

Sistemas Operativos

Boletín de Problemas **Gestión de Memoria**

Equipo "Sistemas Operativos DISCA/DSIC"

Universidad Politécnica de Valencia



1.- Notas y criterios para los problemas:

- Para el bit de validez de los descriptores de páginas el criterio seguido es: 1 página en memoria y 0 la página no está en memoria.
- Se utiliza indistintamente marco, trama o “frame” para referencia un fragmento lógico de igual tamaño que la página en la memoria física.
- En los problemas que impliquen asignación de memoria a los procesos y en los que no se diga nada sobre cómo se realice ésta, deberá suponerse que la memoria se encuentra inicialmente libre y que los marcos de la memoria física se van asignando en orden creciente, a medida que se vayan dando los fallos de página que precisen tales asignaciones.

2.- Problemas y cuestiones de asignación contigua

1. Sea un sistema gestionado con un mecanismo de particiones variables en el que la memoria física tiene 4200 palabras. En un instante la memoria está ocupada por 3 bloques de código/datos de la forma:

Dirección inicial	longitud
1000	1000
2900	500
3400	800

La estrategia utilizada cuando se carga un nuevo bloque en memoria es la del mejor ajuste en primer lugar. Si falla, se crea un hueco mayor desplazando los bloques en memoria hacia la dirección 0. Esta acción siempre empieza con el bloque actualmente en la dirección de memoria más baja, y prosigue únicamente hasta encontrar un hueco suficiente para el nuevo bloque.

A partir de ese momento, hay que cargar tres bloques de 500, 1200 y 200 (en ese orden). Describir el contenido de la memoria una vez satisfechas las peticiones.

2. Sea un sistema gestionado por particiones múltiples de tamaño variable con compactación. En un instante dado, se tiene la siguiente ocupación de la memoria:

P1 180K	libre 400 K	P2 100K	libre 150 K
-------------------	-----------------------	-------------------	-----------------------

Se utiliza la técnica del mejor ajuste. En la cola de trabajos tenemos en este orden: P4(120K), P5(200K) y P6(80K), los cuales deben ser atendidos en orden FIFO. Suponiendo que no finaliza ningún proceso y tras intentar cargar en memoria todos los procesos que están en la cola.

- Indicad cuántas particiones quedan libres y de qué tamaño son.
- Si en esta situación se aplica compactación, indicar qué proceso o procesos deberían moverse para que el número de Kbytes manejados fuese el menor posible y quede un único hueco.
- Si los registros base de cada proceso son, respectivamente, B1, B2, B3, B4, B5 y B6, indicar cómo han cambiado los registros base correspondientes al proceso o procesos que se han movido debido a la compactación.

- Sea un sistema gestionado por particiones múltiples de tamaño variable sin compactación. En un instante dado, se tiene la siguiente ocupación de la memoria:

0		1200K				
S.O. (80K)	P1 (180K)	libre (400K)	P2 (100K)	libre (150K)	P3 (90K)	Libre (200K)

- En la cola de trabajos tenemos en este orden: P4(120K), P5(200K) y P6(300K), los cuales deben ser atendidos en orden FIFO. Suponiendo que no finaliza ningún proceso y tras intentar cargar en memoria todos los procesos que están en la cola, evaluar cuál de las técnicas entre las de mejor ajuste y peor ajuste es conveniente utilizar y por qué.
- Partiendo de la situación inicial suponga que en la cola de trabajos se tiene P4(151K), P5(240K) y P6(200K), en ese orden, los cuales deben ser atendidos con criterio FIFO. Suponiendo que no finaliza ningún proceso y tras intentar cargar en memoria todos los procesos que están en la cola, evaluar cuál de las técnicas entre las de mejor ajuste y peor ajuste es conveniente utilizar y por qué.

3.- Problemas y cuestiones de asignación dispersa

- Suponga que un proceso emite una dirección lógica igual a 2453 y que se utiliza la técnica de paginación, con páginas de 1024 palabras.
 - Indique el par de valores (número de página, desplazamiento) que corresponde a dicha dirección.
 - ¿Es posible que dicha dirección lógica se traduzca en la dirección física 9322 ?. Razónelo.
- Considere los 4 sistemas con gestión de memoria mediante paginación A, B, C y D siguientes:

Sistema	A	B	C	D
Tamaño de página (en palabras)	512	512	1024	1024
Tamaño de palabra (en bits)	16	32	16	32

Asumiendo que sólo hay una tabla de páginas para todo el sistema y que tanto la dirección lógica como el descriptor de página ocupa una palabra (de 16 o 32 bits, según el caso), determine para cada sistema:

- a) El tamaño de la tabla de páginas (número de entradas).
 - b) El tamaño de la memoria lógica (número de páginas).
6. Sea un sistema de memoria segmentado-paginado. Los espacios de direcciones lógicas son de 8Gbytes y el tamaño de página es de 2Kbytes. Un espacio de direcciones puede tener hasta 256 segmentos y los descriptors de página tienen 4 bytes. Diga cual será el tamaño de la tabla de páginas de un segmento.
7. Suponga que un proceso emite las direcciones lógicas (2,18004) (1, 2735) (5, 12219) utilizando un modelo de gestión de memoria basado en segmentación paginada y que estas tres direcciones lógicas producen respectivamente las tres siguientes direcciones físicas: 1108, 1327 y 1211.
- a) ¿Cuál es el número máximo de bits que se están utilizando para el desplazamiento dentro de una página?
 - b) ¿Cuál es el número mínimo de bits que se están utilizando para indicar un número de página?
 - c) ¿Cuál es el número mínimo de bits que se están utilizando para indicar un número de marco?
8. A continuación se presentan varias situaciones en las que uno o más procesos generan varias direcciones lógicas. Indique las **direcciones físicas** correspondientes según cada esquema de gestión de memoria. Si no es posible indique ERROR.
- a) Considere un esquema de particiones variables. Las direcciones lógicas generadas son: (B,530), (A,0,130), (C,1046). Situación de los procesos:

Proceso	Registro base	Registro límite
A	0	1360
B	4020	6300
C	1400	2600

- b) Considere un esquema de paginación, con un tamaño de página de 256 palabras. Las direcciones lógicas generadas son 530, (0, 130), 1046. El contenido de la tabla de páginas es el siguiente:

Tabla de páginas

0	4
1	5
2	3
3	6

- c) Considere un esquema de segmentación paginada con un tamaño de página 128. Las direcciones lógicas generadas son 530, (0, 130), 1046. El contenido de la memoria física es el siguiente:

