

WUOLAH



irenegallegoskh

www.wuolah.com/student/irenegallegoskh



6985

Ejercicios-Tema-2-FS.pdf

Ejercicios Tema 2 FS



1º Fundamentos del Software



Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada**

Linguaskill 
from Cambridge

Ya puedes sacarte tu B1/B2/C1 de inglés desde casa

Demuestra tu nivel en 48 horas

Nuevo

#LinguaskillEnCasa

Ejercicios Tema 2

1. Dado un proceso en un SO con su información de contexto, de datos y de código según se muestra en la figura y que ya ha sido atendido en un 50% y le resta la otra mitad para finalizar su ejecución. Con la idea de optimizar el espacio de memoria para que el SO pudiera disponer de un mayor número de procesos en ésta, ¿podría reducirse el espacio que ocupa en memoria en alguna de las siguientes instancias?

- (a) La lista de procesos.
- (b) Información del contexto del proceso.
- (c) Tamaño de los datos.
- (d) Tamaño del código.

Código	12922 KB
Pila	3002 KB
Datos	434 KB

No se puede quitar nada.

2. ¿Por qué cuando un proceso está en modo “ejecutándose” y pretende acceder a una dirección de memoria fuera del área asignada, se informa de que se ha producido un error en la ejecución? ¿Quién informa de ello? Razone la respuesta.

Cuando un proceso intenta acceder a un SO, se aborta la ejecución.

3. ¿Tiene sentido un modelo de 5 estados de los procesos en un SO monousuario? Razone la respuesta.

No tiene sentidos, solo se ejecuta uno.

4. Dado un proceso que está en modo “ejecutándose” y pretende acceder a una dirección de memoria fuera del área asignada, lo cual sería un error en la ejecución, ¿a qué modo pasaría dicho proceso? Razone la respuesta.

- (a) Bloqueado.
- (b) No cambia de modo.
- (c) Finalizado.
- (d) Preparado.

Finalizado.

5. Un planificador de procesos tiene una tarea concreta dentro de un SO multiprogramado. ¿Tiene sentido disponer de un planificador de procesos en un SO monoprogramado? Razone la respuesta.

No, porque en un sistema monoprogramado solo habrá un proceso en ejecución.

6. Dado un SO multiprogramado, ¿bajo qué circunstancias se podría prescindir del planificador de procesos? Razone la respuesta.

Nunca se puede prescindir del planificador.

Un sistema multiprogramado permite que varios procesos se ejecuten al mismo tiempo, y el planificador permite repartir el tiempo de CPU entre los procesos que están preparados, por lo que si prescindimos del planificador el tiempo de CPU se descoordinaría del sistema.

7. Diga cuales de las siguientes operaciones pueden realizarse únicamente en modo supervisor, o modo kernel:

- a) Consultar la hora del sistema.
- b) Cambiar la fecha del sistema.
- c) Leer una pista/sector de un disco magnético.
- d) Generar una interrupción software.
- e) Generar una interrupción.
- f) Modificar la dirección de un vector de la tabla de vectores de interrupción.
- g) Deshabilitar las interrupciones.

Modificar la fecha del sistema, acceder a un disco magnético, modificar un vector y deshabilitar las interrupciones.

8. En el caso de un ordenador que se vaya a usar únicamente para un único usuario, ¿qué interés puede tener la existencia de los modos de funcionamiento supervisor/usuario?

En caso de que se produzca un error en modo usuario, el supervisor es un usuario privilegiado capaz de acceder al sistema en modo protegido.

9. Cuestiones sobre procesos, y asignación de CPU:

a) ¿Es necesario que lo último que haga todo proceso antes de finalizar sea una llamada al sistema para finalizar? ¿Sigue siendo esto cierto en sistemas monoprogramados?

No. No.

b) Cuando el controlador de un dispositivo produce una interrupción ¿se produce necesariamente un cambio de contexto?, ¿y cuando se produce una llamada al sistema?

No. No.

c) Cuando un proceso se bloquea, ¿deberá encargarse él directamente de cambiar el valor de su estado en el descriptor de proceso o PCB?

No.

PCB ---> conoce el estado del proceso. Para modificarlo lo tiene que hacer en modo kernel.

d) Sea un proceso que cambia de Ejecutándose a Bloqueado, ¿puede este cambio provocar un cambio de estado en otros procesos? Si es así, ¿en qué casos?

Sí. Si hay otro proceso listo, deja libre el procesador y el otro proceso pasará a ejecutarse.

e) Idem para el cambio de estado Bloqueado a Ejecutable.

