

# SISTEMAS OPERATIVOS (2011-12)

## Grupo D

### Ejercicios - 1

1. ¿Cuál de las siguientes combinaciones no es factible? Justifíquelo detalladamente.
  - a) Procesamiento por lotes (batch) con multiprogramación.
  - b) Tiempo compartido sin multiprogramación.
  - c) Multiprogramación en un sistema monousuario.
2. ¿Qué debería hacer el planificador a corto plazo cuando es invocado pero no hay ningún proceso en la cola de ejecutables?
3. ¿Qué algoritmos de planificación quedan descartados para ser utilizados en sistemas de tiempo compartido?
4. La representación gráfica del cociente  $[(\text{tiempo\_en\_cola\_ejecutables} + \text{tiempo\_de\_CPU}) / \text{tiempo\_de\_CPU}]$  frente a  $\text{tiempo\_de\_CPU}$  suele mostrar valores muy altos para ráfagas muy cortas en casi todos los algoritmos de asignación de CPU. ¿Por qué?
5. Sea un sistema multiprogramado que utiliza el algoritmo Por Turnos (*Round-Robin*). Sea  $S$  el tiempo que tarda el despachador en cada cambio de contexto. ¿Cuál debe ser el valor de quantum  $Q$  para que el porcentaje de uso de la CPU por los procesos de usuario sea del 80%?
6. Sea un sistema multiprogramado que utiliza el algoritmo Por Turnos (*Round-Robin*). Sea  $S$  el tiempo que tarda el despachador en cada cambio de contexto, y  $N$  el número de procesos existente. ¿Cuál debe ser el valor de quantum  $Q$  para que se asegure que cada proceso "ve" la CPU al menos cada  $T$  segundos?
7. ¿Puede el procesador manejar una interrupción mientras esta ejecutando un proceso si la política de planificación que utilizamos es no apropiativa (sin desplazamiento)?
8. Suponga que es responsable de diseñar e implementar un sistema operativo que va a utilizar una política de planificación apropiativa (con desplazamiento) y que ya tenemos desarrollado el algoritmo de planificación sin desplazamiento ¿qué partes del sistema operativo habría que modificar para implementar la modalidad apropiativa y cuáles serían tales modificaciones?
9. En el algoritmo de planificación FCFS, la **penalización**  $((t + t^o \text{ de espera}) / t)$ , ¿es creciente, decreciente o constante respecto a  $t$  (tiempo de servicio de CPU requerido por un proceso)? Justifique su respuesta.
10. En la tabla siguiente se describen cinco procesos:

Proceso	Tiempo de creación	Tiempo de CPU
A	4	1
B	0	5
C	1	4
D	8	3
E	12	2

Si suponemos que tenemos un algoritmo de planificación que utiliza una política FIFO (primero en llegar, primero en ser servido), calcula:

- Tiempo medio de respuesta
- Tiempo medio de espera
- La penalización, es decir, el cociente entre el tiempo de respuesta y el tiempo de CPU.

11. Utilizando los valores de la tabla del problema anterior, calcula los tiempos medios de espera y respuesta para los siguientes algoritmos:

- Por Turnos con quantum  $q=1$
- Por Turnos con quantum  $q=4$
- El más corto primero (SJF). Suponga que se estima una ráfaga igual a la real.

12. Calcula el tiempo de espera medio para los procesos de la tabla utilizando el algoritmo: el primero más corto apropiativo (o primero el de tiempo restante menor, SRTF).

Proceso	Tiempo de creación	Tiempo de CPU
A	0	3
B	1	1
C	3	12
D	9	5
E	12	5

13. Utilizando la tabla del ejercicio anterior, dibuja el diagrama de ocupación de CPU para el caso de un sistema que utiliza un algoritmo de colas múltiples con realimentación con las siguientes colas:

Cola	Prioridad	Quantum
1	1	1
2	2	2
3	3	4

y suponiendo que:

- los procesos entran en la cola de mayor prioridad (menor valor numérico). Cada cola se gestiona mediante la política Por Turnos.
- la política de planificación entre colas es por prioridades no apropiativo.
- un proceso en la cola  $i$  pasa a la cola  $i+1$  si consume un quantum completo sin bloquearse.
- cuando un proceso llega a la cola de menor prioridad, permanece en ella hasta que finalice.

14. Consideremos los procesos cuyo comportamiento se recoge en la tabla siguiente

Proceso	Tiempo de creación	Comportamiento						
		CPU	Bloqueo	CPU	Bloqueo	CPU	Bloqueo	CPU
A	0	1	2	1	2	1	-	-
B	1	1	1	1	2	1	-	-
C	2	2	1	2	1	1	1	1
D	4	4	-	-	-	-	-	-

Dibuja el diagrama de ocupación de la CPU para los siguientes algoritmos:

- FIFO
- Por Turnos (Round-Robin), con  $q=1$
- Prioridades, suponiendo que las prioridades son 3 para A y B, 2 para C, y 1 para D (mayor número = menor prioridad).
- Primero el más corto, suponiendo que la estimación de cada ráfaga coincide con la duración de la ráfaga anterior. La estimación para la primera ráfaga es su valor real.