Nº de Marco (segmento, página)

0	(0, 3)
1	(1, 0)
2	(2, 0)
3	--
4	(1, 2)
5	(0, 1)
6	(0, 2)
7	(1,3)

9. En un esquema de segmentación paginada con páginas de 1Kb, ¿Es posible que la dirección lógica (2, 1333) se pudiera traducir a la dirección física 3654? ¿Y a la dirección física 2357? Razónelo.
10. Se tiene un sistema de memoria con paginación a dos niveles que permite agrupar las páginas en "directorios de páginas". Cada directorio de páginas puede contener 1024 páginas. Los espacios de direcciones lógicas de este sistema son de 4Gbytes, el tamaño de página es de 4Kbytes y el del descriptor de página es de 4 bytes. El espacio de direcciones físicas puede tener hasta 1Gb. Describa la estructura de las direcciones lógicas y de las direcciones físicas de este sistema de memoria virtual.
11. Un proceso P0 genera las siguientes direcciones lógicas: 612, 38 y (3,62). Indica las direcciones físicas correspondientes según cada esquema de gestión de memoria. Si no es posible indicar ERROR.
- a) Particiones Variables con Registro Base=150 y Registro Límite=220.
- b) Paginación, con un tamaño de página de 128 palabras. La tabla de

páginas del proceso es la siguiente:

Tabla de páginas

0	1
1	4
2	2
3	5

c) Segmentación. La tabla de segmentos del proceso es la siguiente:

Base Límite

0	200	20
1	50	10
2	105	49
3	320	70

d) Paginación, con un tamaño de página de 128 palabras. El sistema emplea tabla de páginas invertida. Además, en memoria está cargado otro proceso, llamado P1. La tabla de páginas invertida del sistema es la siguiente:

	Proceso	Página
0	P0	2
1	P1	0
2	P0	4
3	P1	1
4	P0	1
5	----	----
6	P0	0
7	P0	3

12. Suponga que un proceso emite la dirección lógica (2,18004) utilizando un modelo de gestión de memoria basado en segmentación y el espacio de memoria física es de 64K bytes.

a) ¿A qué direcciones físicas de las siguientes (11084, 33270 y 22112) sería posible traducir dicha dirección lógica?

b) ¿Cuál sería el resultado de traducir la dirección lógica (0,65536) en dicho sistema? Justificar la respuesta.

13. En un ordenador cuya MMU admite todos los modelos de asignación dispersa (es decir, paginación, segmentación y segmentación paginada), tres procesos P1, P2 y P3 han emitido algunas direcciones lógicas que la MMU ha traducido en sus direcciones físicas asociadas. Asuma que los tres procesos están en memoria y que no se utiliza memoria virtual ni intercambio. Indique qué tipo (o tipos) de gestión de memoria pueden utilizarse en ese sistema para que se haya podido dar esa traducción, cuando los tres procesos emitan la misma dirección lógica. Elija entre: segmentación, paginación, segmentación paginada e imposible (la traducción no puede llevarse a cabo suponiendo que los tres procesos están a la vez en memoria).

- a) Dirección lógica: (0, 5120) Direcciones físicas: P1: 6000, P2: 18000, P3: 40960
- b) Dirección lógica: (0, 17080) Direcciones físicas: P1: 4792, P2: 17080, P3: 33464

4.- Problemas y cuestiones de Memoria Virtual: Paginación por demanda.

14. Un determinado sistema operativo gestiona la memoria virtual mediante paginación por demanda. La dirección lógica tiene 24 bits, de los cuales 14 indican el número de página. La memoria física tiene 5 marcos. El algoritmo de reemplazo de páginas es el LRU LOCAL, y se ha implementado mediante un contador asociado a cada página que indica el instante de tiempo en que se referenció la página por última vez. Las tablas de páginas en el instante 16 son:

Tabla de páginas proceso A				Tabla de páginas proceso B			
	Marco	Bit de validez	Contador		Marco	Bit de validez	Contador
0	1	v	10	0	0	v	7
1	2	v	15	1	-	i	2
2	-	i	6	2	-	i	3
3	-	i	5	3	3	v	4
				4	4	v	11

Indique las direcciones físicas generadas para la siguiente secuencia de direcciones lógicas: (A, 2900) (B, 1200) (A, 1850) (A, 3072) (B, 527) (B, 2987) (A, 27) (A, 2000) (B, 4800) (B, 1500).

15. Un sistema de memoria virtual con paginación por demanda, tiene un tamaño de página de 512 palabras, una memoria virtual de 16 páginas numeradas del 0 al 15 una memoria física de 4 marcos (*frames*) numerados de 0 a 3. El contenido actual de la memoria es:

Nº marco

0	Pág. 4 del proceso P
1	Pág. 9 del proceso P
2	Pág. 5 del proceso P
3	Pág. 1 del proceso P

- a) Mostrar el contenido de la tabla de páginas.
- b) Suponiendo un algoritmo de reemplazo de páginas con estrategia óptima, mostrar el contenido de la tabla de páginas, tras generar cada una de las

siguientes direcciones lógicas: 1112, 1645, 2049, 622, 2776.

- c) Direcciones físicas equivalentes a las lógicas 1628, 851, 2700 y 2432.
- d) ¿Qué pasa cuando se referencia la dirección lógica 1330?
- e) Si la página cargada en el marco 3 es un procedimiento que otro proceso Q quiere compartir, ¿dónde debe aparecer en la tabla de páginas de Q?

- 16.** En un determinado sistema con memoria virtual con paginación por demanda, una dirección lógica consta de 16 bits, 10 de offset (desplazamiento) y 6 para el número de página. Se dispone de 4 marcos. Dada la siguiente secuencia de direcciones lógicas:

512 1102 2147 3245 5115 5200 4090 4207 1070 6200

7168 8200 7200 8300 9300 7410 8525 9700 5300 4387 1007

Se pide:

- a) Dar la serie de referencias a páginas.
 - b) Contar el número de fallos de página suponiendo los algoritmos de reemplazo FIFO, LRU y ÓPTIMO.
- 17.** Supongamos un sistema de gestión de memoria virtual basado en segmentación paginada, con un tamaño de página de 1000 palabras y una memoria principal de 3000 palabras. Las tablas de segmentos y las de página asociadas a cada segmento están ubicadas en registros (no ocupan espacio en memoria principal). Cada proceso consta de un espacio de direcciones lógicas dividido en 3 segmentos, uno para los datos (seg 0), otro para el código (seg 1) y el último para la pila (seg 2). En el sistema tenemos actualmente un solo proceso de las siguientes dimensiones (en palabras):

segmento	longitud
0	1300
1	2100
2	1000

Suponiendo que ha sido generada la siguiente secuencia de direcciones lógicas durante la ejecución del proceso en cuestión: (0, 0300) (0, 1200) (1, 0058) (0, 1112) (1, 1048) (2, 0354) (0, 1035) (0, 0036) (1, 2050) (1, 0128).

