

# Primer parcial de Sistemas Operativos

3 de mayo de 2019

Lea detenidamente las siguientes instrucciones. No cumplir los requerimientos puede implicar la pérdida del parcial.

## Formato

- Indique su nombre completo y número de cédula en cada hoja (no se corregirán las hojas sin nombre). Numere todas las hojas e indique la cantidad total de hojas en la primera.
- Escriba las hojas de un solo lado y empiece cada problema en una hoja nueva.
- Si se entregan varias versiones de un problema solo se corregirá la primera de ellas.

## Dudas

- Sólo se contestarán dudas de letra.
- No se aceptarán dudas en los últimos 15 minutos del parcial.

## Material

- El parcial es **SIN** material (no puede utilizarse ningún apunte, libro ni calculadora). Sólo puede tenerse las hojas del parcial, lápiz, goma y lapicera en su banco. Todas sus demás pertenencias debe colocarlas en el piso debajo de su asiento.

## Finalización

- El parcial dura 2,5 horas.
- Al momento de finalizar el parcial no se podrá escribir absolutamente nada en las hojas, debiéndose parar e ir a la fila de entrega. Identificar cada una de las hojas con nombre, cédula y numeración forma parte de la duración del parcial.

---

## Problema 1 (15 pts)

- (3 pts) ¿Cuál es la principal ventaja de la multiprogramación? ¿Que problemas nuevos le acarrea el sistema operativo?
- (3 pts) ¿Para qué sirve la protección de CPU? ¿Cómo se implementa?
- (3 pts) Describa las acciones del sistema operativo cuando ocurre un cambio de contexto
  - entre hilos.
  - entre procesos.
- (3 pts) ¿En qué momentos es invocado el planificador de corto plazo del sistema operativo?. Analice la frase *"mas vale tarde que nunca"* en el contexto de un planificador de corto plazo en un sistema de tiempo real.
- (3 pts) Describa los 2 comportamientos de la primitiva `signal` en monitores. Cuando se programa utilizando monitores, ¿cómo se puede lograr que el programa funcione correctamente independientemente del comportamiento del `signal`? ¿En qué se diferencia la operación `signal()` asociada a monitores de la operación `V()` asociada a semáforos?

## Problema 2 (15 pts)

Se tiene un sistema operativo multiprogramado en el cual se dispone de 2 procesadores. El planificador del sistema operativo utiliza una estrategia de planificación Round-Robin con quantum de 10 ms. Este sistema maneja hilos con modelo Mx1, utilizando una planificación Round-Robin con quantum de 5 ms a nivel de usuario.

Se tienen los siguientes procesos:

### P1

Ejecuta 10ms  
Bloquea 5ms  
Ejecuta 10ms

### P2

Ejecuta 5ms  
Bloquea 10ms  
Ejecuta 20ms

### P3

Ejecuta 10ms  
Bloquea 10ms  
Ejecuta 10ms

Se tiene que **P1** cuenta con 2 hilos que ejecutan concurrentemente de forma que cada hilo ejecuta todo el código del proceso **P1**. En cambio, **P2** y **P3** cuentan con un solo hilo de ejecución. En el instante de tiempo inicial ( $t = 0$ ) la cola de listos contiene a **P1**, **P2** y **P3** (en este orden).

Se pide:

- (2 pts) Realice un diagrama que muestre los estados y transiciones que tiene un proceso cualquiera en un sistema operativo con esas características. Describa brevemente cada componente del diagrama.
- (10 pts) 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.

**Solución:** El sistema utiliza un modelo de hilos con modelo  $M \times 1$  por lo que los hilos del proceso **P1** ejecutarán en espacio de usuario y utilizarán un único hilo de núcleo. Esto implica que cuando uno de los hilos de **P1** se bloquea, se bloqueará todo el proceso y por lo tanto también el otro hilo.

Nomenclatura:

- **En:** Ejecuta en el CPU "n"
- **Ln:** En la posición "n" de la cola de listos en espacio de núcleo
- **Ln':** En la posición "n" de la cola de listos en espacio de usuario
- **B:** Bloqueado
- **T:** Terminado

<b>P1H1</b>	E1	L1'	E1	B	L1'	L1'	L1'	E1	L1'	E1	T	T	T	Espacio de usuario
<b>P1H2</b>	L1'	E1	L1'	L1'	E2	B	L2'	L1'	E1	L1'	E1	T	T	
<b>P1</b>	E1	E1	E1	B	E2	B	L1	E1	E1	E1	E1	T	T	Espacio de núcleo
<b>P2</b>	E2	B	B	E1	E1	E1	E1	T	T	T	T	T	T	
<b>P3</b>	L1	E2	E2	B	B	E2	E2	T	T	T	T	T	T	
<b>Tiempo</b>	0	5	10	15	20	25	30	35	40	45	50	55	60	

**NOTA:** en  $t = 15$  sucede simultáneamente que el hilo **P1H1** se bloquea y se termina su quantum. En la simplificación planteada siempre es posible que el hilo se bloquee (aún cuando se acabe su quantum) porque la operación de bloqueo no requiere uso de CPU. Sin embargo se considera como correcta una planificación que pospone este bloqueo hasta la próxima vez que se le asigna la CPU al hilo.