# SISTEMAS OPERATIVOS (2011-12)

## Grupo D

### Ejercicios – 2

1. Considere un sistema con un espacio lógico de memoria de 128K páginas con 8 KB cada una, una memoria física de 64 MB y direccionamiento al nivel de byte. ¿Cuántos bits hay en la dirección lógica? ¿Y en la física?

2. Sitúese en un sistema paginado con memoria virtual, en donde:

- la memoria real tiene un tamaño de 16 Mbytes
- una dirección virtual ocupa 32 bits, de los cuales los 22 de la izquierda constituyen el número de página, y los 10 de la derecha el desplazamiento dentro de la página.

Según lo anterior,

- ¿Qué tamaño tiene cada página?
- ¿Cuál es el tamaño del espacio de direccionamiento virtual?
- ¿En cuántos marcos de página se divide la memoria física?
- ¿Qué tamaño deberá tener el campo **Número de Marco** de la Tabla de Páginas?
- Además de dicho campo, suponga que la Tabla de Páginas tiene los siguientes campos con los siguientes valores:

\* Presencia: 1 bit (1= Presente en memoria física, 0= ausente)

\* Modificación: 1 bit (1= Ha sufrido modificación desde que se cargó en memoria)

\* Protección: 1 bit (1= Sólo se permite leer; 0= Cualquier tipo de acceso).

f) ¿Cuál es el tamaño de la Tabla de Páginas para un proceso cuyo espacio de memoria virtual es de 103K bytes?

3. Sea un sistema de memoria virtual paginada con direcciones lógicas de 32 bits que proporciona un espacio virtual de 220 páginas y con una memoria física de 32 Mbytes ¿cuánta memoria requiere en total un proceso que tenga 453Kbytes, incluida su tabla de páginas cuyas entradas son de 32 bits?

4. Un ordenador tiene 4 marcos de página. En la siguiente tabla se muestran: el tiempo de carga, el tiempo del último acceso y los bits R y M para cada página (los tiempos están en tics de reloj). Responda a las siguientes cuestiones justificando su respuesta.

Página	Tiempo de carga	Tiempo ultima Referencia	Bit de Referencia	Bit de Modificación
0	126	279	1	0
1	230	235	1	0
2	120	272	1	1
3	160	200	1	1

- ¿Qué página se sustituye si se usa el algoritmo FIFO?
- ¿Qué página se sustituye si se usa el algoritmo LRU?

5. ¿Depende el tamaño del conjunto de trabajo de un proceso directamente del tamaño del programa ejecutable asociado a él? Justifique su respuesta.

6. ¿Por qué una cache (o la TLB) que se accede con direcciones virtuales puede producir incoherencias y requiere que el sistema operativo la invalide en cada cambio de contexto y, en cambio, una que se accede con direcciones físicas no lo requiere?

7. Un ordenador proporciona un espacio de direccionamiento lógico (virtual) a cada proceso de 65.536 bytes de espacio dividido en páginas de 4096 bytes. Cierta programa tiene un tamaño de región de texto de 32768 bytes, un tamaño de región de datos de 16386 bytes y tamaño de región de pila de 15878. ¿Cabría este programa en el espacio de direcciones? (Una página no puede ser utilizada por regiones distintas). Si no es así, ¿cómo podríamos conseguirlo, dentro del esquema de paginación?

8. Analice qué puede ocurrir en un sistema que usa paginación por demanda si se recompila un programa mientras se está ejecutando. Proponga soluciones a los problemas que pueden surgir en esta situación.

9. Para cada uno de los siguientes campos de la tabla de páginas, se debe explicar si es la MMU o el sistema quién los lee y escribe (en éste último caso si se activa o desactiva), y en qué momentos:

- a) Número de marco.
- b) Bit de presencia
- c) Bit de protección
- d) Bit de modificación
- e) Bit de referencia

10. Suponga que la tabla de páginas para el proceso actual se parece a la de la figura. Todos los números son decimales, la numeración comienza en todos los casos desde cero, y todas las direcciones de memoria son direcciones en bytes. El tamaño de página es de 1024 bytes.

Número de página virtual	Bit de validez o presencia	Bit de referencia	Bit de modificación	Número de marco de página
0	0	1	0	4
1	1	1	1	7
2	1	0	0	1
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

¿Qué direcciones físicas, si existen, corresponderán con cada una de las siguientes direcciones virtuales? (no intente manejar ninguna falta de página, si las hubiese)

- a) 999
- b) 2121
- c) 5400

11. Sea la siguiente secuencia de números de página referenciados: 1,2,3,4,1,2,5,1,2,3,4,5

Calcula el número de faltas de página que se producen utilizando el algoritmo FIFO y considerando que el número de marcos de página de que disfruta nuestro proceso es de

- a) 3 marcos
- b) 4 marcos

¿Se corresponde esto con el comportamiento intuitivo de que disminuirá el número de faltas de página al aumentar el tamaño de memoria de que disfruta el proceso?

12. ¿Qué tipo de fragmentación se produce en un sistema de gestión de memoria virtual paginado? ¿Qué decisiones de diseño se pueden tomar para minimizar dicho problema, y cómo afectan estas decisiones al comportamiento del sistema?

13. Suponga que un proceso emite una dirección lógica igual a 2453 y que se utiliza la técnica de paginación, con páginas de 1024 palabras

a) Indique el par de valores (número de página, desplazamiento) que corresponde a dicha dirección.

b) ¿Es posible que dicha dirección lógica se traduzca en la dirección física 9322? Razónelo.

