

Segundo Examen Parcial - Solución

1. (Cap.7 - 20 puntos) Explique en forma detallada la forma en que funcionan tres mecanismos para PREVENIR la aparición de interbloqueos entre procesos de un sistema computacional. Adicionalmente, indique los inconvenientes de cada mecanismo. (Sugerencia: piense en los Backyardigans)

Para que se produzca un interbloqueo deben cumplirse las cuatro condiciones necesarias. Asegurando que una de las cuatro condiciones no se cumpla, podemos prevenir la aparición de interbloqueos.

- a. Eliminar exclusión mutua

La condición de exclusión mutua se aplica a los recursos que no pueden ser compartidos. Por el contrario, los recursos que sí pueden compartirse no requieren acceso mutuamente excluyente y, por tanto no pueden verse implicados en un interbloqueo. La ventaja aquí es que un proceso no necesita esperar nunca para acceder a un recurso compartible. Sin embargo, algunos recursos son intrínsecamente no compartibles.

- b. Eliminar retención y espera

Se debe garantizar que, cuando un proceso solicite un recurso, el proceso no esté reteniendo ningún otro recurso. Un posible protocolo consiste en exigir que cada proceso solicite todos sus recursos (y que los recursos se le asignen) antes de comenzar su ejecución. La desventaja es que la tasa de utilización de recursos puede ser baja, dado que los recursos pueden asignarse pero no utilizarse durante un período largo de tiempo. Una segunda desventaja es que puede producirse el fenómeno de inanición ya que un proceso que solicite varios recursos muy solicitados puede esperar de forma indefinida, debido a que al menos uno de los recursos que necesita está siempre asignado a otro proceso.

- c. Permitir el desalojo

Para impedir el desalojo, se puede usar el protocolo siguiente. Si un proceso está reteniendo varios recursos y solicita otro recurso que no se le puede asignar de forma inmediata, entonces todos los recursos actualmente retenidos se desalojan. El proceso sólo se reiniciará cuando pueda recuperar sus antiguos recursos, junto con los nuevos

que está solicitando. A menudo, este protocolo se aplica a tipos de recursos cuyo estado puede guardarse y restaurarse luego, como registros del CPU y el espacio en memoria. Sin embargo, generalmente este método no se puede aplicar a recursos como impresoras y unidades de cinta.

d. Eliminar espera circular

Una forma de garantizar que la condición de espera circular nunca se produzca es imponer una ordenación total de todos los tipos de recursos y requerir que cada proceso solicite sus recursos en orden creciente de numeración. Si un proceso solicita una instancia de un tipo de recurso menor al que ya tiene, debe liberar primero dichos recursos.

2. (Cap.8 - 20 puntos) Dadas cinco particiones de memoria de 100KB, 500KB, 200KB, 300KB y 600KB (en ese orden), ingresan procesos con solicitudes de memoria de A:212KB, B:417KB, C:112KB y D:426KB (en ese orden). Muestre cómo se ubicarán dichos procesos en memoria utilizando los siguientes algoritmos (nota: recuerde que los nuevos espacios generados también pueden ser utilizados).

a. Primer ajuste

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|      500      | 200 | 300 |      600      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|  A  | 288 | 200 | 300 |      600      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|  A  | 288 | 200 | 300 |      B      | 183 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|  A  | C | 176 | 200 | 300 |      B      | 183 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
* D no cabe

```

b. Mejor ajuste

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|      500      | 200 | 300 |      600      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|      500      | 200 |  A  | 88 |      600      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|      B      | 83 | 200 |  A  | 88 |      600      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|      B      | 83 | C | 88 |  A  | 88 |      600      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|100|      B      | 83 | C | 88 |  A  | 88 |      D      | 174 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

c. Peor ajuste

100	500	200	300		600	
100	500	200	300		A	388
100	B	83	200		300	A 388
100	B	83	200		300	A C 276

* D no cabe

d. Siguiente ajuste

100	500	200	300	600	
100	A	288	200	300	600
100	A	288	200	300	B 183
100	A	288	200	300	B C 71

3. (Cap.9 - 20 puntos) Para un sistema de memoria virtual basado en paginación, muestre cuándo y cuántos fallos de página se producirán para la siguiente cadena de referencias:

2 3 2 1 5 2 4 5 3 2 5 2

utilizando 3 marcos de página y mediante los siguientes algoritmos:

a. Óptimo

2 3 2 1 5 2 4 5 3 2 5 2

2 3 3 3 3 3 3 3 3 2 2 2

2 2 2 2 2 4 4 4 4 4 4

1 5 5 5 5 5 5 5 5

F F F F F F = 6 fallos

b. LRU (pila)

2 3 2 1 5 2 4 5 3 2 5 2

2 3 2 1 5 2 4 5 3 2 5 2