Calcular la correspondiente secuencia de direcciones físicas que se generará si se utiliza un algoritmo de reemplazo de páginas LRU.

- 18.** Un sistema de gestión de memoria por segmentación paginada posee las siguientes características:
- Una dirección lógica se compone de 12 bits.
 - Una dirección física se compone de 10 bits.

- El máximo número de segmentos en un programa es 4 (segmentos 0 a 3).
- Hay 4 marcos en memoria.
- Existen dos procesos A y B. El proceso A tiene dos segmentos (0 y 1), cuyos tamaños son 420 y 880 octetos, respectivamente. El proceso B tiene tres segmentos (0, 1 y 2) con tamaños respectivos iguales a 300, 530 y 190 octetos.

Se pide:

- ¿Cuál es el máximo número posible de páginas en un segmento?
- Suponiendo el formato (proceso, segmento, desplazamiento dentro del segmento), se genera la siguiente secuencia de referencias a memoria: (A, 1, 27) (B, 2, 48) (A, 0, 411) (B, 2, 132) (B, 1, 400) (A, 1, 428) (B, 0, 25) (B, 2, 177) (A, 1, 186) (B, 1, 280). La memoria está inicialmente vacía, y la política de reemplazo es FIFO con ámbito local. Determine a partir de la situación final cuál es la dirección física correspondiente a cada una de las siguientes direcciones lógicas: (A, 0, 350) (B, 2, 112) (A, 1, 422) (B, 0, 277) (A, 1, 604).

NOTAS:

- si la dirección es errónea, escriba ERROR.
- si está en disco, escriba DISCO.
- suponga que la memoria se llena de menor a mayor número de marco.

19. Sea un sistema con memoria virtual por segmentación paginada, con algoritmo de reemplazo óptimo local. La memoria principal está organizada en 4 marcos de 512 palabras cada uno. En un instante T_i únicamente tenemos a los procesos A y B en el sistema, y el S.O. mantiene la siguiente información:

Proceso A				Proceso B			
Tabla de segmentos		Tablas de páginas		Tabla de segmentos		Tablas de páginas	
tamaño	btp			tamaño	btp		
710	S0:0	S0:0	DISCO	100	S0:0	S0:0	1
200	S1:0	S1:0	DISCO	600	S1:0	S1:0	DISCO
1040	S2:0	S2:0	0	250	S2:0	S1:1	2
		S2:1	DISCO			S2:0	DISCO
		S2:2	3				
			DISCO				

NOTA.- btp (base de la tabla de páginas) guarda realmente la dirección física donde empieza realmente la tabla de páginas en la memoria. Sin embargo, para no hacer el enunciado demasiado largo, hemos dibujado las tres tablas de cada proceso de manera contigua, aunque realmente estén dispersas en la memoria principal. Tampoco hemos empleado direcciones físicas para representar las tablas, sino una indicación acerca de qué segmento está representado en cada entrada.

- Con esta situación, ¿qué porcentaje de memoria se desperdicia debido

a la fragmentación externa?; ¿y debido a la interna?.

- b) A partir de ese instante, continúa la ejecución concurrente de ambos procesos, y se generan las siguientes direcciones lógicas: (A, 0, 100) (A, 1, 25) (B, 2, 200) (A, 2, 1035) (B, 0, 10) (A, 0, 450) (B, 1, 580). Considere el instante T_j , que es el que ocurre tras generar las direcciones lógicas anteriores. Calcule, para cada una de las siguientes direcciones lógicas, la dirección física equivalente, considerando la situación del instante T_j : (A, 2, 1030) (B, 0, 400) (A, 0, 700) (B, 2, 100) (A, 0, 300) (B, 0, 92)

NOTA.- Si es una dirección errónea, conteste ERROR, y si es correcta pero la dirección equivalente no está en memoria física, escriba DISCO.

20. Dado un sistema con memoria virtual con paginación por demanda y el siguiente fragmento de código, con su correspondiente traducción a ensamblador:

```
FOR i := 1 TO 10 DO
  FOR j := 1 TO 10 DO
    C[i, j] := C[i, j] + A[i, j] + B[i, j];
```

En ensamblador

```
INICIO:      Inicializar R0 a 11
BUCLE_i:    Decrementar R0 una unidad
            Salto si cero a FIN_BUCLE_i
            Inicializar R1 a 11
BUCLE_j:    Decrementar R1 una unidad
            Salto si cero a FIN_BUCLE_j
            Cargar R3 con A[R0, R1]
            Cargar R4 con C[R0, R1]
            Cargar R5 con B[R0, R1]
            R3 <- R3 + R5
            R4 <- R4 + R3
            Grabar en C[R0, R1] el contenido de R4
            Salto a BUCLE_j
FIN_BUCLE_j: Salto a BUCLE_i
FIN_BUCLE_i: ...
```

Donde cada instrucción ocupa una palabra y R0 ... R5 son registros del procesador. Suponiendo que la paginación del programa ha quedado como:

MATRIZ C	página 0
MATRIZ A	página 1
MATRIZ B	página 2
Código y variables	página 3

- a) ¿Qué secuencia forman las páginas referenciadas en cada iteración del BUCLE_j?
- b) Suponiendo que cuando se vaya a ejecutar este fragmento de código (desde INICIO hasta FIN_BUCLE_i), la tabla de páginas contiene la

siguiente información:

Tabla de páginas	
0	1
1	-
2	2
3	0

y que se tienen únicamente 3 marcos en memoria, ¿cuántos fallos de página provocaría esta suma de matrices cuando se utiliza una política de reemplazo de páginas ÓPTIMO GLOBAL?. Calcúlese el valor de la tasa de fallos de página (p).¹

21. Sea un sistema con doble nivel de paginación donde las direcciones lógicas tienen 15 bits, el tamaño de página es de 1Kbyte y cada tabla de primer nivel puede contener cuatro descriptores de tablas de segundo nivel. Las tablas del primer y segundo nivel las gestiona el sistema operativo y por tanto no están en el área de memoria destinada a los procesos. Se tiene 4 marcos ubicados en las direcciones más bajas de la memoria física asignable a los procesos, dichos marcos se encuentran inicialmente vacíos, los marcos libres se asignan en orden creciente. Suponga que se emite la siguiente secuencia de direcciones lógicas: 7168, 26624, 16508, 5120, 7168, 26624, 12288, 7168, 26624, 16508, 5120, 12288. Esta secuencia se repite indefinidamente.

