

Sistemas Operativos

Práctico 2

Curso 2024

Objetivos

- Comprender la administración de procesos e hilos de los sistemas operativos.
- Comprender la tarea y la elección de un planificador.

Duración

- 1 semana.
- **Con prueba opcional sin material en la segunda hora de los prácticos de la semana**

Ejercicio 1 (básico) Considere un proceso a nivel del sistema operativo.

- Realice un diagrama de los estados de un proceso y sus transiciones. Explique cuándo se produce cada transición.
- ¿Qué es el PCB (Process Control Block)? Describa sus campos más importantes.
- ¿Por qué en el PCB se reserva lugar para los registros de la CPU?

Ejercicio 2 (básico) Considere el concepto de hilo (thread)

- Indicar la secuencia de pasos que sigue el sistema operativo al realizar un cambio de contexto entre Hilos y entre Procesos.
- Cite dos ventajas que tienen los Hilos por sobre los Procesos. Luego indique qué desventaja importante tienen los Hilos. Sugiera una aplicación que se beneficiaría del uso de hilos y una que no lo haría.
- Considerando los componentes de data (datos), heap (memoria dinámica) y stack (pila) y variables en registros del CPU. Cuáles son compartidos entre:
 - Un proceso y sus hijos.
 - Los threads de un mismo proceso.
- ¿Qué diferencias hay entre los hilos a nivel de usuario (user-level threads) y los hilos a nivel del núcleo (kernel-level threads)? ¿En qué circunstancias es un tipo mejor que el otro?

Ejercicio 3 (básico) Defina la diferencia entre planificación expropiativa (preemptive) y no expropiativa (non preemptive). Explique en qué ámbitos sería preferible utilizar uno u otro mecanismo.

Ejercicio 4 (medio) ¿Qué ventaja tendría definir *quantum* de tiempo de diferente tamaño en distintos niveles de un sistema de colas multinivel?

Ejercicio 5 (avanzado) Considere el siguiente algoritmo de planificación por prioridad expropiativo basado en prioridades que cambian dinámicamente. Un número de prioridad mayor implica una prioridad más alta. Mientras un proceso está esperando la CPU (en la cola de procesos listos), su prioridad cambia según α . Cuando está ejecutándose, su prioridad cambia según β . Todos los procesos reciben una prioridad de 0 al ser creados e ingresar por primera vez a la cola de listos. Los parámetros α y β pueden ajustarse para dar muchos algoritmos de planificación distintos.

- ¿Qué algoritmo se obtiene si $\beta > \alpha > 0$?
- ¿Qué algoritmo se obtiene si $\beta < \alpha < 0$?

Ejercicio 6 (medio) Suponga que un algoritmo de planificación favorece los procesos que han consumido la menor cantidad de tiempo de procesador en el pasado reciente. ¿Por qué este algoritmo favorecería a los procesos limitados por E/S pero sin postergar infinitamente los procesos limitados por CPU?

Ejercicio 7 (avanzado) Consideremos n procesos compartiendo la CPU en forma Round Robin. Asumiendo que cada cambio de proceso lleva s segundos, ¿cuál debe ser el quantum q , para que el tiempo desperdiciado en el cambio de procesos sea minimizado y al mismo tiempo se garantice que cada proceso tenga control de la CPU por lo menos cada t segundos?

Ejercicio 8 (medio) Sea un sistema monoprocesador multiprogramado. Considere el siguiente conjunto de procesos:

Proceso	Tiempo de ejecución (ms)	Prioridad
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Se supone que los procesos llegaron en el orden P_1, P_2, P_3, P_4, P_5 , todos en el instante $t = 0$. Suponga despreciable el tiempo requerido para el cambio de contexto.

- Realice un diagrama de tiempo vs procesos que ilustre qué proceso tiene asignado el procesador y el estado de cada proceso en cada instante de tiempo, utilizando planificación: FCFS, SJF, una técnica por prioridad no expropiativa (a menor número, mayor prioridad), RR con cuanto 1 y RR con cuanto 2.
- Calcule los tiempos de retorno y espera de los procesos anteriores con cada uno de los algoritmos de planificación.
- ¿Cuál algoritmo presenta un menor tiempo promedio de espera? (Considerando todos los procesos).

Ejercicio 9 (con solución) Se tiene un sistema operativo multiprogramado en el cual se dispone de un único procesador y un modelo de hilos **Mx1**.

Cuenta con un planificador por **prioridades** que sigue las siguientes reglas:

- Cuando un proceso ingresa a la cola de listos, su prioridad será el largo del próximo CPU-Burst (**a nivel de proceso**)
- Cada 5ms, los procesos en la cola de listos suman 5 a su prioridad
- El planificador es **expropiativo**

La planificación a nivel de hilos es **round robin** con un cuanto de 5ms.

Existen dos procesos P_1 y P_2 . P_1 a su vez cuenta con dos hilos, P_{1A} y P_{1B} , que ejecutan concurrentemente:

P1		P2
P1A	P1B	
Ejecuta 10ms	Ejecuta 15ms	Ejecuta 10ms
Bloquea 10ms	Bloquea 5ms	Bloquea 10ms
Ejecuta 5ms	Ejecuta 5ms	Ejecuta 10ms

Se asume que la operación de bloqueo lleva un tiempo despreciable, esto podría interpretarse como que la última instrucción del bloque de ejecución se encarga de bloquear.

Inicialmente en el sistema no se tiene ningún proceso de usuario ejecutando y en tiempo $t=0$ se lanza la ejecución de P_1 (P_{1A} y P_{1B}) y P_2 .

