre de 2016

Proyecto II: Uso del Disco (10 %)

Objetivos Generales

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

Definición del problema

Hacer un programa llamado **UsoDisco**, que calcule de forma concurrente el espacio en disco utilizado por los archivos regulares a partir de un directorio particular, usando hilos.
La salida debe contener por cada directorio examinado, el camino completo a partir del directorio y la cantidad de bloques ocupado por los archivos regulares en dicho directorio.

Adicionalmente el programa deberá escribir por la salida estándar (pantalla) un reporte que indique: el número de threads creados, número de directorios examinados y el número de archivos contabilizados.

Sintaxis

A partir de sus códigos fuentes se debe generar un archivo ejecutable llamado `UsoDisco`. El comando para su ejecución es como sigue:

```
>./UsoDisco  [-h] | [-n i] [-d directorio] [ salida ]
```

donde:

`-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 thread trabajador
`-d directorio`: especifica un directorio desde donde calcula el espacio utilizado. Por defecto hace el cálculo desde el directorio actual.

`salida`: archivo que contendrá la lista de directorios y el espacio en bloques ocupado por los archivos regulares. Valor por defecto es la salida estándar

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

`[ ]`: Indica que lo que este 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 UsoDisco.

`-x y`: indica que cuando se vea en la lista de parámetros `-x` el siguiente debe ser interpretado como `y`. 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 válidas:

- ./UsoDisco -h
- ./UsoDisco salida
- ./UsoDisco -n 4 salida
- ./UsoDisco -d dir salida
- ./UsoDisco -n 6 -d dir salida
- ./UsoDisco -d dir -n 8 salida
- ./UsoDisco
- ./UsoDisco -n 4
- ./UsoDisco -d dir
- ./UsoDisco -n 6 -d dir
- ./UsoDisco -d dir -n 8

Las siguientes son ejemplos de invocaciones no son válidas (puede haber más):

- ./usoDisco -h : el comando se llama UsoDisco
- ./UsoDisco -n salida.txt : falta el nivel de concurrencia
- ./UsoDisco -d 4 salida.txt : a menos que haya un directorio con nombre 4
- ./UsoDisco -h salida.txt : -h no puede ser usado con las otras opciones.

Ejemplo de archivo de salida

```
240    ./UsingMPI2/starting
 20    ./UsingMPI2/moreio/f90
120    ./UsingMPI2/moreio
 12    ./UsingMPI2/remotemem/f90
 88    ./UsingMPI2/remotemem2
 44    ./UsingMPI2/other
 12    ./UsingMPI2/spawn/f90
```

Requerimientos de la implementación

El programa debe crear un número (especificado con la opción "-n i") de hilos (trabajadores), que realizarán el cálculo del espacio a partir del directorio pasado por la línea de comando.

- El hilo principal (llamado maestro) funciona de la siguiente forma: Explora el primer nivel del directorio pasado en línea de comando; para cada archivo directorio encontrado, agrega el nombre a la lista de archivos a ser explorados.
- De los hilos trabajadores recibe la sumatoria de bloques de los archivos regulares encontrados en el directorio explorado por el hilo y una lista con los nombres de directorios que encontró en esa revisión. Estos nombres serán agregados a la lista del hilo maestro que contiene los directorios por explorar.
- El hilo maestro le asigna un directorio, de la lista de archivos a explorar, a un hilo libre, elimina el nombre del directorio de la lista y marca a dicho hilo como ocupado.
- Cuando se terminan de explorar todos los directorios, se imprime la respuesta, y terminan todos los hilos.

Los hilos trabajadores libres deben:

- Recibir del maestro el nombre del directorio a explorar.
- Calcular el espacio usado en disco por los archivos regulares y producir la respuesta.
- Al terminar, pasar la respuesta al maestro y cambiar su estado a libre.
- Esperar por un nuevo directorio a explorar.

Recomendaciones

1. Se debe chequear el valor de retorno de las llamadas al sistema.
2. Deben hacer un programa modular, legible y documentado.
3. Los programas deben poder compilarse y ejecutarse en cualquiera de las computadoras (GNU/Linux) del LDC. Si Ud. realizó el programa en su casa o en la USB, pero en alguna otra plataforma, debe asegurarse antes de la entrega que su proyecto funciona en las estaciones GNU/Linux antes mencionadas.
4. Se debe asegurar que su programa no tiene pérdidas de memoria. Para ello se sugiere el uso de **Valgrind** [1].

[1] `http://valgrind.org/`

Condiciones de la entrega

La entrega del proyecto será el día **18 de Noviembre de 2016**. La entrega tiene las siguientes condiciones:

1. Debe crear un archivo llamado `Proyecto2-X-Y.tar.xz`, donde `X` y `Y` son los número de carné de los integrantes del grupo, con todos los códigos fuentes (archivos .c y .h) y el Makefile, el cual debe ser entregado en la página del curso en el Aula Virtual, el día viernes antes de las 1:00 pm.
2. Proyecto que no pueda compilarse usando las reglas dadas en el archivo Makefile, tiene CERO como nota.
3. El programa debe hacer uso eficiente de los recursos que administra el sistema operativo: memoria, tiempo, etc.
4. Debe respetar las especificaciones que se le dan en el enunciado.
5. Debe entregar a la hora del taller, la **Declaración de autenticidad de entrega**, debidamente firmada por los integrantes del equipo.