

Augusto

# Sistemas Operativos

Departamento de Computación - FCEyN - UBA  
Primer cuatrimestre de 2019

Nombre y apellido: SANTIAGO BESTINI  
Nº orden: 6 L.U.: 311/17 Cant. hojas: 5

## Recuperatorio 1er parcial - 27/6 - Primer cuatrimestre de 2019

1	2	3	4	Nota
R	B	B	B	(A)

**ACLARACIONES:** 1) Numere las hojas entregadas. Esta hoja se entrega y es la hoja cero. Complete en la primera hoja la cantidad total de hojas entregadas (sin contar el enunciado). 2) Realice cada ejercicio en **hojas separadas** y escriba **nombre, apellido y L.U. en cada una**. 3) Cada ejercicio se califica con **Bien, Regular o Mal**. La división de los ejercicios en incisos es meramente orientativa. Los ejercicios se califican globalmente. El parcial se aprueba con 2 ejercicios bien y a lo sumo 1 mal/incompleto. 4) El parcial **NO** es a libro abierto. 5) **Justifique adecuadamente cada una de sus respuestas.**

### Ejercicio 1.

Se tienen las siguientes tareas

Proceso	Tiempo total procesamiento	Instante de llegada	Tiempo de bloqueo (relativo al procesamiento)
P1	50	0	46-48
P2	40	8	16-20
P3	20	10	-
P4	60	28	-

- a) Realice un diagrama de Gant para un sistema con dos procesadores y un scheduler de tipo Round Robin con una única global de tareas y quantum de 20ms. Asuma que el costo de cambio de contexto es 2ms y el costo de migración de núcleos es de 6ms.
- b) Calcular el *waiting time* promedio. Escriba el razonamiento de resolución del cálculo

### Ejercicio 2.

Se desea implementar una función `telefonoDescompuesto` que toma dos parámetros: una cadena de texto indicando el nombre de un programa  $p$  y un entero  $n$  (mayor a 1) indicando la cantidad de procesos que deben generar un anillo.

La función debe ejecutar el programa  $p$  (sin utilizar la función `system`) y redirigir su salida  $s$  al proceso 0 del anillo. El proceso número 0 debe agregarle un caracter al azar a la cadena  $s$  (utilizar la función `char letraAlAzar()`) y enviar ese nuevo texto al proceso número 1, el cual realizará el mismo procedimiento y lo enviará al proceso 2. Una vez que la cadena llegue al proceso  $n - 1$  este también agregará su caracter y se lo pasará al proceso 0 que deberá imprimir por pantalla la cadena inicial  $s$  y la cadena resultante de la vuelta  $s'$ .

### Ejercicio 3.

- a) Considerar la siguiente secuencia de referencias a páginas: 1, 2, 3, 1, 2, 4, 1, 5, 4, 2, 2, 5  
Indique cómo se ubican las páginas en los marcos disponibles a medida que se van realizando los pedidos de páginas y cuántos fallos de página se producirán con el algoritmos de reemplazo **FIFO** suponiendo que se tienen 3 marcos de página
- b) Considerar la siguiente secuencia de referencias a páginas: 1, 2, 3, 1, 3, 1, 4, 5, 1, 3  
Indique cómo se ubican las páginas en los marcos disponibles a medida que se van realizando los pedidos de páginas y cuántos fallos de página se producirán con el algoritmos de reemplazo de **Segunda Oportunidad** suponiendo que se tienen 3 marcos de página

### Ejercicio 4.

Una empresa quiere agregar un baño público que pueda ser utilizado por cualquier género, pero para eso el jefe (que tiene un pensamiento binario) impone dos reglas:

- a) No puede haber al mismo tiempo en el baño mujeres y hombres
- b) No puede haber más de tres personas en el baño al mismo tiempo

**Valores Iniciales**

```

vacio = Semaforo(1)
switchHombres = Lightswitch()
switchMujeres = Lightswitch()
barreraHombres = Semaforo(3)
barreraMujeres = Semaforo(3)

```

**Mujer**

```

switchMujeres.lock(vacio)
  barreraMujeres.wait()
  entrar()
  barreraMujeres.signal()
switchMujeres.unlock(vacio)

