

Trabajo Práctico N° 2: Procesos y planificación

1. Asuma que en el instante de tiempo 5 de la ejecución de un sistema operativo no se utilizan recursos del sistema, excepto procesador y memoria. Considere los siguientes sucesos:

- Tiempo 5: P1 realiza una lectura de la unidad de disco 3.
- Tiempo 15: termina el tiempo de P5.
- Tiempo 18: P7 escribe en la unidad de disco 3.
- Tiempo 20: P3 lee del disco 2.
- Tiempo 24: P5 escribe en el disco 3.
- Tiempo 28: P5 es sacado al área de *swapping* (esto implica utilizar disco).
- Tiempo 33: La unidad de disco 2 genera una interrupción: la lectura de P3 se completó.
- Tiempo 36: La unidad de disco 3 genera una interrupción: la lectura de P1 se completó.
- Tiempo 38: P8 finaliza.
- Tiempo 40: La unidad de disco 3 genera una interrupción: la escritura de P5 se completó.
- Tiempo 44: P5 es traído desde el área de *swapping* (esto implica utilizar disco).
- Tiempo 48: La unidad de disco 3 genera una interrupción: la escritura de P7 se completó.

Para los tiempos 22, 37 y 47, identifique en qué estado está cada proceso. No es necesario que identifique el proceso en ejecución actual. Simplemente identifique los que están listos, y además si un proceso está bloqueado esperando indique qué es lo que espera.

2. Considere el siguiente conjunto de procesos:

Proceso	Tiempo de CPU requerido (en ms)
P1	6
P2	3
P3	2
P4	1
P5	5

Todos los procesos ingresan en $t = 0$ y se van ubicando en la cola de listos en el orden dado en la tabla. Los procesos se ejecutan mediante un algoritmo round robin con quantum de 2 ms. ¿Cuál es el tiempo de *context switch* máximo admisible para que el tiempo de uso efectivo de CPU desde $t = 0$ hasta el tiempo en el que todos los procesos finalizan sea mayor al 90%?

3. Considere el siguiente conjunto de procesos:

Proceso	Tiempo de CPU (en ms)	Prioridad
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

Los procesos arriban en orden ascendente en el tiempo cero:

- Grafique 4 diagramas de Gantt ilustrando la ejecución utilizando FCFS, SJF, prioridad sin desalojo y round robin con quantum de 1 ms. La prioridad más alta corresponde a los procesos con menor valor en la columna asociada (esto es, 1 es más prioritario que 3).
- Calcule t_T , t_w y t_r para cada proceso y cada algoritmo de scheduling de a).
- Compare y discuta los resultados.
- Algunos sistemas operativos modernos utilizados en hardware móvil (ej. Android) poseen políticas para regular el uso de CPU para extender el tiempo de batería. Resuelva b) suponiendo que se mantiene en todo momento un tope de uso de CPU del 80 % calculado sobre una ventana de 5 ms y se “duerme” el 20 % restante.

Recuerde que:

t_T : *turnaround time*: tiempo total de ejecución de un proceso.

t_w : *waiting time*: tiempo total de espera en la cola de listos.

t_r : *response time*: diferencia entre tiempo de entrada y tiempo de la primera asignación de CPU.

- Considere el siguiente conjunto de procesos:

Proceso	Tiempo de CPU requerido (en ms)	Tiempo de arribo (en ms)	Prioridad
P1	6	0	3
P2	1	1	2
P3	2	2	2
P4	1	3	1
P5	5	4	3

- Grafique diagramas de Gantt ilustrando la ejecución utilizando los algoritmos SJF con desalojo y prioridad con desalojo. La prioridad más alta corresponde a los procesos con menor valor en la columna asociada (esto es, 1 es más prioritario que 3).
- Por cada uno de los dos algoritmos de a), calcule el tiempo promedio de turnaround para los procesos. ¿Qué algoritmo ofrece menor tiempo promedio de turnaround?

- Considere ahora el siguiente conjunto de procesos:

Proceso	Tiempo de CPU (en ms)	Petición de E/S
P1	4	Lectura de sector disco en $t=3$
P2	1	No realiza E/S
P3	3	Imprime datos en $t=1,2$
P4	1	Imprime por pantalla en $t=1$
P5	2	No realiza E/S

Nuevamente, los procesos arriban en orden ascendente en el tiempo cero, y son gestionados con *round robin* de 2 ms. Algunos procesos realizan peticiones de E/S en cierto tiempo t (relativo al tiempo de inicio de ejecución del proceso). Asuma que una petición de E/S lleva 2 ms.

- Grafique un diagrama de Gantt mostrando el uso de CPU y dispositivos de E/S.
- ¿En qué estado se encuentran los procesos en $t=3,5,7$ (relativo al inicio de la ejecución de todos los procesos)?

- Considere dos procesos comunicados por un pipe:

```
ls | more
```

El sistema operativo realiza scheduling según un algoritmo *round robin* con quantum t . El pipe posee un búfer intermedio de 1024 bytes. `ls` es capaz de producir 512 bytes cada 10 ms, mientras `more` lee 2048 bytes en el mismo tiempo.

- a) Determine cuáles serían las condiciones por las cuáles un proceso iría al estado de bloqueado.
- b) Grafique un diagrama de Gantt mostrando el uso de CPU y dispositivos de E/S con $t = 5ms$, $10ms$, $40ms$.

7. ¿Cuál es el resultado del siguiente código? Justifique.

```
main() {
    printf ("%d", proceso());
    exit(0);
}

int proceso () {
    if (!fork())
        return 1;
    else
        return 0;
}
```

- a) 0
- b) 1
- c) 0 1
- d) No da un resultado determinista.

NOTA: `fork()` realiza el clonado del proceso actual, retornando la función el valor 0 en el proceso hijo y el PID (process ID) del hijo en el proceso padre.