```

    2 3 2 1 5 2 4 5 3 2 5
      3 2 1 5 2 4 5 3 3
F F   F F   F   F F   = 7 fallos

```

c. FIFO

```

    2 3 2 1 5 2 4 5 3 2 5 2

    2 2 2 2 3 1 5 5 2 2 4 3
      3 3 3 1 5 2 2 4 4 3 5
        1 5 2 4 4 3 3 5 2
F F   F F F F   F   F F = 9 fallos

```

d. Reloj

```

    2 3 2 1 5 2 4 5 3 2 5 2

-2-2-2-2 3 1-5-5 2 2 4-3
  -3-3-3 1-5-2-2 4 4-3-5
    -1-5-2-4-4-3-3-5-2
F F   F F F F   F F F F = 10 fallos

```

4. (Cap.11 - 20 puntos) Muestre mediante los métodos de administración de espacio libre, que se indican mas abajo, la forma de llevar el registro de los siguientes bloques libres: 1,2,3,5,10,11,12,14,15,16,17,18,25,26,27 (los bloques van del 0 al 30).

a. Vector de bits

```

0111010000
1110111110
0000011100

```

b. Lista enlazada

```

+-----+
1 | 2 |
+-----+
2 | 3 |
+-----+
3 | 5 |
+-----+
+-----+
5 | 10 |
+-----+
+-----+
10 | 11 |

```

```

      +-----+
11 | 12 |
      +-----+
12 | 14 |
      +-----+
      +-----+
14 | 15 |
      +-----+
15 | 16 |
      +-----+
16 | 17 |
      +-----+
17 | 18 |
      +-----+
18 | 25 |
      +-----+
      +-----+
25 | 26 |
      +-----+
26 | 27 |
      +-----+
27 | X  |
      +-----+

```

c. Agrupamiento (Un bloque apunta a lo m·s a otros 5 bloques)

```

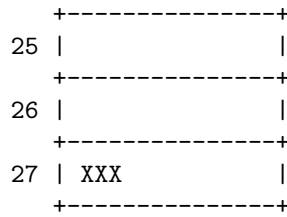
      +-----+
1 |  2,3,5,10,11 |
      +-----+
2 |               |
      +-----+
3 |               |
      +-----+
      +-----+
5 |               |
      +-----+
      +-----+
10 |              |
      +-----+
11 | 12,14,15,16,17|
      +-----+
12 |               |
      +-----+
      +-----+
14 |               |
      +-----+

```

15		
+-----+		
16		
+-----+		
17	18,25,26,27	
+-----+		
18		
+-----+		
+-----+		
25		
+-----+		
26		
+-----+		
27		
+-----+		

d. Recuento

+-----+		
1		
+-----+		
2		
+-----+		
3	5:1	
+-----+		
+-----+		
5	10:3	
+-----+		
+-----+		
10		
+-----+		
11		
+-----+		
12	14:5	
+-----+		
+-----+		
14		
+-----+		
15		
+-----+		
16		
+-----+		
17		
+-----+		
18	25:3	
+-----+		



5. (Cap.12 - 20 puntos) Suponga que una unidad de disco tiene 200 pistas, numeradas de 0 a 199. La unidad está sirviendo actualmente una solicitud en la pista 100. La cola de solicitudes pendientes, en orden FIFO, es: 55, 58, 39, 18, 90, 160, 150, 38, 184

Comenzando desde la posición actual del cabezal, determine la distancia promedio de posicionamiento (average seek length) que el brazo en disco tendrá que moverse para satisfacer todas las solicitudes pendientes para uno de los siguientes algoritmos de planificación de disco (muestre un diagrama para cada algoritmo):

- FIFO (primero en entrar, primero en salir)
- SSTF (primero el tiempo de servicio más corto)
- SCAN
- C-SCAN

(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	Average seek length	27.5	Average seek length	27.8	Average seek length	35.8

Figure 1: