

WUOLAH



josemhv

www.wuolah.com/student/josemhv



4549

Tema2_FS_Teoria.pdf'.pdf

Tema 2 Teoria FS Ejercicios Resueltos



1º Fundamentos del Software



Grado en Ingeniería Informática



UGR - Universidad de Granada

TEMA 2 :: FUNDAMENTOS DEL SOFTWARE

- 1º Relación de ejercicios.
- 2º Teoría.

Relación de ejercicios:

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.
- ☒ Tamaño de los datos.
- ☒ Tamaño del código.

Código	12922 KB
Pila	3002 KB
Datos	434 KB

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.

Porque puede escribir en otra celda de memoria que no corresponde al proceso y provocar un fallo en el sistema.

Informa el gestor de memoria del sistema operativo.

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

Si porque un único usuario puede ejecutar varios procesos a la vez y cada proceso requiere memoria.

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.

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

Las 2 están bien. Depende de lo que vaya a hacer el proceso puede finalizarse o bloquearse. Si se bloquea, es porque en algún momento se dará la circunstancia necesaria para su reanudación. Por ejemplo si a un proceso se le asigna una dirección de memoria que no existe, se finalizará porque no habrá forma de reanudarlo si se bloquea. Cuando direccionamos algo y no está en memoria principal, el proceso se bloquea hasta que llegue el espacio de direcciones que falta.

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.

El planificador de procesos lo que hace es repartir el tiempo disponible de un microprocesador entre todos los procesos que están disponibles para su ejecución.

NO TIENE SENTIDO, "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.

No se puede prescindir nunca del planificador.

"Un sistema multiprogramado permite que varios procesos se ejecuten "al mismo tiempo" y el planificador es el sistema que permite repartir el tiempo de cpu entre los procesos que están preparados; si prescindiéramos del planificador, el tiempo de cpu se descoordinaría el 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.

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 un 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? ¿Siguiendo esto cierto en sistemas monoprogramados?

Si, es la manera de decirle al sistema operativo de que tiene que quitar ese programa de la memoria para cargar otro o cualquier otra cosa. en un monoprogramado también tiene sentido por lo mismo, (monoprogramado es que se ejecuta un programa y hasta que no termina no comienza otro) .

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?

Contexto: el estado del proceso.

No, no tiene porque, ejemplo el reloj, se genera una interrupción pero no cambia necesariamente de estado.

No necesariamente, hay llamadas al sistema que generan interrupción y que 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?

PCB o descriptor de procesos es lo que usa el SO para conocer el estado de cada proceso, para modificarlo lo tiene que hacer en modo kernel, no lo puede hacer el propio proceso.

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?

Si, 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. no afecta a ningún otro proceso. Si ese proceso pasa de bloqueado a ejecutarse es porque la CPU no está ocupada con otro proceso (libre).

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?

El acceso directo a memoria permite liberar al procesador de mucho procesamiento, el tiempo que se ahorra se puede invertir en otros procesos.

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

Porque podría ejecutar órdenes en modo kernel lo cual podría ser peligroso.

La ventaja es que todos los cambios a realizar deben ser a través de llamadas al sistema el cual evalúa si la orden es segura o no.

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(Estado).

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

Si el que vas a abortar se está ejecutando, si hay un cambio de contexto, si está en modo bloqueado(por ejemplo) no hay cambio de contexto.



CURSOS DE INGLÉS EN EL EXTRANJERO

TU FUTURO NO TENDRÁ LÍMITES

DESCARGA
EL CATÁLOGO
GRATUITO

KAPLAN
INTERNATIONAL
ENGLISH

KAPLANINTERNATIONAL.COM/ES

✓ 41 ESCUELAS
ALREDEDOR
DEL MUNDO

✓ 80 AÑOS DE
EXPERIENCIA

✓ TODOS LOS
NIVELES Y
OBJETIVOS

DESCARGA
EL CATÁLOGO
GRATUITO



KAPLANINTERNATIONAL.COM/ES

KAPLAN
INTERNATIONAL
ENGLISH

c) Suspender o bloquear un proceso.

Si, se supone que se suspenden o bloquean los procesos en ejecución.

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

Hay cambio de contexto, porque pasa al estado ejecutándose. No hay cambio de contexto si el proceso no pasa a ejecutarse (se pone en cola por ejemplo).

e) Modificar la prioridad de un proceso.

No, no tiene porqué pasar a ejecutarse cambiándole la prioridad. si le cambias la prioridad para que se ejecute en el instante si hay cambio de contexto.

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

Si, 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?

Como las direcciones están expresadas en binario, es mas fácil separar el segmento del desplazamiento. El esquema de direccionamiento lógico es transparente al programador, al ensamblador y al montador. Cada dirección lógica de un programa es idéntica a su dirección relativa.

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?

2^{10} porque el tamaño de página viene determinado por el desplazamiento.