```

**Hombre**

```

switchHombres.lock(vacio)
  barreraHombres.wait()
  entrar()
  barreraHombres.signal()
switchHombres.unlock(vacio)

```

**Lightswitch**

```

class Lightswitch:

    int contador = 0;
    Semaforo mutex = Semaforo(1);

    def __init__ ():
        self.contador = 0
        self.mutex = Semaphore (1)

    def lock (semaforo):
        self.mutex.wait() ← LOBBY IN
        self.contador += 1
        if(self.contador == 1):
            semaforo.wait() → LOBBY OUT
            self.mutex.signal()

    def unlock(semaforo):
        self.mutex.wait() ← LOBBY IN
        self.contador -= 1
        if(self.contador == 0):
            semaforo.signal()
            self.mutex.signal() → LOBBY OUT.

```

Argumente si se cumplen o no las siguientes propiedades:

- EXCL** (No puede haber simultáneamente hombres y mujeres en el baño)
- 3MAX** (Nunca hay más de tres personas en el baño)
- STARVATION-FREEDOM**

**Importante:** Notar que los dos `Lightswitch` comparten el mismo semáforo `vacio` al momento de llamar a `lock` y `unlock`, pero cada uno tiene su propia variable interna `contador` y su propio `mutex`.



①

FBSJINI SANTIAGO 311/37

FECHA

1. PARA CALCULAR EL WAITING TIME PROMEDIO TENEMOS QUE SUMAR EL TIEMPO DE ESPERA DE CADA UNO DE LOS PROCESOS ~~Y SUMAR~~ Y DIVIDIRLO POR LA CANTIDAD DE PROCESOS, ~~DE ESTA MANERA PODRÍAMOS SABER CUANTO TIEMPO ESP~~

PARA ESTO, SUMAMOS LAS R'S EN EL DIAGRAMA DE GANTT EN LA HOJA ADJUNTA. NOTAMOS QUE NI ~~EN~~ LA CARGA INICIAL, NI EL CAMBIO DE CONTEXTO NI LA MIGRACIÓN DE NODOS NI EL BLOQUEO CORRESPONDEN A TIEMPO DE ESPERA YA QUE SON MECANISMOS EN LOS QUE SE ESTÁN HACIENDO COSAS, MIENTRAS QUE EN R (READY/USO) EL PROCESO ÚNICAMENTE ESTÁ ESPERANDO A QUE SE LIBERE UN PROCESADOR.

24-26

TIEMPO ESPERA DEL PROCESO 1 : 7 BLOQUES — 54-66

TIEMPO DE ESPERA PROCESO 2 : 11 BLOQUES — 30-52

TIEMPO ESPERA PROCESO 3 : 5 BLOQUES — 12-22

TIEMPO ESPERA PROCESO 4 : 10 BLOQUES — 30-44  
 68-74

$$\begin{aligned} \text{WAITING TIME PROMEDIO} &= \frac{7 + 11 + 5 + 10}{4} \cdot \underbrace{2 \text{ MS}}_{\text{TIEMPO QUE REPRESENTA UN BLOQUE.}} \\ &= \frac{33 \cdot 2 \text{ MS}}{4} = 16,5 \text{ MS.} \end{aligned}$$





3

PESTINI SANTIAGO

31/1/17

FECHA

n=3  
 R ← 4  
 P → 0 → 1 → 2 → 3

```
2. void telefonodescompuesto (string p, int n) {
```

```
  vector<int [2]> vpipes;
```

```
  vpipes.push_back
```

```
  int pipe [2]; while (!pipe (pipe)) {
```

Si NO RECUERDO MAL PIPE() DEVOLVIA 0 EN CASO DE ERROR Y SI NO PUDO CREAR.

```
  vpipes.push_back (pipe);
```

```
  if (fork () == 0) {
```

NO ESTOY Y SEGURO DE COMO LLAMAR A LA FUNCION, ASI QUE EXPLICO: QUIERO REDIRIGIR LA SALIDA ESTANDAR DEL CHILD AL PIPE O SE QUE EL [1] DEL PIPE ES ESCRITURA Y SUPONGO I = COUNT.

```
    dup2 (vpipes [0] [1], 1);
```

```
    execl (p);
```