Diga qué página ocupará cada marco de la memoria física después de la primera repetición de la secuencia suponiendo que el algoritmo de reemplazo de páginas es el ÓPTIMO y utilizando la siguiente notación: (i, j) : identifica la página cuyo descriptor de primer nivel es el i , y cuyo descriptor de segundo nivel es el j .

22. Se tiene un sistema con memoria virtual que utiliza la técnica de la segmentación paginada por demanda. En este sistema se ejecuta un proceso A con tres segmentos T (código), D (datos) y S (pila). Los segmentos de datos y de pila pueden crecer. Las longitudes de estos segmentos se especifican en la tabla 1. La memoria física consta de 4 marcos de 512 bytes cuyo contenido en un instante t es el de la tabla 2, donde la nomenclatura T_i indica la página lógica i del segmento T (ej: D0 es la página 0 del segmento D).

Tabla-1	
segmento	Longitud
T	2500
D	1500

Tabla-2	
marco	Segmento,página
0	T2
1	D0

¹ La tasa de fallos de página indica el ratio entre el número de accesos que han provocado fallos de página y el número total de accesos a memoria realizados. Es decir: $p = \text{NFP} / \text{NTA}$. Donde NFP es el número de fallos de página y NTA es el número total de accesos.

S	300
---	-----

2	T1
3	S0

Si en ese instante t se produjese la referencia a memoria que se especifica en los siguientes casos, diga si se produciría fallo de página y, en caso afirmativo, el tipo de fallo (violación de protección, dirección fuera de rango, página en disco, ...) y el manejo o tratamiento que recibiría por parte del sistema operativo.

- a) Acceso a la dirección (D,1000) para escribir en una variable.
- b) Acceso a la dirección (T, 3510) para ejecutar una instrucción.
- c) Acceso a la dirección (D, 1510) para crear una variable dinámica.
- d) Acceso a la dirección (T,950) para ejecutar una instrucción.

23. Se tiene un sistema de gestión de memoria segmentado paginado con páginas cuya longitud es 1Kb. Este sistema dispone de una tabla de descriptores de página donde cada descriptor guarda, entre otros, la siguientes datos:

RO	LIST-SEG	etc.	Nº Marco
----	----------	------	----------

- RO: bit de protección frente escrituras. (1: sólo lectura, 0:lectura / escritura).
- LIST-SEG: Lista de segmentos que comparten la página.
- Nº Marco: Número de marco.

En este sistema cuando se produce un fallo de página no se aplica el algoritmo de reemplazo ya que un proceso del sistema ("demonio") se encarga de mantener una lista de marcos libres quitándoselos, cuando sea preciso, a los procesos ubicados en memoria. La lista de marcos libres que gestiona el demonio es: 15, 23, 40, 75, 90, 110, 125, 140, 142, 156, 200, 215 ...

Sobre el sistema anterior existe definida una operación de copia de segmentos

```
copiar(Id: Identificador_De_Segmento) return
    Identificador_De_Segmento
```

Esta operación toma como parámetro el identificador del segmento a copiar y devuelve el identificador de un nuevo segmento que es copia del anterior. La copia de segmentos se realiza utilizando la técnica de *copy-on-write* (copiar al escribir) consistente en lo siguiente: Cuando se invoca $s2 := \text{copiar}(s1)$ el sistema operativo crea un segmento $s2$ que *comparte* todas sus páginas con las del segmento $s1$. Sólo en el instante en que se intente realizar una operación de modificación de un segmento, se producirá un fallo de protección y el sistema operativo realizará una copia privada de la página que se desea

modificar para el segmento en el que va a producirse la modificación. El sistema operativo debe fijar los bits de protección de una página de tipo "copy-on-write" para que los intentos de escritura provoquen un fallo de protección. La rutina de tratamiento de fallo de protección debe ocuparse de realizar la copia de la página.

En el sistema de gestión de memoria descrito existe un segmento de 3500 bytes de longitud, que se representará como segmento A. Los descriptores de segmento en este sistema son una lista de referencias a una tabla de páginas global. De acuerdo con esto, el descriptor del segmento A y el estado de la tabla de páginas en el instante que se considera son:

Seg. A		Tabla de páginas			
		RO	Lista-seg	etc	Nº marco
0	→	0	A	...	10
1	→	1	A	...	11
2	→	2	A	...	12
3	→	3	A	...	13
		4	-	-	-

Se Pide: El estado de los descriptores de segmento implicados y de la tabla de descriptores de después de cada una de las siguientes operaciones:

- B := Copiar(A)
- Escribir en la dirección (A , 365)
- Escribir en la dirección (B , 1100)
- C := Copiar(B)
- Escribir en la dirección (B , 3000)
- Escribir en la dirección (C , 3000)
- Escribir en la dirección (A , 3000)

24. En un sistema de gestión de memoria virtual se decide utilizar paginación y un algoritmo de reemplazo LOCAL basado en una aproximación al LRU, denominada SEGUNDA OPORTUNIDAD. En el sistema se ejecutan actualmente 3 procesos A, B y C. Cada acceso se codifica de forma que el primer símbolo representa el proceso que lo realiza y el segundo la página accedida (se supone que todas las páginas accedidas están dentro del espacio lógico del proceso).

La secuencia de accesos a analizar es la siguiente: A1, A1, B0, C3, A0, A1, B0, B1, A3, B3, C0, C1, C1, A1, C0, A0, B0, A2, A2, A1, C0, C1, B1, C2, A3, C3, A4, C3, A2, B0, B0, C2, C3, A0, C2, C0, A4, C0, A1, A0.

Se pide:

- a) Determinar el contenido de la tabla de páginas asignada a cada

proceso, una vez que la memoria se ha llenado por primera vez. La memoria consta de 8 marcos que se asignan en orden de direcciones crecientes (desde el marco 0 al 7) e inicialmente están vacíos. Cada descriptor de página consta de dos campos: primero el **número de marco** asignado a la página y segundo el **bit de referencia** asociado. Si la página no está en memoria, ello se indica con un guión (-) en el número de marco correspondiente a dicha página.