14. El tiempo medio de ejecución de una instrucción en un procesador es de 30 nsg. Tras diversas medidas se ha comprobado que:

a) El 0.001% de las instrucciones producen falta de página.

b) El 30% de las ocasiones en que se produce la falta de página, la página que hay que sustituir está "sucio".

c) La velocidad de transferencia al dispositivo de disco es de 2MB/sg. El tamaño de cada página es de 4 KB.

Calcule el tiempo efectivo de una instrucción (el  $t^o$  que tarda en ejecutarse).

15. Suponga que tenemos 3 procesos ejecutándose concurrentemente en un determinado instante y que todas sus páginas deben estar en memoria principal. El sistema operativo utiliza un sistema de memoria con paginación. Se dispone de una memoria física de 131072 bytes (128K). Sabemos que nuestros procesos al ser ejecutados tienen los siguientes parámetros:

Proceso	código	pila	datos
A	20480	14288	10240
B	16384	8200	8192
C	18432	13288	9216

Los datos indican el tamaño en bytes de cada uno de los segmentos que forman parte de la imagen del proceso. Sabiendo que una página no puede contener partes de dos segmentos diferentes (pila, código datos), hemos de determinar el tamaño de

o página que debería utilizar nuestro sistema y se barajan dos opciones: páginas de 4096 bytes (4K) o páginas de 512 bytes (1/2K). Se pide:

a) ¿Cuál sería la opción más apropiada, 4096 bytes o 512 bytes?. Justifica totalmente la respuesta mostrando todos los cálculos que has necesitado para llegar a dicha conclusión.

b) ¿Cuál es el formato de cada entrada de la Tabla de Páginas con el tamaño de página elegido? Justifica el tamaño de los campos con direcciones. Puedes añadir los bits que consideres necesarios para el buen funcionamiento del sistema indicando para que van a ser utilizados.

c) ¿Cuántas Tablas de Páginas habrá en este sistema? ¿Cuántas entradas hay en cada tabla de páginas (filas)?

16. En la gestión de memoria en un sistema paginado, ¿qué estructura/s de datos necesitará mantener el Sistema Operativo para administrar el espacio libre?

17. Situándonos en un sistema paginado, donde cada proceso tiene asignado un número fijo de marcos de páginas. Supongamos la siguiente situación: existe un proceso con 7 páginas y tiene asignados 5 marcos de página. Indica el contenido de la memoria después de cada referencia a una página si como algoritmo de sustitución de página utilizamos el LRU (la página no referenciada hace más tiempo). La secuencia de referencias es la indicada en la figura.

Referencias	2	1	3	4	1	5	6	4	5	7	4	2
Marcos de página												

¿Cuántas faltas de página se producen? \_\_\_\_\_

18. Supongamos que tenemos un proceso ejecutándose en un sistema paginado, con gestión de memoria basada en el algoritmo de sustitución **frecuencia de faltas de página**. El proceso tiene 5 páginas ( 0, 1, 2, 3, 4 ). Represente el contenido de la memoria real para ese proceso (es decir, indique que páginas tiene cargadas en cada momento) y cuándo se produce una falta de página. Suponga que, inicialmente, está cargada la página 2, el resto de páginas están en memoria secundaria y que no hay restricciones en cuanto al número de marcos de página disponibles. La cadena de referencias a página es: 0 3 1 1 1 3 4 4 2 2 4 0 0 0 0 3 y el parámetro es  $\tau=3$ .

19. Describa el funcionamiento del algoritmo de sustitución basado en la **frecuencia de faltas de página**, con los siguientes datos: 4 marcos de página, en  $t = 0$  la memoria contiene a la página 2. El tamaño de la ventana es  $\tau= 3$  y se produce la secuencia de referencias de páginas, 1 4 2 2 2 4 5 5 3 3 5 1 1 1 1 4

2																

20. Describa el funcionamiento del algoritmo de sustitución global basado en el **algoritmo basado en el modelo del conjunto de trabajo**, con los siguientes datos: 4 marcos de página, en  $t= 0$  la memoria contiene a la página 2 que se referenció en dicho instante de tiempo. El tamaño de la ventana es  $\tau= 3$  y se produce la secuencia de referencias de páginas, 1 4 4 4 2 4 1 1 3 3 5 5 5 5 1 4

2																

21. Una computadora con memoria virtual paginada tiene un bit U por página virtual, que se pone automáticamente a 1 cuando se realiza un acceso a la página. Existe una instrucción **limpiar\_U (dir\_base\_tabla)** que permite poner a 0 el conjunto de los bits U de todas las páginas de la tabla de páginas cuya dirección de comienzo pasamos como argumento. Explica cómo puede utilizarse este mecanismo para la implementación de un algoritmo de sustitución basado en el modelo del conjunto de trabajo.

22. Un Sistema Operativo con memoria virtual paginada tiene el mecanismo **fijar\_página(np)** cuyo efecto es proteger contra la sustitución al marco de página en que se ubica la página virtual **np**. El mecanismo **des\_fijar (np)** suprime esta protección.

- a) ¿Qué estructura/s de datos son necesarias para la realización de estos mecanismos?
- b) ¿En qué caso puede ser de utilidad estas primitivas?
- c) ¿Qué riesgos presentan y qué restricciones deben aportarse a su empleo?