→ CAMBIO A LA EJECUCION DE P.

```
  for (int i = 0; i < n; i++) {
```

```
    int pipe [2];
```

```
    while (!pipe (pipe)) {
```

```
      vpipes.push_back (pipe);
```

ESTO QUEDA PARA EL PADRE.

```
int id = n;
```

```
  for (int i = 0; i < n; i++) {
```

```
    if (fork () == 0) {
```

```
      id = i;
```

```
      i = n;
```

```
  for (int i = 0; i <= n; i++) {
```

```
    if (i == id) → close (vpipes [i] [1]);
```

```
    else if (i == id + 1) and id != n →
```

```
      close (vpipes [i] [0]);
```

```
      close (vpipes [i] [1]);
```

CIERRO EL PIPE DE ESCRITURA DEL PIPE QUE ME CONECTA CON EL MODEO ANTERIOR, YA QUE SOLO QUIERO LEER DESDE.

CIERRO LOS PIPES QUE NO USO.

ANADIDO A, N NO CIERRA PORQUE NO EXISTE!

```
if (id < n) {  
    STRING PALABRA = READ(VPIPER[id][0]);  
    STR_CONCAT(PALABRA, LETRAALAZAR());  
    WRITE(VPIPER[id][1], PALABRA);  
    CLOSE(VPIPER[id][0]);  
    CLOSE(VPIPER[id][1][1]);  
    EXIT();  
}
```

```
STRING PALABRA = READ(VPIPER[id][0]);  
CLOSE(VPIPER[id][0]);  
PRINT(P);  
PRINT(PALABRA);
```

```
RETURN
```



(4)

FBSTINI SANTIAGO 3/1/27

FECHA

3\_A\_Fifo 1, 2, 3, 1, 2, 4, 1, 5, 4, 2, 2, 5

	X	X	X	FAULO PÁGINA	CADENA FIFO:
0: 0					
1: 1	1	X	X	X	1
2: 2	1	2	X	X	1-2
3: 3	1	2	3	X	1-2-3
4: 1	1	2	3		1-2-3
5: 2	1	2	3		2-2-3
6: 4	4	2	3	X	2-3-4
7: 1	4	1	3	X	3-4-1
8: 5	4	1	5	X	4-1-5
9: 4	4	1	5		4-1-5
10: 2	2	1	5	X	1-5-2
11: 2	2	1	5		1-5-2
12: 5	2	1	5		1-5-2

• SE PRODUCEN 7 FALLOS DE PÁGINA.

• RECORDEMOS QUE FIFO ES UNA POLÍTICA DE REEMPLAZO QUE CUANDO SE PRODUCE UN FAULO DE PÁGINA Y NO HAY PÁGINAS LIBRES EN MEMORIA, REEMPLAZA A AQUELLA QUE ESTABA MÁS TIEMPO.

B →

B\_ SECOND CHANCE: 1, 2, 3, 1, 3, 1, 4, 5, 1, 3

	X	X	X	FALLO PÁGINA
0: 1	1	X	X	X
1: 1	1	2	X	X
2: 2	1	2	3	X
3: 3	1	2	3	
4: 1	1	2	3	
5: 3	1	2	3	
6: 1	1	2	3	
7: 4	1	4	3	X
8: 5	5	4	3	X
9: 1	5	1	3	X
10: 3	5	1	3	

CADESA FIFO + REFERENCIA.

1

1-2

1-2-3

1(R)-2-3

1(R)-2-3(R)

1(R)-2-3(R)

2-3(R)-1 → 3(R)-1-4

1-4-3 → 4-3-5

3-5-1

3(R)-5-1

• SE PRODUCEN 6 FALLOS DE PÁGINA.

- RECORDAMOS QUE SEGUNDA OPORTUNIDAD ES UN PROTOCOLO DE REEMPLAZO BASADO EN FIFO (ES. A), CON UNA MODIFICACIÓN QUE LE PERMITE ORDENAR UNA SEGUNDA OPORTUNIDAD A AQUELLAS PÁGINAS QUE FUEON REFERENCIADAS. DE ESTA MANERA, EN VEZ DE CAMBIAR LAS PÁGINAS REFERENCIADAS, ESTAS SON DEVUELTAS AL FINAL DE LA CADESA, PERO SIN SU CONDICIÓN DE REFERENCIA.

