PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2333 — Sistemas Operativos y Redes — 1/2017

Tarea 3

Viernes 05-Mayo-2017

Fecha de Entrega: Domingo 21-Mayo de 2017, 23:59
Composición: grupos de n personas, donde $n \leq 2$

Descripción

El objetivo de esta tarea es **simular** el funcionamiento de dos sistemas de archivos: FAT *(File Allocation Table)* y un sistema *combined multilevel index*[1] usando bloques de un disco simulado.

La simulación consiste en el manejo de los aspectos administrativos (*metadata*) para mantener un sistema de archivos funcional. **No se crearán contenidos en bloques de datos**, sino que solamente se crearán archivos que contengan información administrativa de cada sistema de archivos simulado.

La simulación puede iniciarse con un disco existente como base, o bien crear un disco inicialmente vacío. Una vez que la simulación termina, los datos que hay en el disco simulado deben ser exportados a un archivo en el sistema de archivos real. Este archivo exportado puede ser usado en otra ejecución de la simulación.

Formato y modelamiento del disco

El disco será modelado como una colección de bloques de 4KB. El disco a simular deberá ser de tamaño 4GB. Para administrar un disco con estas características se requieren $4\text{GB}/4\text{KB} = 2^{20}$ punteros a bloques de disco. Cada puntero a un bloque de disco ocupa 4B (32 bit). Se debe, entonces, utilizar un archivo binario de $2^{20} \times 4\text{B} = 4\text{MB}$ para implementar el sistema de archivos deseado (FAT, o *multilevel indexed*). Dentro de cada puntero de 4B, se debe usar 3B para almacenar una referencia, y 1B para *metadata*. Los 3B para referencia se pueden usar para referenciar: (1) a otro bloque del disco; (2) a un indicador EOF (`0xFFFFFF`); (3) a una posición en un archivo en el sistema de archivos real, donde este archivo contiene la información de un directorio en formato de texto; (4) a un archivo en el sistema de archivos real que contenga datos de bloques índice (en formato binario).

El Byte para *metadata* del bloque debe seguir el siguiente formato. Los campos que no son usados quedarán siempre en 0.

bit	contenido
0,1	unused
2	bloque libre
3	directorio
4	contenido
5	*index block*
6	bloque intermedio, nivel 1
7	bloque intermedio, nivel 2

Almacenamiento (exportación) del disco

Mientras se ejecuta la simulación, se puede mantener en memoria todas las estructuras que considere necesarias para manipular el sistema de archivos.

Sólo al momento de terminar la simulación, el contenido del disco simulado debe ser exportado a un archivo binario en el sistema de archivos real. Este archivo generado deberá tener el nombre `simdisk.fat`, o `simdisk.mli` de

[1] Como se menciona en el libro de Silberschatz

acuerdo al sistema de archivos que se ha simulado. Un archivo bien exportado podría ser leído (importado) por otro implementación del simulador que utilice **el mismo** sistema de archivos.

Directorio

En cada caso se requiere de un archivo de texto de nombre `directorio.txt`. Este archivo contendrá las tuplas ⟨ *ruta del archivo*, *primer bloque (FAT)* ó *bloque índice (MLI)*, *tamaño en byte* ⟩.

Este directorio sirve para ubicar la *metadata* de los archivos dentro de `simdisk.*`.

Acceso y uso de archivos externos

En el caso de *multilevel index*, el campo de referencias (3B) debe apuntar a un *bloque índice* (*inodo*). El *bloque índice* debe contener la información de *metadata* del archivo, la cantidad de punteros directos, el puntero de indirección simple, y el puntero de indirección doble.

Al momento de exportar el disco, el contenido de cada *bloque índice*, debe almacenarse en algún lugar. Para esto se debe utilizar un archivo binario externo en el sistema de archivos real (de 4KB). La referencia (3B) debería almacenar la ruta de este archivo, pero esto no cabría en 3B. Por lo tanto, se utilizará la referencia para almacenar una posición en un archivo de texto llamado `accesos.txt` que contendrá las rutas de los archivos (de 4KB) que contienen información de los bloques índice de cada archivo. La ruta usada debe ser *relativa* respecto a la ubicación donde se ejecuta la simulación.

De esta manera, el sistema de archivos simulado consiste de: (1) el archivo `simdisk.*`, (2) un archivo de directorio, (3) un archivo `accesos.txt` con las rutas de los archivos en el sistema de archivos real que contienen los datos de los bloques índice, y (4) el conjunto de archivos indicado en `accesos.txt` que deben encontrarse en el mismo directorio, o en un subdirectorio de aquél en que se encuentra `simdisk.*`. Los ítemes (3) y (4) no se usan para la versión FAT.

Los archivos externos deben ser almacenados replicando la estructura de directorio especificado en el sistema de archivos simulado. Al momento de exportar el archivo `simdisk.*`, la estructura de directorio de los archivos externos debe reflejar la estructura de directorio del sistema de archivos simulado. Esto significa que puede necesitar borrar o crear archivos en el sistema de archivos real. Los nombre de este archivos deben ser creados de manera consistente para poder leerlos en otra ejecución de la simulación.