- b) Determinar el contenido de los descriptors de página, tras realizar la secuencia completa de accesos propuesta e indicad aquellos accesos que en su momento produjeron fallos de página.
- c) A qué algoritmo de reemplazo equivale el de SEGUNDA OPORTUNIDAD, si todos los bits de referencia de la tabla de páginas están puestos a 1 en el momento previo al fallo de página.

25. Un sistema se define con las siguientes características:

- Utiliza la técnica de gestión de memoria virtual de paginación por demanda. El algoritmo de reemplazo es LRU GLOBAL, y se ha implementado mediante el uso de un contador que almacena el instante en que una página se referencia.
- Se utiliza un algoritmo de planificación del procesador por prioridades fijas expulsivas, y un algoritmo de planificación del dispositivo de paginación FCFS (por orden de llegada).
- Se dispone de 4 marcos de memoria.
- Durante la ejecución de un proceso, cada referencia a una página que está en memoria, consume una unidad de tiempo.
- Copiar una sola página al/del dispositivo de paginación desde/a memoria consume 1 unidad de tiempo. El uso del dispositivo de paginación a cargo de un proceso, durante los instantes de tiempo necesarios, se representará en la columna DISCO.

En $t=10$, hay en el sistema 3 procesos preparados A, B y C, con las siguientes características:

prioridad de A > prioridad de B > prioridad de C

- El contenido de la memoria en $t=10$ es:

Marco 0		Marco 1		Marco 2		Marco 3	
Página	Contador	Página	Contador	Página	Contador	Página	Contador
A0	5	C0	10	B0	9	-	-

- Secuencia de referencias a páginas durante la ejecución de cada proceso:

Proceso	Números de página
A	0, 0, 2, 3
B	0, 0
C	0, 0, 0, 1

Se pide rellenar la siguiente tabla en que se muestra la ocupación de los recursos del sistema en cada instante, hasta que finalice la ejecución de los procesos A, B y C:

t	Evento	Preparados	CPU	Cola DISCO	DISCO	Marco 0	Marco 1	Marco 2	Marco 3
11									

26. Un sistema se define con las siguientes características:

- Utiliza la técnica de gestión de memoria virtual de paginación por demanda. El algoritmo de reemplazo es FIFO GLOBAL, y se ha implementado mediante el uso de un contador que almacena el instante en que una página se carga en memoria.
- Se utiliza un algoritmo de planificación del procesador por prioridades fijas expulsivas, y un algoritmo de planificación del dispositivo de paginación FCFS (por orden de llegada).
- Se dispone de 4 marcos de memoria.
- Durante la ejecución de un proceso, cada referencia a una página que está en memoria, consume una unidad de tiempo.
- Copiar una sola página al/del dispositivo de paginación desde/a memoria consume 1 unidad de tiempo. El uso del dispositivo de paginación a cargo de un proceso, durante los instantes de tiempo necesarios, se representará en la columna DISCO.
- Se considerará que cualquier página víctima seleccionada por el algoritmo de reemplazo siempre habrá sido modificada, por lo que habrá que copiarla a disco.
- En $t=10$, hay en el sistema 3 procesos preparados A, B y C, con las siguientes características:

prioridad de A > prioridad de B > prioridad de C

- El contenido de la memoria en $t=10$ es:

Marco 0		Marco 1		Marco 2		Marco 3	
Página	Contador	Página	Contador	Página	Contador	Página	Contador

A0	9	C0	10	B0	5	-	-
----	---	----	----	----	---	---	---

- Secuencia de referencias a páginas durante la ejecución de cada proceso:

Proceso	Números de página
A	0, 0, 2, 3
B	0, 0
C	0, 0, 0, 1

Se pide rellenar la siguiente tabla en que se muestra la ocupación de los recursos del sistema en cada instante, hasta que finalice la ejecución de los procesos A, B y C:

t	Evento	Preparados	CPU	cola DISCO	DISCO	Marco 0	Marco 1	Marco 2	Marco 3
11									Libre

5.- Hiperpaginación y Área activa.

-Tanto en los enunciados como en las soluciones de los problemas de este apartado se emplean las siglas que pasamos a definir:

- TAA = Tamaño de área activa.

- 27.** En un sistema de gestión de memoria virtual se decide utilizar un modelo de área activa para controlar la demanda de memoria. En el sistema se ejecutan actualmente 3 procesos A, B y C. Cada acceso se codifica de forma que el primer símbolo representa el proceso que lo realiza y el segundo la página accedida (se supone que todas las páginas accedidas están dentro del espacio lógico del proceso).

Suponiendo que el tamaño de la ventana de área activa es 4, calcule el mínimo y máximo tamaño de área activa de cada proceso después de la siguiente secuencia de accesos y suponiendo que el muestreo se inicia inmediatamente antes del primer acceso a cada proceso: A1, A1, B0, C3, A0, A1, B0, B1, A3, B3, C0, C1, C1, A1, C0, A0, B0, A2, A2, A1, C0, C1, B1, C2, A3, C3, A4, C3, A2, B0, B0, C2, C3, A0, C2, C0, A4, C0, A1, A0, A0, C0.

- 28.** En un sistema operativo se ha decidido utilizar el modelo del área activa para controlar el número de páginas que deben residir en memoria por cada proceso. La estrategia utilizada es la siguiente:

- Cada vez que se realiza un acceso a memoria por parte de un proceso, su área activa es recalculada.
- En todo momento sólo deben estar presentes en memoria las páginas que forman las áreas activas de todos los procesos, siempre y cuando exista disponibilidad de memoria física.

- Si al recalcular el área activa de un proceso P, se detecta que no hay suficiente memoria física, este proceso P es eliminado completamente de memoria y es suspendido.
- Cuando se produzca nueva disponibilidad de memoria física, debe cargarse en memoria el área activa del primer proceso que se encuentre suspendido

En este sistema los parámetros que lo caracterizan son los siguientes:

- El número de referencias que se utilizan para calcular el área activa es 3
- Existen 8 marcos en memoria física

Inicialmente la memoria física esta vacía y llegan simultáneamente, pero en el orden indicado, cuatro procesos (A, B, C y D). En la siguiente tabla se indica la secuencia de accesos a **páginas** que realiza cada proceso dentro de su espacio lógico de direcciones. En el último acceso aparece la letra "T", la cual indica que el proceso termina, lo que implica que sus páginas deben ser eliminadas de memoria y debe atenderse la reanudación de un proceso suspendido si fuera necesario.