• LA REFERENCIA NO ES ACUMULATIVA (BOOL) Y EN EJERCICIO, LA NOSO "LR)".



5

FESTINI SANTIAGO 31/1/27

FECHA

- 4\_ • NO PUEDE HABER AL MISMO TIEMPO MÚJERES Y HOMBRES.
- NO PUEDE HABER MÁS DE 3 PERSONAS AL MISMO TIEMPO.

• EL CÓDIGO TIENE UN PROBLEMA EN EL LIGHTSWITCH.LOCK() VEMOS QUE SOLO ESPERA AL SEMAFORO VACÍO SI CONTADOR == 1.

ANALICEMOS EL SIGUIENTE ESCENARIO:

→ UNA MUJER ENTRA AL BAÑO. → VACÍO = LOCKEO, OSEA OCUPADO

→ UN HOMBRE TRATA DE ENTRAR.

↳ SWITCH HOMBRES.LOCK(VACÍO). →

~~MAI !!!~~ NO HABÍA NOTADO EL MUTEX PROPIO DE LA CLASE. ✓

→ EL CÓDIGO SATISFACE LA PROPIEDAD EXCL. PENSAMOS EN ~~QUE~~ QUE EL BAÑO COMPARTIDO POSEE ADERÁS DOS LOBBYS, UNO PARA CADA GÉNERO DE CAPACIDAD 1 PERSONA. ESTOS LOBBYS ~~SON~~ REPRESENTAN AL MUTEX PROPIO DE LA CLASE LIGHTSWITCH() COMO SOLO ENTRA 1 PERSONA A LA VEZ AL LOBBY (MARCAO EN FOTOCOPIA) NO SE PUEDE SOBREPASAR I EN EL CONTADOR INTERNO SIN YA TENER LA PROPIEDAD DEL BAÑO. ~~DELOTRA~~ → SINO SE PODRÍA SOBREPASAR EL CONTADOR Y MEZCLAR GÉNEROS.

COMO VACÍO SOLO <sup>NIENDE</sup> SER LOCKEADO CUANDO HAY 1 PERSONA EN EL LOBBY Y NADIE DENTRO DEL BAÑO, ~~TA TAMBIEN PODRÍA DESBLOQUEAR EL BAÑO~~ Y COMO VACÍO SOLO PUEDE SER DESBLOQUEADO CUANDO NO HAY NADIE EN EL BAÑO (CONTADOR == 0) Y YO SOY LA



ÚNICA PERSONA EN EL LOBBY (RECORDAMOS QUE POR DEFINICIÓN SON ENTRA 1 PERSONA), ENTONCES ES IMPOSIBLE QUE HAYA DOS PERSONAS DE DISTINTO GÉNERO EN EL MISMO BAÑO.

DADO QUE NO HAY FUNCIÓN SALIR(), ASUMO QUE AL FINALIZAR LA EJECUCIÓN DE ENTRAR() LA PERSONA SALE DEL BAÑO.

EN ESTE CASO, AMBAS BARRERAS ESTÁN INICIALIZADAS CORRECTAMENTE CON CAPACIDAD MÁXIMA DE 3 Y SON "WAITERAS" Y "SEÑALADAS" ACORDEMENTE, Y ADEMAS, POR LO VISTO ANTES NO PUEDE HABER PERSONAS DE DISTINTO GÉNERO, ASÍ QUE PODRÍAMOS AFIRMAR QUE COMO MÁXIMO PUEDE HABER 3 PERSONAS EN EL BAÑO (ESTO ES, CLARAMENTE, SIN CONTAR LOS LOBBYS).

POR ÚLTIMO, ESTE PROCESO NO SATISFACE LA PROPIEDAD DE STARVATION-FREEDOM YA QUE ES CLARO VER QUE SIEMPRE Y CUANDO UN GÉNERO MANTENGA 1 PERSONA REPRESENTANTE DE SU GÉNERO DENTRO DEL BAÑO, EL GÉNERO CONTRARIO NO PODRÁ ACCEDER POR LA PROPIEDAD VISTA EN (A), MIENTRAS QUE EL GÉNERO CON CONTROL ACTUAL DEL BAÑO PODRÁ ~~INDE~~ CIRCULAR TRANQUILAMENTE.