

Laboratorio 3: Scheduler

Sistemas Operativos - FaMAF - UNC

Original 2014: Carlos Bederián

Introducción

El planificador de xv6 usa el algoritmo más sencillo para distribuir tiempo de procesador entre los procesos, Round Robin. La implementación simplemente recorre la lista de procesos en orden y a cada proceso que está marcado como listo para correr le asigna un quantum de tiempo de procesador. Esta política es equitativa pues le asigna el mismo tiempo de procesador máximo a todos los procesos elegibles, pero tiene algunas propiedades que no son muy buenas. En particular, los procesos que realizan más entrada/salida que cómputo pasan su mayor parte del tiempo esperando, usando una porción muy pequeña de su quantum antes de bloquear esperando que las system calls de entrada/salida que realizan se completen. Un subconjunto muy importante de esta clase de aplicaciones es el de las aplicaciones que tienen interacción directa con el usuario, que esperan continuamente eventos en el teclado, mouse u otros periféricos. La percepción del usuario sobre la performance del sistema operativo depende en gran parte del rendimiento de estas aplicaciones interactivas, y por lo tanto es deseable que un sistema operativo pueda hacer que este tipo de aplicaciones tengan baja latencia.

Para lograr este objetivo, implementaremos en xv6 un nuevo planificador que le da mayor prioridad a las aplicaciones que realizan operaciones de entrada/salida. El algoritmo de planificación que usaremos es [MLFQ](#).

Enunciado

Se deberá:

1. Agregar un atributo prioridad a cada proceso, que tenga dos valores posibles. La prioridad de un proceso debería empezar siendo la más alta, y:
 - Descender cada vez que el proceso pasa todo un quantum realizando cómputo.
 - Ascender cada vez que el proceso bloquea antes de terminar su quantum.
1. Integrar a xv6 los programas **verduiops** y **frutaflops**, que realizan mediciones de performance poco precisas de operaciones de entrada/salida y poder de cómputo, respectivamente. Utilizarlos (ejecutando varias instancias de ambos concurrentemente) para ver la distribución de tiempo de procesador que realiza el scheduler original de xv6 y las versiones modificadas que realizarán en las consignas que siguen.

2. Modificar el scheduler de xv6 de modo que ejecute los procesos de mayor prioridad primero, y luego los de menor prioridad. Los procesos con la misma prioridad deben ejecutarse por orden de llegada. (i.e. FIFO)
3. Elaborar un informe sobre lo realizado.

El informe

El informe debe presentarse en un archivo `README.md` el directorio raíz en formato [markdown](#) e incluir:

- Una explicación de lo que hace el scheduler original de xv6 en cada transición de estado del proceso:
- Hacer que un proceso que duerme sea planificable.
- Ejecutar un proceso planificable.
- Replanificar un proceso que ha agotado su tiempo de procesador disponible.
- Bloquear un proceso que está esperando entrada/salida.
- Una explicación de lo que hace el nuevo scheduler para los mismos cambios de estado.

Ayudas

- Todas las modificaciones a realizar (salvo los extras) están confinadas a `proc.h` y `proc.c`.
- Comience analizando cómo xv6 realiza las transiciones de estado del proceso, que dan la pauta del código a modificar.
- Implementar una cola sin tener memoria dinámica es problemático. Sabiendo la longitud máxima de la cola, se puede reproducir su comportamiento utilizando un [buffer circular](#), que sólo requiere un arreglo de tamaño estático.

Extras

- Del scheduler:
- Generalizar el scheduler para que utilice una cantidad arbitraria de prioridades `NPRIO` definida vía `#define`.
- Hacer que el scheduler dé timeslices más largos a programas de menor prioridad.
- De las herramientas de medición:
- Mejorar los programas `verduiops` y `frutaflops` haciendo que usen `gettimeofday` implementado anteriormente.
- (Pesado) Llevar cuenta de cuánto tiempo de procesador se le ha asignado a cada proceso, con una system call para leer esta información desde espacio de usuario.

- Del informe:
- ¿Se puede producir *starvation* en el nuevo scheduler? Si fuera verdadero, ¿cómo?

Entrega

- Deberán entregar via commits+push a su grupo en bitbucket.
- Deberán crear un repositorio git nuevo **Lab3** dentro del grupo **so2014gXX** con un directorio **xv6** dentro sobre el cual deberán hacer sus modificaciones. No copiar el Lab1, comenzar en limpio.
- El *coding style* deberá respetar a rajatabla las convenciones de **xv6**.