23. Implemente la política de sustitución global basada en la medida de la tasa de faltas de página de un proceso; es decir, dé respuestas a las siguientes cuestiones:

- a) ¿Qué parte del Sistema Operativo deberá tomar parte?
- b) ¿Cuándo entra en ejecución dicho módulo del S.O.?
- c) ¿Qué estructuras de datos será necesario mantener?
- d) ¿Qué decisiones podría adoptar?

24. Disponemos de un ordenador que cuenta con las siguientes características: tiene una memoria RAM de 4KBytes, permite usar memoria virtual paginada, las páginas son de 1KBytes de tamaño y las direcciones virtuales son de 16 bits. El primer marco de página (marco 0) se usa únicamente por el Kernel y los demás marcos están disponibles para su uso por los procesos que se ejecutan en el sistema. Supongamos que tenemos sólo dos procesos, P1 y P2, y que utilizan las siguientes direcciones de memoria virtual y en el siguiente orden:

Proceso	Direcciones virtuales
P1	0-99
P2	0-500
P1	100-500
P2	501-1500
P1	3500-3700
P2	1501-2100
P1	501-600

- a) ¿Cuántos marcos de página tiene la memoria RAM de este ordenador?
- b) ¿Cuántos bits necesitamos para identificar los marcos de página?
- c) Describe los fallos de página que tendrán lugar para cada intervalo de ejecución de los procesos, si la política de sustitución de páginas utilizada es LRU. Suponga que se dicho algoritmo es de asignación variable y sustitución global.

25. Estamos trabajando con un sistema operativo que emplea una gestión de memoria paginada sin memoria virtual. Cada página tiene un tamaño de 2.048 bytes. La memoria física disponible para los procesos es de 8 MBytes. Suponga que primero llega un proceso que necesita 31.566 posiciones de memoria (o bytes) y, después, llega otro proceso que consume 18.432 posiciones cuando se carga en memoria. Se pide:

- a) ¿Qué fragmentación interna provoca cada proceso?
- b) ¿Qué fragmentación externa provoca cada proceso?

26. Suponga un sistema que utiliza paginación a dos niveles. Las direcciones son de 8 bits con la siguiente estructura: 2 bits en la tabla de páginas de primer nivel, 2 bits en la tabla de páginas de segundo nivel y 4 bits para el desplazamiento). El espacio de direccionamiento virtual de un proceso tiene la estructura del dibujo. Represente gráficamente las tablas de páginas y sus contenidos, suponiendo que cada entrada de la tabla de páginas ocupa 8 bits y que todas las páginas están cargadas en memoria principal (elige tú mismo la ubicación en memoria principal de dichas páginas,

Texto	0
Datos	16B
<div>sin usar</div>	48B
	224B
Pila	256B



RIE1 ¿Cuál de los sig. es falsa? a) Proc. por lotes con multiprogramación  
b) Tiempo compart. do sin mult.p., c) Mult.p. en un sist. monousuario

La b, el tiempo compartido consiste en compartir el tiempo de CPU entre varios usuarios "a la vez", para hacer esto el S.O. va cambiando de proceso en proceso de forma que a los usuarios les da la impresión de que todo el sistema está dedicado a ellos mismos. Para que el S.O. cambie de proceso es necesaria la mult. programación.

RIE2 ¿Qué debería hacer el planif. a corto plazo cuando es invocado pero no hay ningún proceso en la cola de ejecutables?

Suponiendo que no hay procesos a lp ni de S.O., el planificador a c/p llama a un proceso especial que ejecuta un bucle infinito sin instrucciones (llamado espera ocupada)

RIE3 ¿Qué alg. de planif. quedan descartados para ser usados en sistemas de tiempo compartido?

Todos los "no apropiativos" <sup>o los que no están basados en quantum</sup> (FCFS, SJF, planif. por prio. no apropiativo, \*

El motivo es que un proceso no puede monopolizar la CPU cuando hay varios usuarios en el sistema ejecutando procesos \*al SRTF Temporal

RIE5 Sistema con RR. S tiempo que tarda el despachador en cambiar de contexto, Q valor del Quantum ¿Valor de Q para que los procesos usen el 80% de CPU?

Queremos que Q ocupe el 80% de CPU, por tanto si tenemos

5 unidades de tiempo:  $\boxed{Q} \boxed{S}$  Queremos que Q ocupe 4 y S una

así pues  $Q = 4 \cdot S^*$   $+ \frac{\text{Punto por proceso}}{T. \text{ total}} \cdot 100 = \frac{Q}{Q+S} \cdot 100 = 80$

RIE6 Con N como n° de procesos ¿Valor de Q para asegurar que cada proceso "ve" a la CPU al menos cada T segundos?

$$Q = \frac{T - (N \cdot S)}{N} \quad T = N(Q + S) = NQ + NS$$

$$T - NS = NQ; \quad Q = \frac{T - NS}{N}$$

RIE7 ¿Puede el procesador manejar una interrupción mientras está ejecutando un proceso si la política de planificación es no apropiativa?

Si, el SO cambiaría al modo núcleo, trataría la interrupción y volvería con el proceso.

RIE8 Suponiendo que tenemos desarrollado un algoritmo de planif. expulsiva ¿Qué partes del SO hay que modificar para implementar el sistema?