No, el proceso se ejecutará cuando el planificador lo determine.

10. En los primeros ordenadores, cada byte de datos leído o escrito, era manejado directamente por la CPU (es decir, no existía DMA - Acceso Directo a Memoria). ¿Qué implicaciones tenía esta organización para la multiprogramación?

Que se priorizaban las operaciones E/S frente al resto y la memoria se quedaba ociosa.

11. ¿Por qué no es el intérprete de órdenes (shell) parte del propio sistema operativo? ¿Qué ventajas aporta el no serlo?

Porque no puede tener todos los permisos. Aporta seguridad al sistema.

12. Para cada una de las llamadas al sistema siguientes, especificar y explicar si su procesamiento por el sistema operativo implica un cambio de contexto:

a) Crear un proceso.

No obligatoriamente, a menos que el sistema operativo pase a ejecución el proceso creado, si no se ejecuta no hay cambio de contexto.

b) Abortar un proceso, es decir, terminarlo forzosamente.

Si se aborta el que está ejecutando sí hay cambio de contexto, si está en modo bloqueado no hay cambio de contexto.

c) Suspende o bloquear un proceso.

Sí.

d) Reanudar un proceso (inverso al caso anterior).

Sí, porque pasa al estado ejecutándose. No hay cambio si el proceso no pasa a ejecutarse.

e) Modificar la prioridad de un proceso.

Si cambias la prioridad para que se ejecute en el instante sí. No, no tiene porqué pasar a ejecutarse cambiándole la prioridad.

13. ¿Tiene sentido mantener ordenada por prioridades la cola de procesos bloqueados? Si lo tuviera, ¿en qué casos sería útil hacerlo?

Sí, cuando hay varios procesos esperando por un mismo recurso, por ejemplo para escribir en el disco.

14. ¿Por qué se utilizan potencias de dos para los tamaños de página, número de páginas en el espacio lógico de un proceso, y números de marcos de página?

Porque se trabaja en lenguaje binario.

15. Sitúese en un sistema paginado, en donde la memoria real tiene un tamaño de 16 Mbytes, una dirección lógica 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,

a) ¿Qué tamaño tiene cada página?

b) ¿En cuántos marcos de página se divide la memoria física?

c) ¿Qué tamaño deberá tener el campo Número de Marco de la Tabla de Páginas?

d) Además de dicho campo, suponga que la Tabla de Páginas tiene los siguientes campos con los siguientes valores: Protección: 1 bit (1= Sólo se permite leer; 0= Cualquier tipo de acceso).

¿Cuál es el tamaño de la Tabla de Páginas para un proceso cuyo espacio de memoria lógico es de 103Kbytes?



tamaño memoria = 16 MB
dir. pag → 32 bits
dirección lógica
nº pag = 22
desp.: 10

a) ¿Tamaño pag?

$$2^{10} = 1 \text{ KB}$$

b) ¿Marcos pag?

$$16 \text{ MB} = 2^4 \cdot 2^{20} \rightarrow \text{tamaño memoria}$$

$$\frac{2^4 \cdot 2^{20}}{2^{10}} = 2^4 \cdot 2^{10} = 2^{14} \text{ marcos}$$

tamaño pag

c) ¿Tamaño campo nº marcos?

$$14 \text{ bits}$$

d) ¿Tamaño tabla pag para un proceso de 103 KB?

$$\frac{15 \cdot 103}{1411} \rightarrow \text{el tamaño de pag es de 1 KB}$$

y el proceso tiene 103 KB

16. 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	Número de marco de página
0	4
1	7
2	1
3	2
4	10
5	0

¿Qué direcciones físicas corresponderán con cada una de las siguientes direcciones lógicas del proceso?

- a) 999
b) 2121
c) 5400

a) 999

$$\begin{array}{r} 999 \\ \underline{1024} \\ 999 \quad 0 \end{array}$$

pag 0, desp 999 → Dirección lógica
Dirección física = $4 \cdot 1024 + 999 = 5095$

b) 2121

$$\begin{array}{r} 2121 \\ \underline{1024} \\ 0073 \quad 2 \end{array}$$

pag 2, desp: 73 → Dirección lógica
Dirección física = $1 \cdot 1024 + 73 = 1097$

c) 5400

$$\begin{array}{r} 5400 \\ \underline{1024} \\ 0280 \quad 5 \end{array}$$

pag 5, desp: 280 → Dirección lógica
Dirección física = $0 \cdot 1024 + 280 = 280$

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

Fragmentación interna. Haciendo las páginas más pequeñas. El rendimiento se deteriora cuanto más pequeña sea la página. Cuanto más pequeño es el tamaño de página mayor es la tabla de página. Necesitaríamos más memoria para almacenar la misma información.