A	0	0	0	7	8	T
B	8	3	5	1	5	T
C	3	4	5	6	7	T
D	0	1	2	2	1	T

El sistema atiende a los procesos siguiendo un turno rotatorio, de tal forma que permite a un proceso emitir una dirección y luego se lo permite al siguiente. Ejemplo: A0, B8, C3, D0, A0, B3 ...

Complete la siguiente tabla de tal forma que describa para cada acceso a memoria en qué estado queda la memoria física tras dicho acceso, expresando para cada marco el identificador del proceso y el número de página que alberga.

Instante	Proceso Página	Marco 0	Marco 1	Marco 2	Marco 3	Marco 4	Marco 5	Marco 6	Marco 7

29. Dada la siguiente lista de referencias emitida por cierto proceso: ... 5 5 8 3 2 2 3 2 3 5 8 2 9 7 (instante₁) 7 9 2 2 9 10 9 10 9 10 9 10 2 (instante₂) ... si se utiliza el modelo del área activa con $\Delta=10$. Indique si las siguientes afirmaciones son verdaderas (V) o falsas (F) y justifique su respuesta.

- El tamaño del área activa en instante₁ es 10 y en instante₂, también.
- El área activa en instante₂ está formada por los marcos {2, 9, 10}.
- El área activa en instante₁ está formada por las páginas {2, 3, 5, 7, 8, 9}.

- d)** Con la información presentada no se puede saber qué páginas forman las áreas activas de este proceso en los instantes 1 y 2, deberíamos conocer también su localidad de referencia.
- e)** El tamaño del área activa en instante₂ es 4.
- f)** Si este proceso tiene 6 marcos asignados en el instante 1 a las páginas de su área activa (y no tiene otros marcos), no dará ningún fallo de página entre los instantes 1 y 2.

6.- Soluciones

6.1- Problemas y cuestiones de asignación contigua

En esta sección de soluciones únicamente se facilitarán los resultados finales de aquellas cuestiones y problemas que impliquen la utilización y desarrollo de los conceptos explicados en las clases de teoría. Las cuestiones que no necesiten la aplicación de fórmulas o algoritmos no se resolverán. Si el alumno no encontrara la respuesta a tales cuestiones en sus apuntes, ni en los libros recomendados como bibliografía, puede preguntar por ellas en el horario de consultas de su profesor de la asignatura.

1.

0000	0999	1000	1499	1500	2699	2900	3399	3400	4199
Bloque 1	Bloque 4	Bloque 5	Bloque 6	Bloque 2	Bloque 3				

2.

a) Quedan dos particiones de tamaños 120K y 30 K respectivamente.

P1 180K	P5 200K	P6 80K	libre 120K	P2 100K	P4 120 K	libre 30K
------------	------------	-----------	---------------	------------	-------------	--------------

- b) Debería moverse el proceso P4 al hueco de 120K, con lo cual quedaría un solo hueco de 150K.
- c) Todos los registros bases quedan igual que antes de la compactación exceptuando el del proceso P4 cuyo B4 contendrá 460 K (se ha considerado que P1 se encuentra ubicado en las direcciones más bajas de memoria).

3.

- a) Mejor ajuste.
- b) Peor ajuste.

6.2- Problemas y cuestiones de asignación dispersa

4.

- a) número página: 2 desplazamiento: 405
 b) No. No tendría idéntico desplazamiento.

5.

Sistema	A	B	C	D
Tamaño de página (en palabras)	512	512	1024	1024
Tamaño de palabra (en bits)	16	32	16	32
Tamaño de la tabla de páginas (en descriptores)	128	8 M	64	4M
Tamaño de memoria virtual (en páginas)	128	8M	64	4M

6. 64 KB.

7.

- a) 7 bits.
 b) 8 bits.
 c) 4 bits.

8.

a)

Direc. Lógica	(B,530)	(A,0,130)	(C, 1046)
Direc. Física	$4020+530=$ 4550	Error no vale este formato	$1400+1046=$ 2446

b)

Direc. Lógica	530	(0,130)	1046
Direc. Física	$3*256+18=$ 786	ERROR no vale este formato	ERROR (*)

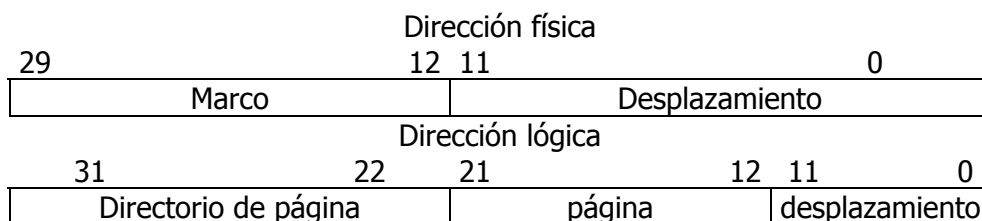
c)

Direc. Lógica	530	(0,130)	1046
Direc. Física	ERROR, no vale este formato	$5*128+2=$ 642	ERROR, no vale este formato

9.

- a) No
 b) Sí.

10.



11.

	Dirección Lógica 612	Dirección Lógica 38	Dirección Lógica (3, 62)
a) Dirección física	ERROR, excede el límite.	Base + 38 = 188	ERROR de formato de dirección
b) Dirección física (paginación)	ERROR, excede el nº de páginas del proceso.	$1 \cdot 128 + 38 = 166$	ERROR de formato de dirección
c) Dirección física (segmentación)	ERROR de formato de dirección	ERROR de formato de dirección	$320 + 62 = 382$
d) Dirección física (Tabla de páginas invertida)	Página 4, despl. 100 Cargado en marco 2 $2 \cdot 128 + 100 = 356$	Página 0, despl. 38 Cargado en marco 6 $6 \cdot 128 + 38 = 806$	ERROR de formato de dirección

12.

- a) 22112, 33270
- b) Error. Desplazamiento fuera de rango.

13.

- a) Segmentación.
- b) Segmentación paginada.

6.3- Problemas y cuestiones de memoria virtual: Paginación por demanda

14. 1876, 3248, 2874, 1024, 527, 5035, 2075, 2000, 3776, 476.