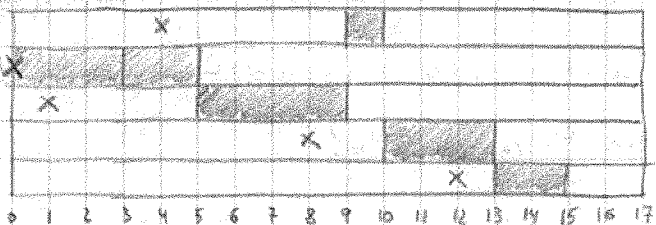
Habría que modificar el planificador a corto plazo, que ahora se tiene que encargarse de ir ordenando los procesos cuando llegan y de comprobar que si en la cola de listos hay algún proceso con mayor prioridad que el que está ejecutando, expulsar a éste para que ceda el de mayor prio.

RIE9 En FCFS, la penalización  $(\frac{t+t_{\text{espera}}}{t})$  ¿es creciente, decreciente o constante respecto a  $t$  (tiempo de CPU requerido por un proceso)?

La penalización  $(\frac{T_{\text{espera}}}{t})$  es decreciente, es decir cuanto más corto sea un proceso mayor penalización, esto es así porque los procesos cortos requieren poca CPU pero tienen que esperar mucho (sobre todo si delante tienen procesos largos)

RIE10 Obtener el tiempo medio de respuesta, de espera y la penalización por FCFS

PRO	T. Llegada	T. CPU	INI	SAL	T.E.	TR	PEN	
A	4	1	9	10	$9-4=5$	$1+5=6$	$\frac{6}{1}=6$	A
B	0	5	0	5	0	5	$\frac{5}{5}=1$	B
C	1	4	5	9	$5-1=4$	$4+4=8$	$\frac{8}{4}=2$	C
D	8	3	10	13	$10-8=2$	$3+2=5$	$\frac{5}{3}=1.66$	D
E	12	2	13	15	$13-12=1$	$2+1=3$	$\frac{3}{2}=1.5$	E
					16	27	12.466	



$$T. \text{ Respuesta} = T. \text{ CPU} + T. \text{ espera} = T. \text{ fin} - T. \text{ Llegada}$$

$$\text{Media } T. R = \frac{27}{5} = 5.4$$

$$T. \text{ Espera} = T. \text{ inicio} - T. \text{ Llegada}$$

$$\text{Media } T. E = \frac{16}{5} = 3.2$$

$$\text{Penalización} = \frac{T. \text{ CPU} + T. \text{ espera}}{T. \text{ CPU}} = \frac{T. \text{ Respuesta}}{T. \text{ CPU}}$$

$$\text{Media PEN} = \frac{12.466}{5} = 2.4932$$

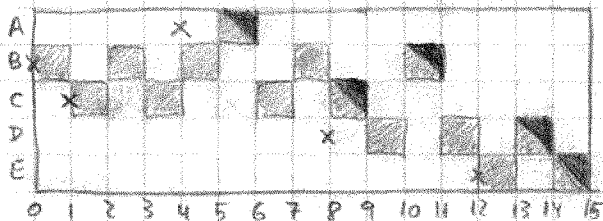
# RELACIÓN 1

EJERC 2

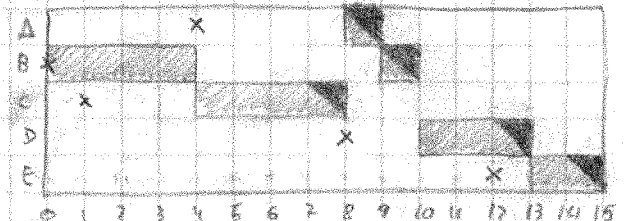
RIE II Igual que el anterior pero ahora con RR  $q=1$ ,  $q=4$  y SJF

PROCESO	T.LLEG	T.CPU	RR $Q=1$			RR $Q=4$			SJF		
			T.E	T.R	PEN	T.E	T.R	PEN	T.E	T.R	PEN
A	4	1	1	2	$\frac{3}{1}=2$	4	5	$\frac{5}{1}=5$	1	2	$\frac{3}{1}=2$
B	0	5	6	11	$\frac{11}{5}=2.2$	5	10	$\frac{10}{5}=2$	0	5	$\frac{5}{5}=1$
C	1	4	4	8	$\frac{8}{4}=2$	3	7	$\frac{7}{4}=1.75$	5	9	$\frac{9}{4}=2.25$
D	8	3	3	6	$\frac{6}{3}=2$	2	5	$\frac{5}{3}=1.66$	2	5	$\frac{5}{3}=1.66$
E	12	2	1	3	$\frac{3}{2}=1.5$	1	3	$\frac{3}{2}=1.5$	1	3	$\frac{3}{2}=1.5$
			15	30	9.7	15	30	11.91	9	24	8.41

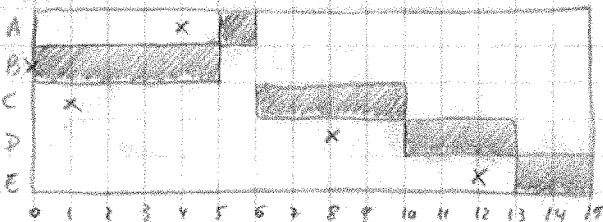
RR  $Q=1$



RR  $Q=4$



SJF



	RR $Q=1$	RR $Q=4$	SJF
Media TR	6	6	4.8
Media TE	3	3	1.8
Media PEN	1.94	2.382	1.692