18. 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.

Dir-log = 2453
pág de 1024 palabras
" tamaño

a) N° pág, desp

b) ¿dir-física 9322?

a)
$$\begin{array}{r} 2453 \\ 405 \quad 2 \end{array}$$

N° pág = 2
Desp = 405

b) No

$9322 - 405 = 8917$

19. Suponga que tenemos 3 procesos ejecutándose concurrentemente en un determinado instante. 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 parámetros que se muestran en la tabla.

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 o datos), hemos de determinar el tamaño de 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)?

a) Opción 4K

$$\text{Proceso A: 12 páginas} \rightarrow \frac{20480}{4096} + \frac{14288}{4096} + \frac{10240}{4096} = 5 + 4 + 3 = 12$$

$$\text{Proceso B: 9 páginas} \rightarrow \frac{16384}{4096} + \frac{8200}{4096} + \frac{8192}{4096} = 4 + 3 + 2 = 9$$

$$\text{Proceso C: 12 páginas} \rightarrow \frac{18432}{4096} + \frac{13288}{4096} + \frac{9216}{4096} = 6 + 3 + 3 = 12$$

Total: 33 páginas \times 4096 = 135168 bytes \rightarrow excedemos la memoria disponible \rightarrow no es posible paginar en 4K
(memoria física disponible 131072)

Opción 1/2 K:

$$\text{Proceso A: 88 páginas} \rightarrow \frac{20480}{512} + \frac{14288}{512} + \frac{10240}{512} = 40 + 28 + 20 = 88$$

$$\text{Proceso B: 65 páginas} \rightarrow \frac{16384}{512} + \frac{8200}{512} + \frac{8192}{512} = 32 + 17 + 16 = 65$$

$$\text{Proceso C: 80 páginas} \rightarrow \frac{18432}{512} + \frac{13288}{512} + \frac{9216}{512} = 36 + 26 + 18 = 80$$

Total: 233 páginas \times 512 = 119296 bytes \rightarrow no sobrepasa la memoria física y por tanto es la opción más apropiada.

b) Necesitamos:

- 1 bit de protección.
- 2 bits para distinguir tipo de página
- $131072B / 512B$ cada marco = 256 marcos
 $\rightarrow 256 = 2^8 \rightarrow$ 8 bits para direccionar marcos.

Cada fila de la tabla de páginas necesita: $1 + 2 + 8 = 11$ bits

c) Habrá 3 tablas de páginas, 1 por cada proceso.

Necesitamos una entrada por cada página como mínimo:

Proceso A: 88 entradas

Proceso B: 65 entradas

Proceso C: 80 entradas

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

Una tabla de página.

21. Estamos trabajando con un sistema operativo que emplea una gestión de memoria paginada. 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 calcular la fragmentación interna provocada en cada proceso.

Primer proceso:

$$\begin{array}{r} 31566 \\ 11086 \\ 0846 \end{array} \begin{array}{l} \underline{12048} \\ 15 \rightarrow \text{páginas} \\ \text{completas} \end{array}$$

Fragmentación:

$$2048 - 846 = 1202 \text{ B}$$

Segundo proceso:

$$\begin{array}{r} 18432 \\ 09 \end{array} \begin{array}{l} \underline{12048} \\ 9 \rightarrow \text{páginas completas} \end{array}$$

No se producirá fragmentación.

22. Considere la siguiente tabla de segmentos:

¿Qué direcciones físicas corresponden a las direcciones lógicas (nº segmento, desplazamiento) siguientes? Si no puede traducir alguna dirección lógica a física, explique el por qué.

a) 0, 430

b) 1, 10

c) 3, 400

d) 4, 112

a) 0, 430

$$219 + 430 = 649$$

b) 1, 10

$$2300 + 10 = 2310$$

c) 3, 400

$$1327 + 400 = 1727$$

d) 4, 112

$112 > 96 \rightarrow \text{desplazamiento} > \text{longitud}$

Segmento	dirección base	longitud
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

23. ¿Qué cambio de contexto tardará menos y por qué?

a) El producido entre dos hebras del mismo proceso.

b) El producido entre dos hebras de distintos procesos.

El producido entre dos hebras del mismo proceso, porque comparte la información del proceso.