

Sistemas Operativos

ITBA

Cuatrimestre 2 - 2009

Trabajo Práctico 2: Multitasker

Objetivo

Realizar un programa que conmute diferentes procesos, asignándoles a cada uno un tiempo de ejecución. El multitasker no corre sobre ningún Sistema Operativo, se ubica en memoria utilizando un bootloader GRUB.

Enunciado

El trabajo consta de la realización de un Multitasker, cuyo objetivo es asignar tiempo de ejecución a diferentes procesos en memoria. El sistema deberá ser implementado para plataformas Intel de 32 bits, utilizando el procesador en modo protegido. El multitasker debe ser preemptivo, es decir, cualquier tarea puede ser desalojada del microprocesador. El encargado de administrar el CPU es el scheduler el cual tomará como base de tiempo la interrupción de hardware INT8 correspondiente al timer tick, para realizar la asignación de tiempo (time slot).

Contexto de tareas

Cada grupo deberá elegir la forma en resguardar el contexto de cada tarea, las opciones son: utilizar los TSS que provee el microprocesador Intel 386 una implementación propia de código.

Scheduler

El Multitasker deberá implementar 2 (dos) tipos de scheduling distintos. Al menos uno de ellos deberá considerar la prioridad de los procesos para asignar los slots de tiempo.

Estados de procesos

El sistema deberá estar programado de manera que se diferencien los estados básicos de "Corriendo", "Esperando" y "Listo". Por otra parte, cada proceso deberá tener un valor de prioridad entre 0 y 4, configurable al inicio de la ejecución que indique la importancia del proceso. Además se deberán demostrar el funcionamiento de los mismos con programas de prueba y se deberá poder corroborar el estado del proceso y el porcentaje de procesador que está ocupando

con la ayuda del comando "top". También deberá haber un comando kill que permita matar procesos en ejecución. Tener en cuenta que kill debe matar también a todos los hijos de ese proceso.

Intérprete de comandos

El usuario debe poder ejecutar las diferentes tareas a través de comandos ingresados por teclado. La forma de los comandos quedan a elección del alumno.

Procesos en background

El sistema debe tener la posibilidad de correr los mismos procesos tanto en foreground como en background. Para este ultimo se deberá utilizar el carcter "&" al igual que en UNIX

Administración de memoria

El sistema debe tener un módulo de administración de memoria mediante paginación para los procesos, el mismo se encargará de lo siguiente:

- Cada proceso tendrá su stack propio en una página, a la cual solamente él tendrá acceso. Cada proceso podrá leer y escribir libremente sobre esta página pero no páginas de otros procesos.
- Los procesos no poseerán un heap propio, ya que están corriendo sobre la misma zona de datos del SO.
- Ningún proceso deberá leer o escribir directamente ninguna variable global del SO. En caso de que haya variables globales que estén pensadas para ser leídas por procesos usuario, deberán tener una función que las copie a una zona de heap propia al proceso, simulando un system call.
-

Filesystem en Memoria

Los alumnos deberán implementar un filesystem no persistente. Al inicializar el sistema operativo, deberá ser creada la estructura en blanco. Desde allí el alumno podrá:

- Crear, renombrar y eliminar archivos de texto con algún comando, y visualizarlos con otro comando.
- Crear, dar nombre, renombrar y eliminar unidades lógicas de agrupación (ej: directorios).
- Desplazarse por estas unidades lógicas pudiendo explorar su contenido.
- Hacer una búsqueda de un archivo dentro de las agrupaciones lógicas, de manera recursiva o no, según el criterio de agrupación.

Naturalmente, al no haber persistencia de datos, los archivos y la estructura jerárquica se perderán al reiniciar la PC.

Programas de prueba

Cada grupo deberá desarrollar tareas, que funcionarán como programas de prueba, los mismos deberán demostrar vulnerabilidades y virtudes del trabajo cuales servirán para demostrar la implementación del TP.

Material a entregar

Cada grupo deberá entregar los fuentes, una imagen booteable con el bootloader y el multitasker. Además deberán entregar un informe impreso detallando las decisiones respecto a los items que quedaron a elección del grupo, problemas presentados y solución de los mismos durante la realización del trabajo

Integrantes

El trabajo puede hacerse en grupos de no más de tres integrantes. Se evaluará la funcionalidad de la aplicación así como el estilo del código y la calidad del informe entregado.

Consideraciones

Todo punto no explícito en este documento podrá ser interpretado a conveniencia del alumno, siempre dentro de los márgenes del sentido común. Ante la duda, consultar a los docentes o enviar un mail al mail de la cátedra.

Fecha de entrega

La fecha de entrega del trabajo práctico es el Jueves 29 de Octubre a las 16:00.