Se pide:

- (a) Realice un diagrama de planificación (tiempo vs hilos), comenzando en el tiempo $t=0$, indicando el estado de cada uno de los hilos y su posición en cada una de las colas. Se recomienda utilizar la siguiente notación: Estado_{prioridad} (un proceso en la cola de listos con prioridad 10 L_{10} , un proceso ejecutando E, un hilo en la primera posición de la cola de listos L_1 , etc.).
- (b) Calcule el tiempo de espera y el tiempo de retorno de cada proceso.

Ejercicio 10 (opening) Se tiene un sistema operativo multiprogramado en el cual se dispone de un único procesador. El planificador del sistema operativo utiliza una estrategia de planificación por prioridad **expropiativo**. Este sistema maneja hilos con modelo $M \times 1$, utilizando una planificación Round-Robin con quantum de 2ms a nivel de usuario. Las prioridades de los procesos se actualizan cada 2ms y se llama al planificador.

R1	R2	R3
Ejecuta 2ms	Ejecuta 4ms	Ejecuta 4ms
Bloquea 2ms	Bloquea 2ms	Bloquea 4ms
Ejecuta 2ms	Ejecuta 2ms	Ejecuta 8ms

Se tiene que el proceso P_1 cuenta con 2 hilos (P_1H_1 , P_1H_2) que ejecutan concurrentemente de forma que un hilo ejecuta el código de R1 y el otro R2. En cambio, P_2 cuenta con un único hilo de ejecución y ejecuta el código de R3. En el instante de tiempo inicial ($t = 0$) la cola de listos contiene a P_1 y P_2 . La prioridad está definida por el tiempo de cómputo restante (más tiempo más prioridad) y luego por menor identificador, es decir, P_i tendrá mayor prioridad que P_j si $i < j$.

En todos los casos, los empates se resuelven con el mismo criterio que la prioridad.

Se pide: Realice un diagrama de planificación (tiempo vs hilos), comenzando en el tiempo $t=0$, indicando el estado de cada uno de los hilos y su posición en cada una de las colas.

Ejercicio 11 (avanzado) Se tiene un sistema operativo simétrico, multiprogramado, con un planificador con dos colas de prioridad (alta/baja) con retroalimentación (es decir, una multi-level feedback queue con dos niveles de prioridad). En la cola de alta prioridad se utiliza un algoritmo round-robin con quantum de 2 unidades de tiempo mientras que en la otra un SJF expropiativo. Cualquier caso de empate se define con una prioridad dada por el índice del proceso, donde a menor índice, mayor prioridad (i.e. $P_1 > P_2 > \dots > P_n$).

A los procesos al ser creados se les asignan prioridad alta, que luego podrá ser modificada según los siguientes criterios:

- Un proceso baja a la cola de prioridad baja si en las últimas 2 unidades de tiempo a utilizado completamente el recurso procesador.
- Un proceso sube a la cola de prioridad alta si en las últimas 2 unidades de tiempo no ha usado el recurso procesador en absoluto.

En el instante $t=0$ se lanza la ejecución de tres procesos (P_1 , P_2 y P_3) donde P_1 y P_3 ejecutan el código de R1 mientras que P_2 el de R2.

R1	R2
Ejecuta 1	Ejecuta 1
Bloquea 3	Bloquea 3
Ejecuta 5	Ejecuta 4

Se pide: Asumiendo un sistema con dos procesadores, realice el diagrama de planificación (tiempo vs. Procesos) en que se indique el estado de cada proceso (listo/ejecutando/bloqueado/terminado) en cada intervalo de tiempo. Además, se debe indicar, en cada caso, el nivel de prioridad (alta/baja) en el que se encuentra el proceso.

Ejercicio 12 (Primer parcial 2023) Considere un sistema con un único procesador sobre el que ejecuta un sistema operativo con un planificador Round Robin con un cuanto de 10 ms y un modelo de hilos $M \times 1$. Inicialmente en el sistema no se tiene ningún proceso de usuario ejecutando y en el tiempo $t = 0$ se lanza la ejecución del siguiente programa de usuario:

```
begin
  pid = fork();
  if pid = 0 then
    Ejecuta 5ms
    Bloquea 5ms
    Ejecuta 5ms
  else
    create_thread(proc1)
    create_thread(proc1)
    print(pid)
  end if
end

procedure proc1()
begin
  Ejecuta 5ms
  Bloquea 10ms
  Ejecuta 10ms
end
```

La operación `fork()` crea un nuevo **Proceso** y retorna 0 al proceso Hijo (nuevo proceso con el mismo código) y un identificador del hijo al proceso Padre. La operación `create_thread()` crea un nuevo **Hilo** de ejecución y comienza su ejecución en el procedimiento recibido como argumento. Salvo que se indique explícitamente, el tiempo de ejecución de todas las funciones (`fork`, `create_thread`, condición del `if`, etc.) es de 5 ms. La asignación en `pid = fork()` se considera instantánea (solo cuenta el tiempo de ejecución del `fork`). La planificación definida para los hilos a nivel de usuario es SJF expropiativo. En caso de empates en las políticas de reemplazo se desempata por menor identificador (si los procesos P_i y P_j empatan, se selecciona como próximo proceso a ejecutar P_i si $i < j$ y P_j en caso contrario).

- Realice un diagrama de planificación (tiempo vs. hilos y tiempo vs. procesos), comenzando en el tiempo $t = 0$ e indicando el estado de cada uno de los hilos y procesos.
- Calcule el tiempo de espera y de retorno de cada proceso.