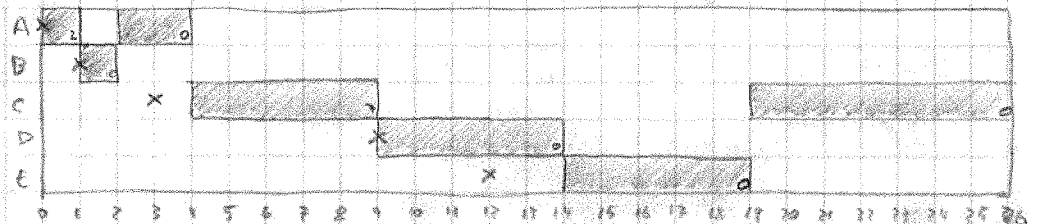
Para calcular T.E. contamos el T. que entra el proceso en la cola de listas

Para calcular T.R. sumamos T.E + T.CPU que es lo mismo que T. resp.

RIE 12 Calcular el T. espera medio usando SRTF (SJF exporpiativo)

P.	T.LLEG	T.CPU	T.E
A	0	3	1
B	1	1	0
C	3	12	11
D	9	5	0
E	12	5	2
			14

$$TE_{media} = \frac{14}{5} = 2.8$$



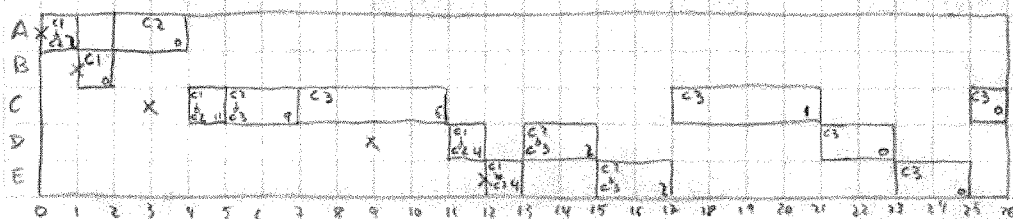
RIE 13 Dibujar el diagrama de CPU con alg. de colas multiples dado:

T.LLEG

P.	T.LLEG	T.CPU
A	0	3
B	1	1
C	3	12
D	9	5
E	12	5

COLA	PRIOR	QUANT
1	1	1
2	2	2
3	3	4

1 → mayor prio  
3 → menor prio

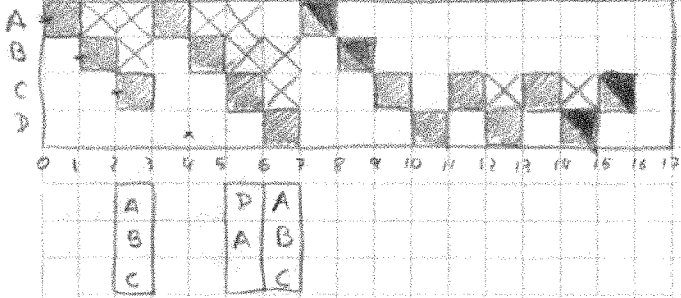


RIE14 Dibujar diagrama para FCFS, RR Q=1, Prioridad, SJF (estimación ráfaga = ref. anterior.)

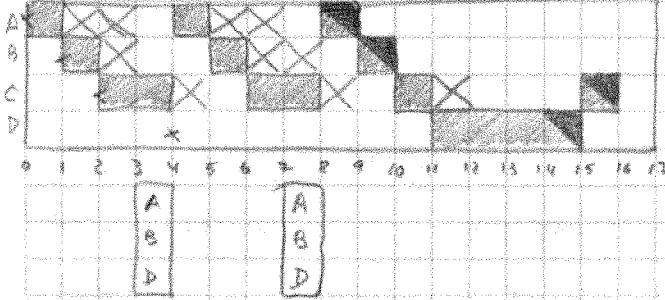
P.T. LLEG.	CPU	BLOQUE	C	B	C	B	C	PRIO
A	0	1	2	1	2	1	-	3
B	1	1	1	1	2	1	-	3
C	2	2	1	2	1	1	1	2
D	4	4	-	-	-	-	-	1

