

CC4302 Sistemas Operativos

Tarea 1 – Pintos

Alumno José Miguel Cisternas

Para esta tarea todos los cambios realizados se hicieron principalmente en 4 archivos, timer.c, synch.c, thread.c y thread.h, Además las líneas agregadas/cambiadas se encuentran definidas dentro del código entre dos separadores de la forma `/*-----cambios-----*/`. Los cambios realizados se resumen en lo siguiente:

thread.h:

A la estructura thread se le agregaron variables necesarias para poder materializar algunos estados o información necesaria para el thread, en este caso: tick_to_wake, variable que cuenta el tiempo que le falta para despertar, en caso de no estar durmiendo su valor es “-1”; d_to y d_from, hacen referencia a un thread en caso de que este donando o le estén donando; y por ultimo old_priority, guarda su prioridad antes de que le hayan donado.

thread.c:

En este archivo se encuentra el grueso de los cambios hechos para pintos, estos están repartidos en varias funciones que se encuentran a continuación:

- `thread_tick`: En esta función se le agrego la funcionalidad de reducir en 1 tick la variable tick_to_wake de todos los threads, en caso de que alguno llegue a un numero menor que 0 se desbloqueará el thread.
- `thread_unblock`: se le agrego la condición para hacer un yield al thread actual en caso de que el thread que se desbloquee tenga mejor prioridad que el thread corriendo. Hay que agregar que se le añadió la condición necesaria para que no se ejecutara un yield en una interrupción
- `set_priority`: se le agrego una condición para que frente a un cambio de prioridad si este thread queda con menos prioridad que algún thread en la lista de ready se haga yield.
- `init_thread`: se agrega la iniciación de las variables creadas en thread.h.
- `next_thread_to_run`: se modifica para poder hacer round robin, de esta manera se busca el primer thread con mayor prioridad en la lista de ready y luego este se selecciona para ser el siguiente a ejecutarse, en caso que la lista este vacia se devuelve el thread idle.

timer.c:

En este archivo lo único que se cambio fue la implementación del sleep, la cual se modifiko para no producir busy wait, la solución consiste en cambiar la variable de tick_to_wake a los ticks pedidos y luego bloquear el thread, de esta manera cuando pasen los ticks este sea desbloqueado por la función del thread_tick.

synch.c:

En synch se encuentran principalmente las funciones para los locks, en este archivo se hicieron los cambios necesarios para la donación y los cambios de prioridad. Se modifiko el sema_up de manera que el thread que sea despertado sea siempre el con mayor prioridad. Además se cambiaron las funciones de lock_acquire y lock_release, en estas se agregan las condiciones para la donación, en caso de que algún thread tenga el holder, se verificara si el thread tiene mayor prioridad, en este caso se le entrega la prioridad al thread que tiene el lock y luego se va a dormir, luego en el release se revierten estos cambios.