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

$$16\text{MB}/2^{10} = 2^4 * 2^{20}/2^{10} = 2^{14}$$

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

$$\log_2 2^{14} = 14 \text{ bits.}$$

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 103K bytes?

$$103\text{KB}/1\text{KB} = 103 \text{ caracteres.}$$

$$103 * (14+1) = 1545.$$

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

$$999/1024 = 0; \text{ resto } 999$$

$$\text{Dirección física} = \text{Número de marco} * \text{Tamaño página} + \text{desplazamiento(resto)}$$

$$\text{Dirección física} = 4 * 1024 + 999 = 5095$$

b) 2121

$$2121/1024 = 2; \text{ resto } 73$$

$$\text{Dirección física} = 1 * 1024 + 73 = 1097$$

c) 5400

$$5400/1024 = 5; \text{ resto } 280$$

$$\text{Dirección física} = 0 * 1024 + 280 = 280$$

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?

En paginado es interna.

Haciendo las páginas mas pequeñas. El rendimiento se deteriora cuanto mas pequeña sea la página cuanto mas pequeño es el tamaño de página, mayor es la tabla de página. necesitaríamos mas 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.

$$2453/1024 = 2 \rightarrow \text{Número de páginas}$$

$$\text{resto } 405 \rightarrow \text{desplazamiento.}$$

$$(2, 405)$$

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

$9322 - 405 = 8917 \rightarrow$ n° de marco de página

No es posible ya que el n° de marco de página es impar y por temas de eficiencia siempre son numero pares , es decir, múltiplos de dos.

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.

OPCIÓN DE 4K: dividimos el tamaño de cada uno de los elementos de la imagen del proceso entre el tamaño de página :

- Proceso A: 12 páginas
- Proceso B: 9 páginas
- Proceso C: 12 páginas

TOTAL: $33 \text{ páginas} * 4096 = 135168$ bytes de memoria necesitamos; excedemos la memoria disponible por lo tanto no es posible paginar en 4K

OPCIÓN DE 1/2K:

- Proceso A: 88 páginas
- Proceso B: 65 páginas
- Proceso C: 80 páginas

TOTAL: 119296 bytes de memoria .

La mejor opción sería paginar en 1/2K. "El problema de paginar en páginas mas pequeñas es que se reduce el rendimiento del equipo".

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.

Necesitamos:

- 1bit de protección.
- Tamaño de página = $512 = 2^9 = 9$ bits de tamaño de página .
- $31072 \text{ B} / 512 \text{ B}$ cada marco = 256 marcos. $256 = 2^8 = 8$ bits para direccionar marcos.

Cada fila de la tabla de páginas necesita $8 + 9 + 1 = 18$ bits.

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

Habrá tres tablas de página , una por cada proceso.

Necesitamos una entrada por cada página como mínimo. por tanto:

- 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: $31566 \text{ bytes} / 2048 \text{ bytes} = 15$ páginas completas ; resto 846

La fragmentación será de $2048 - 846 = 1202 \text{ B}$.

Segundo proceso: $18432 \text{ bytes} / 2048 \text{ bytes} = 9$ páginas completas; resto 0

No se produciría fragmentación.



CURSOS DE INGLÉS EN EL EXTRANJERO

TU FUTURO NO TENDRÁ LÍMITES

DESCARGA
EL CATÁLOGO
GRATUITO

KAPLAN
INTERNATIONAL
ENGLISH

KAPLANINTERNATIONAL.COM/ES

41 ESCUELAS
ALREDEDOR
DEL MUNDO

80 AÑOS DE
EXPERIENCIA

TODOS LOS
NIVELES Y
OBJETIVOS

DESCARGA
EL CATÁLOGO
GRATUITO



KAPLANINTERNATIONAL.COM/ES

KAPLAN
INTERNATIONAL
ENGLISH

22. Considere la siguiente tabla de segmentos:

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

¿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

Comprobar si el desplazamiento es menor que la longitud:

$$430 < 600 \text{ SÍ}$$

$$\text{Dirección física} = \text{Dirección Base} + \text{Desplazamiento} = 219 + 430 = 649$$

b) 1, 10

Comprobar si el desplazamiento es menor que la longitud: $10 < 14$ SÍ

$$\text{Dirección física} = \text{Dirección Base} + \text{Desplazamiento} = 2300 + 10 = 2310$$

c) 3, 400

Comprobar si el desplazamiento es menor que la longitud: $400 < 580$ SÍ

$$\text{Dirección física} = \text{Dirección Base} + \text{Desplazamiento} = 1327 + 400 = 1727$$

d) 4, 112

Comprobar si el desplazamiento es menor que la longitud:

$$112 < 96 \text{ NO}$$

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.

Es más rápido cambiar de un hilo a otro dentro del mismo proceso, que cambiar de un proceso a otro. Este fenómeno se debe a que los hilos comparten datos y espacios de direcciones, mientras que los procesos, al ser independientes, no lo hacen.