 → T.CPU  
 → BLOQUE  
 → Último CPU y fin

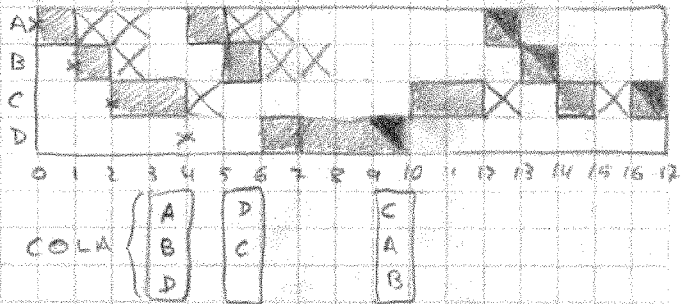
RR Q=1



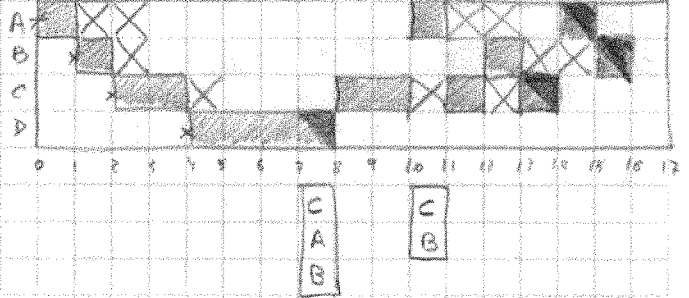
SJF



FCFS



PRIO



## Relación 2

Eje 3

R2E1 128K pág con 8KB cada una .64MB de mem

$$128000 \cdot 8KB = 2^{30} \text{ Bytes}$$

- hay  $2^{30}$  Bytes en la dir. lógica

- hay 26 bits en la dir. física

dir física:  $64MB \times 1024 \times 1024 = 67108864 \rightarrow \log_2(n) = 26$  // dir lógicas:  $128 \cdot 1024 \cdot 8 \cdot 1024 = 1073741824 \rightarrow \log_2(n) = 30$

R2E3 Proceso de 453K bytes, 1 fila de tabla de pág = 32 bits

$$n^{\circ} \text{ de pág de este proceso} = \frac{453 \cdot 1024 B}{\text{Tam pág}}$$

$$\text{Tam de pág} = \frac{2^{32} \text{ bits}}{220 \text{ pág}} =$$

\* ya q los dir. lóg son de 32 bits, dividimos todas las direcciones posibles entre el  $n^{\circ}$  de pág.

R2E10

a) 999

000 111 110 011 1

$\hookrightarrow n^{\circ} PV = 0 \Rightarrow$  pág no presente

b) 2121

010 000 100 100 1

$\hookrightarrow n^{\circ} PV = 2 \Rightarrow$  pág presente (marco 1) 001 000 100 100 1  $\Rightarrow 1097$

c) 5400

101 010 000 110 00

$\hookrightarrow n^{\circ} PV = 5 \Rightarrow$  pág presente (marco 0) 000 010 000 110 00  $\Rightarrow 280$

R2E9

a)  $N^{\circ}$  de marco: El SO le da valor cuando asigna un marco. La MMU usa el  $n^{\circ}$  de marco para generar la dirección física (+desplaz)

b) bit de presencia La MMU lo chequea en la traducción de direcciones. El SO lo chequea para saber qué partes están en MP. Además lo pone a 1 al cargar la pág en MP y a 0 al quitarla de MP

c) bit de protección La MMU lo chequea en la traducción de direcciones. El SO le da valor al crear/cargar la página a partir del ejecutable



d) bit de modificación El SO lo pone a 0 al cargar la página. La MMU lo pone a 1 al modificar alguna dirección dentro de la página. El SO lo pone a 0. Además, si es necesario para implementar algoritmos de gestión de memoria

R2.E8 Si se le da un nuevo contenido, puede mezclarse en memoria algunas partes antiguas y nueva y es un caché no descarteado.

Mientras se está mod un proc. el archivo no debe modif

R2.E2 MP: 16 MB. Dir Virtual: 32b, 22b n° pág y 10 desp.

n° pág	desp
22	10

a)  $2^{10} = 1024 B$

b)  $2^{32} = 4GB$  No sé qué pide ¿N° dir posibles? / Tamaño de todas las pág?

$2^{22} \text{ pág} \cdot 1024 B / \text{pág} = 4GB$  // Pregunta que pide el n° total de pág:  $2^{22} = 4M$

c)  $16 MB \cdot 1024 \cdot 1024 = 16777216 B$

$\frac{16 MB}{1024 B} = 16384 \text{ marcos}$

d)  $\log_2(16384) = 14 \text{ bits}$

f)

n° Marco	Pág	Mod	Prot
14b	1b	1b	1b

$103 KB / 1024 B = 103 \text{ marcos}$

$Tam = (14+3) \text{ bits} \cdot 103 \text{ marcos} = 1751 \text{ b} \approx 219 \text{ Bytes}$

R2.E3 M.V: 32b, 220K pág de espacio virtual. MP: 32 MB, Proceso con 453KB y T.P. de 32b. (Supongo direccionamiento por byte)

$\log_2(220 \cdot 1024) = 17.78 \Rightarrow \text{bits para pág } 18$

18	14
pág	desp

$Tam \text{ pág} = 2^{14} = 16384 B = 16 KB$

$n^\circ \text{ pág proceso} = \frac{453 KB}{16 KB} = 28.31 \Rightarrow 29 \text{ pág}$   $42812 = 116 B$

$Tam \text{ proc} = 453 KB + (29 \text{ pág} \cdot 32b) = 453 KB + 928 b = 463988 B$

R2E4

- a) Se sustituye la más antigua ( $T.C = 170$ ) que es la pág. 2  
 b) Se sustituye la que lleva más tiempo sin ref. ( $TR = 200$ ), la 3

R2E5

No, de lo que depende es del intervalo de tiempo y de las pág. referenciadas en él. Si un proceso puede tener muchas páginas pero si en el intervalo sólo referencia a una de ellas, el tamaño será de una página.

