

Laboratorio de Sistemas de Operación I

Diciembre 2014-Marzo 2015

Proyecto II

1. Objetivo

Familiarización con las llamadas al sistema para manejar comunicación entre procesos a través de pipes y señales y para acceder al sistema de archivos.

2. Introducción al 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 usando procesos. Además debe identificar cuántos enlaces lógicos existen.

Sintaxis

```
UsoDisco [-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 proceso 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 y el nro total de enlaces lógicos encontrados. A continuación se muestra un ejemplo 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
```

Total de enlaces lógicos: 3

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 `UsoDisco`.
- `-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:

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

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

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

El programa debe crear un número (especificado con la opción `--n i`) de procesos (*trabajadores*), que realizarán el cálculo del espacio a partir del directorio pasado por la línea de comando. El proceso principal (llamado *maestro*) funciona de la siguiente forma:

- Explora el directorio pasado en línea de comando; para cada archivo *directorio*, pasa el nombre de dicho directorio, a través de un pipe a un proceso trabajador para que haga el cálculo.
- De los procesos trabajadores recibe el tamaño total en bloques de los archivos regulares encontrados en el directorio explorado por el proceso trabajador, el número total de enlaces lógicos encontrados y una lista con los nombres de directorios que encontró en esa revisión. Estos nombres serán agregados a una lista del proceso maestro que contiene los directorios por explorar.
- Cuando un proceso trabajador termina pasa a un estado libre.
- El proceso maestro le asigna un directorio a un proceso libre y marca a dicho proceso como ocupado.
- Cuando se terminan de explorar todos los directorios, se imprime la respuesta, y se terminan todos los procesos (esta información de que terminen puede ser manejada con una señal)

Los procesos trabajadores libres deben:

- Recibir del maestro el nombre del directorio a explorar
- Calcular el espacio total usado en disco por los archivos regulares, contar los enlaces lógicos encontrados y producir la respuesta
- Al terminar, pasar la respuesta al maestro a través de un pipe
- Esperar por un nuevo directorio a explorar, e iterar

La salida debe indicar: directorio examinado (camino completo) y la cantidad de bloques ocupado por los archivos regulares en dicho directorio, y al final indicar el número total de enlaces lógicos encontrados por todos los procesos trabajadores.

¿Qué debe entregar y cómo debe hacerlo?

La entrega del proyecto se hará vía aula virtual el jueves de semana 12 y debe consistir de un archivo `tar.gz` que contenga: los programas fuente, el `makefile`, un archivo de texto `LEEME.txt` que indique cómo compilar, ejecutar el programa al estilo de la información que da el “`man`”.

Yudith Cardinale Villarreal