



Primer Parcial de Sistemas Operativos.  
Grupo 81 M. 20 de octubre de 2010.

NOTAS:

- \* La fecha de publicación de las notas, así como de revisión se notificarán por Aula Global.
  - \* Para la realización del presente examen se dispondrá de 1:25 hora.
  - \* No se pueden utilizar libros ni apuntes
  - \* Será necesario presentar el DNI o carnet universitario para realizar la entrega del examen
- 

Ejercicio 1 (4 puntos). Responda de forma clara y concisa a las siguientes preguntas. Espacio máximo una cara por pregunta.

- 1) ¿Cuáles son las principales funciones de un sistema operativo?
- 2) ¿Cómo se solicita una llamada al sistema operativo? Acompañelo con un esquema gráfico de los pasos a seguir.
- 3) ¿Qué es un proceso? ¿De que partes se compone la información de un proceso guardada en el BCP?
- 4) Indique los posibles estados de un proceso es su ciclo de vida y cómo se pasa de uno a otro en algoritmos con expulsión.  
¿Si el proceso tiene varios threads de biblioteca, que ocurre cuando uno de ellos hace una llamada al sistema read?

Ejercicio 2 (3 puntos)

Se desea implementar un sistema de avisos que permita saber que un proceso está vivo mediante señales que envía otro. Para ello será necesario implementar dos procesos:

- Avisador.
  - Proceso que cada 5 s. envía una señal SIGUSR al registrador.
  - El Avisador envía 5 señales al Registrador y luego le envía una señal SIGKILL y termina..
- Registrador.
  - Proceso que recibe las señales SIGUSR del Avisador y las trata escribiendo un mensaje en un archivo log, que debe abrir previamente y cerrar al final. Mensaje: Recibido aviso: número. Donde número empieza en 1 y se incrementa cada vez que hay un nuevo aviso.
  - En caso de que reciba SIGKILL debe terminar.

Se pide:

- a) Implementar el proceso Registrador.
- b) Implementar el proceso Avisador.
- c) Implementar una aplicación principal (proceso Padre) que ejecute los procesos Avisador y Registrador y que espere a que terminen.

Nota: Especificar claramente todas las llamadas al sistema usadas.



### Ejercicio 3 (3 puntos)

En un determinado sistema operativo que planifica usando colas por prioridades, los procesos se ejecutan en función de su prioridad (1 es la más alta). Se asume que los procesos entran en el sistema en su cola de prioridad y no se mueven de ella. Cuando varios procesos tienen la misma prioridad se utiliza una política de planificación en rodaja (Round-robin), con una rodaja de 100 ms. Cuando se acaba la rodaja de un proceso se le pone al final de su cola de prioridad.

En la siguiente tabla se especifica para cada proceso, su prioridad, su tiempo de llegada y el tiempo que necesitan para ejecutarse, distinguiendo CPU y Entrada/Salida bloqueante.

Nota: Con la entrada/salida bloqueante el proceso libera la CPU y esta queda lista para el proceso siguiente. Cuando están listos los datos de E/S, el proceso pasa a la cola de listos para ejecutar, pero al final de su cola de prioridad.

PROCESOS	PRIORIDAD	T. DE LLEGADA	T DE EJECUCIÓN
P1	3	0	250 CPU + 100 E/S + 200 CPU
P2	2	200	300 CPU
P3	1	400	100 CPU + 250 E/S + 50 CPU
P4	1	500	400 CPU
P5	2	400	100 CPU + 100 E/S + 100 CPU

Para las dos situaciones siguientes:

- Planificación sin expulsión. Se elige al preparado de más prioridad pero no se expulsa al que está.
- Planificación con expulsión. Se expulsa al que está ejecutando si llega uno de más prioridad o vence su rodaja de tiempo.

Se pide

- Realizar un cronograma de la ejecución de los procesos.
- Calcular el tiempo que cada proceso se mantiene en espera desde su llegada al sistema hasta que finaliza.
- Calcular el tiempo de retorno de cada proceso (tiempo transcurrido desde que el proceso llega hasta que finaliza su ejecución).
- El tiempo medio de espera y el tiempo medio de retorno.