Manejo del espacio libre

Para considerar el espacio libre, es decir, aquellos bloques donde el bit correspondiente es 0, se debe utilizar un bitmap el cual puede ser exportado como un archivo .bmp en blanco y negro.

Acciones de la simulación

El sistema de archivos simulado debe ser capaz de leer instrucciones de un archivo (en el sistema de archivos real) llamado `acciones.txt`. Estas acciones pueden ser:

- `cd [ruta relativa a directorio donde reubicarse]`
- `mkdir [ruta relativa]/[nombre para el nuevo directorio]`
- `mkfile [ruta relativa]/[nombre para el nuevo archivo]`
- `mv [actual ruta relativa del archivo o dir] [nueva ruta relativa]`
- `rm [archivo/directorio]`

En caso que se proporcione una ruta o acción inválida, el simulador debe detectar el error y continuar su ejecución. Puede intentar corregir la acción, o bien ignorar el error, pero no es aceptable que el disco quede en un estado inconsistente, o que el simulador termine.

Para simular una escritura se pueden usar dos acciones que permiten hacer crecer o reducir el tamaño de un archivo

- `ad [archivo] [tamaño a agregar en byte] [posicion en el archivo donde agregar (por defecto al final)]`
- `rd [archivo] [tamaño a remover en byte] [posicion en el archivo desde donde remover (por defecto a distancia suficiente para remover hasta el final. Si la cantidad supera el tamaño del archivo, el archivo queda con tamaño 0 (no se borra)]`

Puede generar un archivo `acciones.txt` de manera aletoria para efectos de probar su sistema de archivos.

Ejecución de la simulación

El simulador debe ser llamado con la instrucción:

```
./disk_simulator <steps> [-(s)(a(u))(d)] <seed> <actions> <disk>
```

- `<steps>` corresponde al número de acciones que deben ser simuladas en el disco
- Las opciones `sad` indican si existe alguno de los parámetros a continuación y su orden. Si no está presente, no se asume su existencia (maneje los posibles errores de tipeo)
- `<seed>` corresponde a la semilla (opcional) que se puede usar si se quiere generar acciones de manera aleatoria (puede ser útil para debuggear)
- `<actions>` corresponde a una lista de acciones a ser simuladas
- `<disk>` corresponde a un directorio que contiene un disco ya simulado sobre el que se debe ejecutar la simulación

Su programa debe atenerse a las diversas consideraciones mencionadas en secciones anterior con el fin de poder recibir como *input* no sólo el *output* de si mismo, si no que las de otras implementaciones.

El simulador debe ser suficientemente robusto como para detectar cuando el *input*, ya sea el disco o las acciones, no cumple con el formato correcto, indicando mensajes de error apropiados. Si una acción no puede ser ejecutado debido a un error, el error debe ser detectado y el simulador debe continuar con las siguientes acciones.

README

Deberá incluir un archivo `README` que indique quiénes son los autores de la tarea, a grandes rasgos cuáles fueron las decisiones de diseño para hacer el programa, qué supuestos adicionales ocuparon, entre otras cosas que considere necesarias para una mejor corrección de su tarea. Se sugiere utilizar formato **markdown**.

Formalidades

La tarea será entregada mediante `git` en un repositorio **privado** que ustedes deberán crear. Se revisará el contenido de la rama `master` al día domingo 21 de mayo de 2017, 23:59.

- Puede ser realizada en forma individual, o en grupos de 2 personas.
- Su tarea deberá compilar utilizando el comando `make` en la raíz de su repositorio, y generar un ejecutable llamado `disk_simulator` en esa misma carpeta.
- Su repositorio **DEBE ser privado**, de lo contrario calificará como copia (alguien les podría haber copiado). Esta restricción es fundamental. Si no la cumple **no** se revisará su tarea.

- La entrega será automatizada, basta que **registren** su grupo y repositorio mediante un formulario que habilitaremos para ello (no por email).
- Como el repositorio debe ser privado, tendrá que permitir acceso especial al curso, autorizando al servidor de tareas acceder al contenido. Para esto basta registrar la llave pública del curso en las **Deployment Keys** de su repositorio.
- Lo mejor es que use Bitbucket.

Evaluación

- 10 %. Formalidades. Esto incluye cumplir las normas de la sección formalidades.
- 30 %. Implementación FAT
  - 10 %. Formato del disco (estructuras en memoria)
  - 10 %. Exportación de `simdisk.fat`
  - 10 %. Implementación de acciones
- 40 %. Implementación MLI
  - 15 % Formato del disco (estructuras en memoria)
  - 15 % Exportación de `simdisk.mli`
  - 10 % Implementación de acciones
- 10 %. Funcionamiento línea de comandos (`disk_simulator`)
- 5 %. Interacción correcta con otro disco simulado
- 5 %. Mensajes al usuario. Debe entregar mensajes claros y precisos y que permitan entender lo que está pasando, idealmente que no ocupen más de una línea. Además debe capturar el término forzado.

El no respeto de las formalidades o un código extremadamente desordenado podría originar descuentos adicionales.

Preguntas

A través del foro de EdX.