15.

a)

Tabla de páginas			Tabla de páginas		
	Marco	Bit de validez		Marco	Bit de validez
0	-	0	8	-	0
1	3	1	9	1	1
2	-	0	10	-	0
3	-	0	11	-	0
4	0	1	12	-	0
5	2	1	13	-	0
6	-	0	14	-	0
7	-	0	15	-	0

- b) Se empieza con la tabla anterior. El primer acceso la deja así:

Tabla de páginas			Tabla de páginas		
	Marco	Bit de validez		Marco	Bit de validez
0	-	0	8	-	0
1	3	1	9	1	0
2	1	1	10	-	0
3	-	0	11	-	0
4	0	1	12	-	0
5	2	1	13	-	0
6	-	0	14	-	0
7	-	0	15	-	0

Y el siguiente elige como víctima a la página 2 y mete en su lugar la 3. A partir de entonces ya no hay más fallos:

Tabla de páginas			Tabla de páginas		
	Marco	Bit de validez		Marco	Bit de validez
0	-	0	8	-	0
1	3	1	9	1	0
2	1	0	10	-	0
3	1	1	11	-	0
4	0	1	12	-	0
5	2	1	13	-	0
6	-	0	14	-	0
7	-	0	15	-	0

- c) 604, 1875, 1164, 384.
d) Se da un fallo de página.
e) En el descriptor asociado a su página 1.

16.

- a) 0, 1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 0.
b) FIFO: 14 LRU: 14 ÓPTIMO: 12

17. 300, 1200, 2058, 1112, 48, 2354, 1035, 36, 2050, 1128.

18.

- a) 4 páginas.
b) 606, 880, DISCO, DISCO, DISCO.

19.

- a) Por utilizar segmentación paginada, no puede haber fragmentación

externa. La interna será:

marco	Contenido de memoria	Fragmentación interna
0	Proceso A, seg 1, pág. 0	512-200=312 palabras
1	Proceso B, seg 0, pág. 0	412 palabras
2	Proceso B, seg 1, pág. 1	1024-600= 424 palabras
3	Proceso A, seg 2, pág. 0	0 palabras

b) 6, 912, DISCO, DISCO, 1836, 604.

20.

a) 3, 3, 3, 1, 3, 0, 3, 2, 3, 3, 3, 0, 3. La serie de referencias correspondiente es: 3, 1, 3, 0, 3, 2, 3, 0, 3.

b) Se dan 2 fallos en cada iteración, pero como hay 100 iteraciones, habrá 200 fallos en la suma. La tasa resultante es $p = 200 / 1363 = 0.14735$.

21. {(0, 7), (3, 2), (0, 5), (1, 4)}

22.

Referencia y Tipo de fallo	Manejo
(D,1000) → Fallo de página.	La página D1 debe cargarse en memoria reemplazando a una de las cuatro cargadas.
(T,3510)→ Dirección fuera de rango.	La dirección no está dentro del tamaño del segmento. El sistema abortará el proceso.
(D,1510) → Fallo de página.	Aunque la dirección está fuera del rango actual del segmento D, el sistema permite que crezca dicho segmento. La página D2 será asignada al proceso (puede que no se necesite cargarla desde disco si todavía no ha sido utilizada y pertenece al área de datos sin valores iniciales), reemplazando a una de las ya cargadas.
(T,950): Sin fallo.	Es una dirección de la página T1, ya presente en memoria. Por tanto, se atenderá la solicitud de dicha dirección

23.

1.- B := Copiar(A)

Seg.A	SegB	Seg. C
0	0	
1	1	
2	2	
3	3	

	RO	seg	marco
0	1	A B	10
1	1	A B	11
2	1	A B	12
3	1	A B	13
4			
5			
6			

	RO	seg	marco
7			
8			
9			
10			
11			
12			
13			

2.- Escribir en la dirección (A , 365)

Seg. A	Seg. B	Seg. C		RO	seg	marco		RO	seg	marco
4	0		0	0	B	10	7			
1	1		1	1	A B	11	8			
2	2		2	1	A B	12	9			
3	3		3	1	A B	13	10			
			4	0	A	15	11			
			5				12			
			6				13			

3.- Escribir en la dirección (B , 1100)

Seg. A	Seg. B	Seg. C		RO	Lista-seg	Nº marco		RO	Lista-seg	Nº marco
4	0		0	0	B	10	7			
1	5		1	0	A	11	8			
2	2		2	1	A B	12	9			
3	3		3	1	A B	13	10			
			4	0	A	15	11			
			5	0	B	23	12			
			6				13			

4.- C := Copiar(B)

Seg. A	Seg. B	Seg. C		RO	Lista-seg	Nº marco		RO	Lista-seg	Nº marco
4	0	0	0	1	B C	10	7			
1	5	5	1	0	A	11	8			
2	2	2	2	1	A B C	12	9			
3	3	3	3	1	A B C	13	10			
			4	0	A	15	11			
			5	1	B C	23	12			
			6				13			

5.- Escribir en la dirección (B , 3000)

Seg. A	Seg. B	Seg. C		RO	Lista-seg	Nº marco		RO	Lista-seg	Nº marco
4	0	0	0	1	B C	10	7			
1	5	5	1	0	A	11	8			
2	6	2	2	1	A C	12	9			
3	3	3	3	1	A B C	13	10			
			4	0	A	15	11			
			5	1	B C	23	12			
			6	0	B	40	13			

6.- Escribir en la dirección (C , 3000)

Seg. A	Seg. B	Seg. C		RO	Lista- seg	Nº marco		RO	Lista- seg	Nº marco
4	0	0	0	1	B C	10	7	0	C	75
1	5	5	1	0	A	11	8			
2	6	7	2	0	A	12	9			
3	3	3	3	1	A B C	13	10			
			4	0	A	15	11			
			5	1	B C	23	12			
			6	0	B	40	13			

7.- Escribir en la dirección (A , 3000)

Seg. A	Seg. B	Seg. C		RO	Lista- seg	Nº marco		RO	Lista- seg	Nº marco
4	0	0	0	1	B C	10	7	0	C	75
1	5	5	1	0	A	11	8			
2	6	7	2	0	A	12	9			
3	3	3	3	1	A B C	13	10			
			4	0	A	15	11			
			5	1	B C	23	12			
			6	0	B	40	13			

24.

a)

Memoria		Tabla Pag. A		Tabla Pag. B		Tabla Pag. C	
		Pg.	marco Bit ref.	marco Bit ref.	marco Bit ref.	marco Bit ref.	
0	A1	0	3 1	1 1	7 1		
1	B0	1	0 1	4 1	-		
2	C3	2	- -	- -	-		
3	A0	3	5 1	6 1	2 1		
4	B1	4	- -	- -	-		
5	A3	5	- -	- -	-		
6	B3	6	- -	- -	-		
7	C0						

b) Las páginas sombreadas son las que producen fallos de página, una vez la memoria se ha llenado por primera vez.