- (3 pts) Calcule y defina porcentaje de uso de CPU, tiempo de retorno y tiempo de espera de cada hilo.

**Solución:**

**Tiempo de retorno:** Es el intervalo de tiempo desde que un hilo es cargado hasta que este finaliza su ejecución. En este caso todos los hilos son cargados en el instante  $t = 0$ .

- P1H1: 50ms
- P1H2: 55ms
- P1: 55ms
- P2: 35ms
- P3: 35ms

**Tiempo de espera:** Es la suma de los intervalos de tiempo que un hilo estuvo en la cola de procesos listos (ready queue).

- P1H1:  $(10 - 5) + (35 - 20) + (45 - 40) = 25\text{ms}$
- P1H2:  $(5 - 0) + (20 - 10) + (40 - 30) + (50 - 45) = 30\text{ms}$
- P1:  $(35 - 30) = 5\text{ms}$
- P2:  $0\text{ms}$
- P3:  $(5 - 0) = 5\text{ms}$

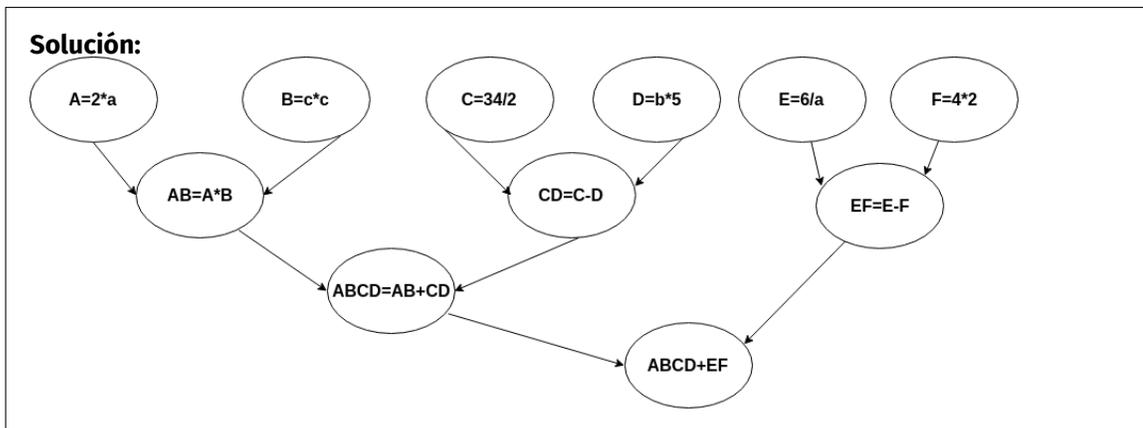
**Uso de CPU:** Es el porcentaje de uso (en cuanto a ejecución de tareas de usuario o del sistema que son consideradas útiles) que tiene un procesador.

- CPU 1:  $\frac{55}{55} \times 100 = 100,00\%$
- CPU 2:  $\frac{(15-0)+(35-20)}{55} \times 100 = 54,55\%$
- Total:  $\frac{(15-0) \times 2 + 20 - 15 + (35-20) \times 2 + 55 - 35}{55 \times 2} \times 100 = 77,27\%$

**Problema 3** (8 pts)

- (a) (3 pts) Implemente con `cobegin` y `coend` la solución de la expresión 1 con el máximo nivel de concurrencia posible.

$$(2 \times a \times c^2 + 34/2 - 5 \times b) + (6/a - 4 \times 2) \quad (1)$$



```
cobegin
  begin
    cobegin
      begin
        cobegin
          A=2*a
          B=c*c
        coend
        AB=A*B
      end
      begin
        cobegin
          C=34/2
          D=b*5
        coend
        CD=C-D
      end
    coend
    ABCD=AB+CD
  end
  begin
    cobegin
      E=6/a
      F=4*2
    coend
    EF=E-F
  end
coend
return ABCD+EF
```

- (b) (5 pts) Implemente con semáforos la solución al problema del productor-consumidor con buffer finito de tamaño  $N$  pero considerando que el sistema tiene la capacidad de devolver un valor útil (al consumidor) cuando el buffer está vacío. Se dispone de los siguientes funciones/procedimientos:

**producir()** : Función, produce y devuelve un producto.

**consumir(p)** : Procedimiento, consume el producto  $p$ .

**insertar(p)** : Procedimiento, inserta el producto  $p$  en el buffer.

**obtener()** : Función, devuelve el primer producto del buffer. En caso que el buffer esté vacío igual devuelve un valor válido que debe ser consumido.

**vacío()** : Función, devuelve **true** si el buffer está vacío, **false** en caso contrario.

**Solución:**

```
sem capacidad , mutex;

void productor() {
    p=producir();

    P(capacidad);
    P(mutex);
    insertar(p);
    V(mutex);
}

void consumidor() {
    P(mutex);
    if (!vacio()) V(capacidad);
    p=obtener();
    V(mutex);

    consumir(p);
}

void main() {
    init(capacidad , N);
    init(mutex, 1);
    cobegin {
        productor();... productor();
        consumidor();... consumidor();
    }
}
```