

Final

Sistemas Operativos

El final debe entregarse a más tardar el día 7 de diciembre de 2017. Debe elegir uno de los dos problemas. Deben entregar los programas fuente de su solución con documentación de cómo compilar y ejecutar sus programas, describiendo cómo dar la entrada y qué salida se debe esperar. Esta documentación tiene un peso del 30 % de la calificación.

Problema 1 El problema consiste en programar en Java o en C una simulación de asignación de memoria usando particionamiento dinámico. La memoria es de 256KB y, para simplificar, se asignarán pedazos de 2KB, de tal manera que cada «proceso» que requiera memoria pedirá de 3 a 10 pedazos.

La simulación deberá consistir de 3 módulos: El módulo de memoria que es el que se encarga de la asignación y desasignación de memoria. Se deben programar 4 de éstos módulos que implementen cada una de las posibles políticas de asignación (*best-fit*, *worst-fit*, *first-fit*, *next-fit*).

El módulo que genere las peticiones de memoria las cuales leerá de un archivo que contendrá los siguientes datos: Una línea que contendrá el número n de procesos que se procesarán seguido de n líneas con la información de cada proceso. La información correspondiente a los procesos será de la siguiente manera, el tiempo de llegada, el tiempo de salida y la cantidad de memoria en pedazos de 2KB que usarán. Los procesos se identifican por su posición en el archivo.

Un módulo que reporte el estado de la memoria, esto es, qué pedazos hay libres y qué pedazos están ocupados.

Sugerencias: El módulo de memoria puede tener dos funciones. La primera llamada **asigna** que debe tomar como parámetros el número de proceso y la cantidad de pedazos de memoria a asignar. La segunda, llamada **devolver** que tome el número de proceso y devuelva la cantidad de memoria del proceso en cuestión.

El módulo de reporte debe tener una función que, al ser llamada, reporte qué pedazos de memoria están siendo usados y qué pedazos están libres.

Un ejemplo pequeño de archivo es el siguiente:

```
5
0 10 4
3 20 6
5 8 3
9 15 8
12 25 9
```

Este archivo indica que se procesarán 5 procesos. El primero llega en el momento 0, termina en el momento 10 y necesita 4 pedazos de memoria. El segundo llega en el momento 3, termina en el 20 y necesita 6 pedazos de memoria, etc.

Problema 2 El problema consiste en hacer una simulación de una FAT. El tamaño del disco manejado es de 512MB con bloques de tamaño 512KB. La FAT debe ser simulada con un arreglo de números enteros del tamaño adecuado para el disco en cuestión.

En los documentos anexos pueden encontrar la información pertinente a la FAT. En ellos se menciona el directorio «root», que es el único directorio que se requiere para guardar la información necesaria de los archivos en este caso específico.

La simulación consiste en leer los datos de los procesos que aparecen en un archivo de texto que estará formateado de la siguiente manera:

Una línea con un número natural n , seguida de n líneas con las peticiones de escritura o borrado (0 y 1 respectivamente). En el caso de que la operación sea de escritura, se dará el nombre del archivo (una cadena de caracteres) y el número de KB que se requieren para el archivo. En el caso de borrado, solo vendrá el nombre del archivo. En caso de que se haga escritura de un archivo que ya existe en el directorio, el número de bytes que aparezca en los datos debe añadirse al archivo ya existente.

La simulación debe mostrar la FAT cada cierto número de operaciones.

Un ejemplo de archivo de datos sería:

```
7
0 uno 2048
0 dos 5120
0 tres 51200
1 dos
0 cuatro 51200
0 cinco 20
1 uno
```