A1	A1	B0	C3	A0	A1	B0	B1	A3	B3	C0	C1	C1	A1
C0	A0	B0	A2	A2	A1	C0	C1	B1	C2	A3	C3	A4	C3
A2	B0	B0	C2	C3	A0	C2	C0	A4	C0	A1	A0.		

Memoria		Tabla Pag. A		Tabla Pag. B		Tabla Pag. C	
		Pg.	marco Bit ref.	marco Bit ref.	marco Bit ref.	marco Bit ref.	
0	A1	0	3 1	1 1	7 1		
1	B0						

2	C3	1	0	1	4	1	-	0
3	A0	2	-	0	-	-	-	0
4	B1	3	-	0	6	1	2	0
5	A4	4	5	1	-	-	-	
6	B3	5	-	-	-	-	-	
7	C0	6	-	-	-	-	-	

c) FIFO

25. Se presentan dos soluciones, considerando que solo se referencian las páginas y considerando que es necesario la actualización en disco.

Solución con sólo referencias no es necesario actualizar									
t	Evento	Prep	CPU	Cola DISCO	DISCO	Marco 0	Marco 1	Marco 2	Marco 3
11		B,C	A0			A0(11)	C0(10)	B0(9)	-
12		B,C	A0			A0(12)	C0(10)	B0(9)	-
13	Fallo de página A2	C	B0		A2	A0(12)	C0(10)	B0(13)	A2
14		B,C	A2			A0(12)	C0(10)	B0(13)	A2(14)
15	Fallo página A3, víctima marco1	C	B0		A3	A0(12)	A3	B0(15)	A2(14)
16	Fin B0, libera memoria	C	A3			A0(12)	A3(16)	-	A2(14)
17	Fin A3, libera memoria Fallo página C0		----		C0			C0	
18			C0					C0(18)	
19			C0					C0(19)	
20			C0					C0(20)	
21	Fallo de pagina C1		----		C1	C1		C0(20)	
22			C1			C1(22)		C0(20)	
23	Fin de C Libera memoria		-	-	-	-	-	-	-

Solución considerando que la página víctima siempre ha sido modificada.									
t	Evento	Prep	CPU	cola DISCO	DISCO	Marco 0	Marco 1	Marco 2	Marco 3
11		B,C	A0			A0(11)	C0(10)	B0(9)	-
12		B,C	A0			A0(12)	C0(10)	B0(9)	-

13	Fallo de página A2	C	B0		A2	A0(12)	C0(10)	B0(13)	A2
14		B,C	A2			A0(12)	C0(10)	B0(13)	A2(14)
15	Fallo de página A3 Victima marco-1, actualizar C0 en disco, por tanto A3 consume 2 u. de disco	C	B0		A3	A0(12)	C0	B0(13)	A2(14)
16	Fin B, libera memoria. Fallo de página de C0.		-	C0	A3	A0(12)	A3	C0	A2(14)
17			A3		C0	A0(12)	A3(17)	C0	A2(14)
18	Fin de A, libera memoria Se salva en disco A0		C0					C0(18)	
19			C0					C0(19)	
20			C0					C0(20)	
21	Fallo página C1		-		C1	C1	-	C0(20)	-
22			C1		-	C1(22)	-	C0(20)	-
23	Fin de C Se libera memoria		-		-	-	-	-	

26.

t	Evento	Prep	CPU	cola DISCO	DISCO	M0	M1	M2	M3
11	Procesos A,B y C preparados	B,C	A(0)	-	-	A0	C0	B0	Libre
12		B,C	A(0)	-	-	A0	C0	B0	Libre
13	Fallo pag. A2	C	B(0)		A(2)	A0	C0	B0	XX
14		B,C	A(2)	-	-	A0	C0	B0	A2
15	Fallo pag. A3, victima marco 2 (B0). Fallo pag. B0, victima marco 0, A0		C(0)	B(0,A0)	A(3,B0)	XX	C0	XX	A2
16			C(0)	B(0,A0)	A(3)	XX	C0	XX	A2
17		C	A(3)		B(0,A0)	XX	C0	A3	A2
18	Fin de A, libera memoria		C(0)		B(0)	XX	C0	libre	libre
19		C	B(0)			B0	C0	libre	libre
20	Fin de B, libera memoria		-		C(1)	XX	C0	libre	libre

	Fallo pág. C1, se elige marco 0								
21			C(1)		C(1)	XX	C0	libre	libre
22	Fin de C, libera memoria		-		libre	libre	libre	libre	libre

6.4- Problemas y cuestiones de Hiperpaginación

27. Proceso A: mínimo=2, máximo=4. Proceso B: mínimo=2, máximo=3.
Proceso C: mínimo=2, máximo=4.

28.

Instante	Proceso Página	Marco 0	Marco 1	Marco 2	Marco 3	Marco 4	Marco 5	Marco 6	Marco 7
0	A0	A0							
1	B8	A0	B8						
2	C3	A0	B8	C3					
3	D0	A0	B8	C3	D0				
4	A0	A0	B8	C3	D0				
5	B3	A0	B8	C3	D0	B3			
6	C4	A0	B8	C3	D0	B3	C4		
7	D1	A0	B8	C3	D0	B3	C4	D1	
8	A0	A0	B8	C3	D0	B3	C4	D1	
9	B5	A0	B8	C3	D0	B3	C4	D1	B5
10	D2	A0	B8	D2	D0	B3		D1	B5
11	A7	A0	B8	D2	D0	B3	A7	D1	B5
12	B1	A0	B1	D2	D0	B3	A7	D1	B5
13	D2	A0	B1	D2		B3	A7	D1	B5
14	A8	A0	B1	D2	A8	B3	A7	D1	B5
15	B5	A0	B1	D2	A8		A7	D1	B5
16	D1	A0	B1	D2	A8		A7	D1	B5
17	AT	C3	B1	D2	C4			D1	B5
18	BT	C3		D2	C4			D1	
19	DT	C3			C4				
20	C5	C3	C5		C4				
21	C6	C6	C5		C4				
22	C7	C6	C5		C7				
23	CT								

29. Todas falsas, excepto la “c” que es verdadera.

