



Sistemas Operativos

Semestre 2017-2

Prof. Federico Meza

Proyecto de Programación de la Unidad#4

Entrega: Miércoles 13 de diciembre de 2017, hasta las 12:00 horas.

Descripción

En este proyecto Usted debe implementar un sistema de archivos para un sistema operativo mono-proceso. El disco será emulado con un archivo convencional del sistema operativo anfitrión —sea éste Linux o Windows— y contendrá 512 sectores de 512 bytes cada uno. El directorio tendrá un único nivel, y estará limitado a la cantidad de entradas que caben en un sector del disco. Además, supondremos que los nombres de archivos no superan los 8 caracteres.

Su diseño debe ser por capas o niveles. La capa inferior está encargada de interactuar con el disco, leyendo y escribiendo sectores físicos. Usted debería implementar una clase de objetos **Disco**, que ofrece esta funcionalidad, así como una clase utilitaria **Sector**, que modela un sector del disco. En un nivel superior Usted debe administrar los *file control blocks* que le permitan almacenar, reconstruir y operar sobre los archivos. Por simplicidad puede suponer que un FCB ocupa un sector en disco.

Es requisito de este proyecto administrar el espacio asignado a los archivos utilizando una FAT, cuyos puntos de ingreso deberán encontrarse en el FCB de cada archivo. En un nivel superior debe manejar el directorio, que es de un único nivel.

El sistema de archivos debe implementar las siguientes operaciones:

Format: Para crear un disco y su sistema de archivos asociado.

Create: Para crear un archivo, con su correspondiente entrada en el directorio; además del nombre, se debe especificar el tamaño en bytes con que debe crearse el archivo.

Remove: Para borrar un archivo, eliminando su entrada del directorio y liberando el espacio en disco que ocupaba.

ReadAt: Para leer de una determinada cantidad de bytes de archivo, a partir de una posición lógica (*offset*).

WriteAt: Para escribir en un archivo una determinada cantidad de bytes, a partir de una posición lógica (*offset*).

PrintFile: Para mostrar en la pantalla el contenido de un archivo.

List: Para mostrar en pantalla las entradas del directorio, incluyendo el tamaño en bytes de cada archivo.

Por simplicidad puede suponer que los archivos no crecen una vez que son creados. Tan sólo es posible sobrescribir su contenido.

Tome en cuenta que varias de las estructuras de datos del sistema de archivos deben almacenarse tanto en memoria como en disco. Implemente operaciones para recuperar y volver a guardar estas estructuras. La representación en memoria y en el disco no tiene por qué ser idéntica.

No olvide que debe gestionar el espacio libre en el dispositivo.

Programa de prueba

Escriba un programa de prueba que permita verificar el correcto funcionamiento de su sistema de archivos. El programa debe probar todas las funcionalidades solicitadas.

Forma de trabajo, entrega y evaluación

El trabajo debe desarrollarse en equipos de 2 estudiantes. Ambos integrantes deben poder responder preguntas sobre la totalidad de la implementación de su proyecto.

La evaluación tomará en cuenta el cumplimiento de los requisitos establecidos (80 %) y un informe escrito, en formato PDF (20 %). El informe debe ser completo pero preciso. Debe contener una descripción (de preferencia con figuras) de las decisiones de diseño que se tomaron, y de las características del sistema de archivos resultante.

La entrega debe hacerse en un único archivo `.zip` o `.tar.gz`, que contendrá el informe y el código.

Pauta de evaluación

Informe (20 %)	<p>Créditos (100 puntos):</p> <ul style="list-style-type: none">▪ Forma, ortografía, capacidad de síntesis (30 puntos)▪ Descripción del diseño y las características (70 puntos) <p>Descuentos potenciales:</p> <ul style="list-style-type: none">▪ Falta análisis de limitaciones, en caso de existir (30 puntos)▪ No cumplimiento de las normas de entrega (30 puntos)
Programa (80 %)	<p>Créditos (100 puntos):</p> <ul style="list-style-type: none">▪ Diseño del sistema de archivos y sus estructuras de datos, incluidos el directorio, la FAT y el manejo del espacio libre (20 puntos)▪ Implementación de las operaciones solicitadas sobre las estructuras diseñadas (40 puntos)▪ Correcta interacción memoria–disco para las estructuras que lo requieren (20 puntos)▪ Programa de prueba (20 puntos) <p>Descuentos:</p> <ul style="list-style-type: none">▪ Malas prácticas de programación (20 puntos)▪ Ausencia de comentarios (20 puntos)▪ Falta de separación entre directorio y estructura de asignación de espacio (30 puntos) <p>Cotas de puntaje:</p> <ul style="list-style-type: none">▪ Errores de compilación (Puntaje máximo: 30 % del puntaje correspondiente)▪ Errores de ejecución (Puntaje máximo: 30 % del puntaje correspondiente)▪ Resultados incorrectos (Puntaje máximo: 40 % del puntaje correspondiente)