

Primer parcial de Sistemas Operativos

27 de setiembre de 2022

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 **3 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.

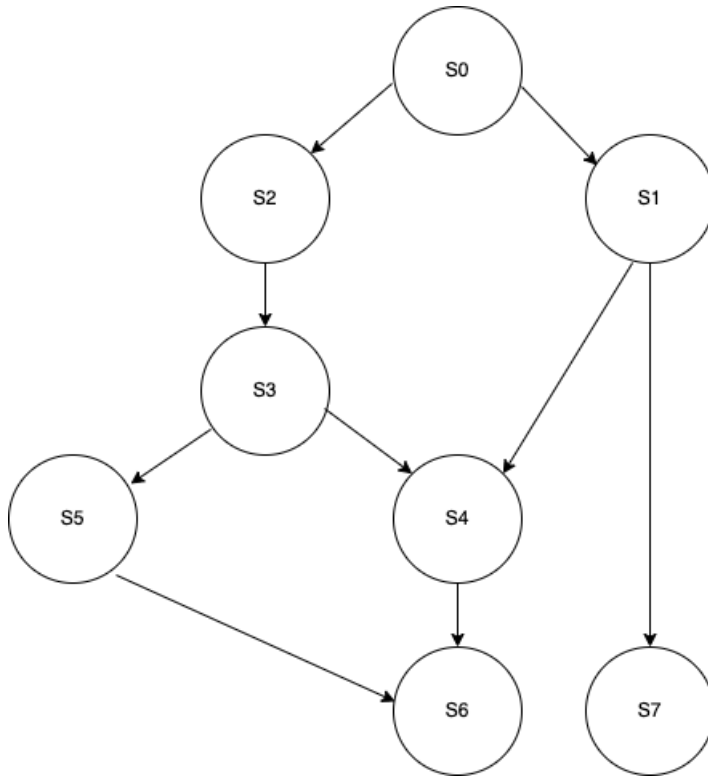
Importante: Debe justificar todas las respuestas. Siempre.

Problema 1 (8 pts)

- (a)
 - (1 pt) Describa brevemente los 3 métodos vistos en el curso para que el Sistema Operativo para efectuar una operación de entrada salida.
 - (1 pt) Elija uno de ellos y describa sus principales características.
- (b)
 - (1 pt) Describa brevemente la arquitectura de un Sistema Operativo con arquitectura en capas.
 - (1 pt) Describa ventajas y desventajas de esta arquitectura.
- (c) (2 pts) Describa bajo qué condiciones es posible construir un Sistema Operativo seguro sin contar con un modo de operación privilegiado en hardware. Justifique su respuesta.
- (d)
 - (1 pt) Describa brevemente qué es un “context switch” (cambio de contexto).
 - (1 pt) Describa brevemente las distintas tareas que implica un “context switch” (cambio de contexto).

Problema 2 (8 pts)

Considere el siguiente grafo de precedencia para los procesos S_0, S_1, \dots, S_7 .



(a) (2 pts) Justifique si el grafo es representable con Cobegin-Coend.

Solución:

No se puede representar este grafo. Solo aquellos grafos que formen subgrafos disjuntos concurrentes pueden ser representados con cobegin-coend.

(b) (6 pts) En caso afirmativo, de una representación con Cobegin-Coend. En caso contrario, de una representación con la herramienta más abstracta posible.

Solución:**Begin**

```

total1 := 2;
total2 := 2;
S0;
Fork L2;
S2;
S3;
Fork L4;
Join total1 NEXT;
L2:
S1;
Fork L7;
Join total1 NEXT;
NEXT:

```

```
S4;  
  Goto L5;  
L4:  
  S5;  
L5:  
  Join total2 L6;  
L7:  
  S7;  
  Goto FIN  
L6:  
  S6;  
FIN:  
  
End
```

Problema 3 (24 pts)

En una piscina, por medidas de seguridad, los nadadores amateurs solo pueden nadar utilizando flotadores. Dependiendo del nadador podrá requerir uno o dos flotadores. La piscina dispone de 30 flotadores. La piscina también es utilizada por nadadores profesionales, que no utilizan flotadores pero no pueden compartir la piscina con los nadadores amateurs y, como máximo, pueden haber 10 de ellos nadando a la vez en la piscina. Cuando llega un nadador profesional tiene prioridad para utilizar la piscina (es decir, no ingresarán más nadadores amateurs hasta que haya nadado el profesional), los nadadores amateurs utilizan la piscina en orden de llegada.

Se pide: Implementar con monitores los procesos nadador_amateur y nadador_profesional según la realidad planteada.

Se dispone de las siguientes funciones:

dar_flotador(): invocada por los nadadores amateurs y retorna hasta un flotador. En caso de que sea invocada una segunda vez por un nadador que requiere un flotador no retornará nada. Una vez invocada queda esperando obtener un flotador.

nadar(): invocada tanto por los nadadores amateurs como los profesionales para nadar.

Solución:

```
monitor piscina{  
  int flotadores = 30;  
  int cantPros = 0;  
  int cantAmateurs = 0;  
  condition esperaAmateur, esperaPro;  
  
  procedure llegaAmateur(){  
    cantAmateurs++;  
    if(cantAmateurs>=29 || cantPros>0) {  
      esperaAmateur.wait();  
    }  
    amateursNadando++;  
  }  
  
  procedure saleAmateur(){  
    amateursNadando--;  
    cantAmateurs--;  
  }  
}
```

```
        if(amateursNadando==0 && cantPros>0) {
            for (int i = 0; i < 10; ++i) {
                esperaPro.signal();
            }
        }
        else if(cantPros==0) {
            esperaAmateur.signal();
        }
    }

    procedure nadaPro(){
        cantPros++;
        if(cantPros>10 || amateursNadando>0) {
            esperaPro.wait();
        }
    }

    procedure salePro(){
        cantPros--;
        if(cantPros>0) {
            esperaPro.signal();
        }
        else {
            for (int i = 0; i < 29; ++i){
                esperaAmateur.signal();
            }
        }
    }
}

nadador_profesional(){
    piscina.nadaPro();
    nadar();
    piscina.salePro();
}

nadador_amateur(){
    piscina.llegaAmateur();
    dar_flotador();
    dar_flotador();
    nadar();
    piscina.saleAmateur();
}

cobegin
    nadador_profesional();
    ...
    nadador_profesional();
    nadador_amateur();
    ...
    nadador_amateur();
coend
```