R2E7 Esp. virtual: 65536 B con pág. de 4096 B. Programa 3 secciones

Texto: 32768 B, datos: 16386 B, pila: 15878

$$n^{\circ} \text{ pág. virtual} = \frac{65536}{4096} = 16 \text{ pág.}$$

$$n^{\circ} \text{ pág. texto} = \frac{32768}{4096} = 8 \text{ pág.}; n^{\circ} \text{ pág. datos} = \frac{16386}{4096} = 4.0004 \Rightarrow 5 \text{ pág.}$$

$$n^{\circ} \text{ pág. pila} = \frac{15878}{4096} = 3.87 \Rightarrow 4 \text{ pág.}$$

$$n^{\circ} \text{ pág. prog.} = 8 + 5 + 4 = 17 \text{ pág.} \Rightarrow \text{el prog. no cabe en esp. fís.}$$

Ídea para hacer que el proc. entre en M.V. sin modif. esta Duple. espacio virtual o haciendo que los datos ocupen 16384 que son 4 pág.

R2E11 Secuencia: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5. No. de faltas con FIFO

a) 3 marcos:

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

 $\Rightarrow 9 \text{ faltas}$ 

b) 4 marcos:

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

 $\Rightarrow 10 \text{ faltas}$ 

Como vemos en el algoritmo FIFO al aumentar el n.º de marcos no tiene por qué reducirse el n.º de faltas.

R2E12 Con M.V. paginada se produce fragmentación interna para minimizar el problema se puede reducir el tamaño de página pero hacer eso aumenta el n° de páginas y por tanto el tamaño de la tabla de páginas y el n° de transferencias entre MP y disco

R2E13 dir lóg 2453 con paginación y 1024 palabras/pág.

$$2453 / 1024 = 2 \rightarrow \text{n° de pág}$$

$$2453 \% 1024 = 405 \rightarrow \text{desplazamiento}$$

$9322 \% 1024 = 106 \Rightarrow$  no es posible que la dir lóg. 2453 se traduzca a la dir física 9322 porque ambos tienen que tener el mismo desplazamiento y no es así

R2E16 MP: 128KB

Proc	código	pila	datos	código	pila	datos	código	pila	datos
A	20480	14288	10240	5	4	3	40	28	20
B	16384	8200	8147	4	3	2	32	12	16
C	18432	13288	9216	5	4	3	35	26	18

a)  $\frac{20480}{4096} = 5 \text{ pág (el resto igual)}$

Pág en MP con 4K =  $(5+4+3) + (4+3+2) + (5+4+3) = 33 \text{ pág} \Rightarrow 135168 \text{ Bytes}$

Pág en MD con 0.5K =  $233 \text{ pág} \Rightarrow 119296 \text{ Bytes}$

Tomamos pág de 512B pq con 4KB no caben en MP

R2E17 Proc con 7 pág y 5 marcos. LRU

2	1	3	4	1	5	6	4	5	7	4	2
2*	2	2	2	2	2	6*	6	6	6	6	6
	1*	1	1	1*	1	1	1	1	1	1	2*
		3*	3	3	3	3	3	3	7*	7	7
			4*	4	4	4	4*	4	4	4*	4
				5*	5	5	5*	5	5	5*	5
x	x	x	x		x	x		x		x	

Se producen 8 fallos de pág



# Relación 2

Eje 5

R1E18 Sistema pag, Algoritmo ~~TFF~~ <sup>modelo WS</sup>. Proceso con 5 pag. Inicialmente la pag 2 cargada. N° marcas limitadas. T=3 y ref: 0 3 1 1 1 3 4 4 2 2 4 0 0 0 0 3

$[t-v, t]$	$(-2, 1]$	$(-1, 2]$	$(0, 3]$	$(1, 4]$	$(2, 5]$	$(3, 6]$	$(4, 7]$	$(5, 8]$	$(6, 9]$	$(7, 10]$	$(8, 11]$	$(9, 12]$	$(10, 13]$	$(11, 14]$	$(12, 15]$	$(13, 16]$
WS	2	3,0	203	031	31	1	13	134	34	42	42	24	240	40	0	03
t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Secuencia		0	3	1	1	1	3	4	4	2	2	4	0	0	0	3
0		0	0	0									0	0	0	0
1				1	1	1	1	1								
2	2	2	2							2	2	2	2			3
3			3	3	3		3	3	3							
4								4	4	4	4	4	4	4		
4. pag	x	x	x				x	x		x			x			x

R2E19 FFP. T=23. Ref 1, 4, 2, 2, 2, 4, 5, 5, 3, 3, 5, 1, 1, 1, 1, 4. En t=0 2 en FP

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ref	1	4	2	2	2	4	5	5	3	3	5	1	1	1	1	4	
1	1	1	1	1	1	1						1	1	1	1	1	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
3									3	3	3	3	3	3	3	3	
4		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
5						5	5	5	5	5	5	5	5	5	5	5	
L. Pag	x	x					x	x	x			x				x	
t. en FP	0	0	1	2	3	4	0	0	0	1	2	0	1	2	3	0	

R2E24 RAM 4KB. Pag 1KB. Dir virt de 16b. 1<sup>er</sup> mapeo del kernel, resto proc

2	Proc	dir virt
P1	0-99	
P2	0-500	
P1	100-500	
P2	501-1500	
P1	3500-3700	
P2	1701-2100	
P1	501-600	

a)  $\frac{4KB}{1KB} = 4$  marcas de pag

b) Necesitamos 2 bits

P1: 0-600 y 3500-3700