

# Relacion-2.pdf



**ferluque**



**Fundamentos del Software**



**1º Doble Grado en Ingeniería Informática y Matemáticas**



**Facultad de Ciencias  
Universidad de Granada**

**NEW**

## **WUOLAH** Print

Lo que faltaba en Wuolah



**Imprimir**



- ☐ Todos los apuntes que necesitas están aquí
- ☐ Al mejor precio del mercado, desde **2 cent.**
- ☐ Recoge los apuntes en tu copistería más cercana o recíbelos en tu casa
- ☒ Todas las anteriores son correctas

## Relación 2

### Ejercicio 1

No se podría reducir ninguno pues todos son necesarios para la correcta ejecución del proceso

### Ejercicio 2

Pues porque podría modificar alguna dirección de memoria perteneciente a algún otro proceso (incluso al SO), lo que podría llevar a errores en sus respectivas ejecuciones. Informa de ello el gestor de memoria del sistema operativo

### Ejercicio 3

Sí, pues a pesar de ser monousuario puede ser multiprogramado, lo que hace necesaria la existencia de este modelo

### Ejercicio 4

Pasaría a finalizado, informando del error producido

### Ejercicio 5

En este caso no tendría sentido, pues en un sistema monoprogramado los procesos simplemente se ejecutan de forma secuencial uno detrás de otro, por ello no necesitan ninguna planificación.

### Ejercicio 6

En ninguna. Siempre que tengamos varios procesos ejecutándose con multiprogramación es necesario el planificador de procesos

### Ejercicio 7

b), c), f), g)

### Ejercicio 8

Pues principalmente el objetivo de proteger direcciones de memoria reservadas al sistema operativo. Si esto no fuera así, cualquier programa del usuario podría acceder a estas direcciones de memoria y provocar fallos en el funcionamiento del sistema.

Básicamente se trata de establecer una jerarquía de poderes.

### Ejercicio 9

a) Sí, pues es necesario que el sistema sepa que el proceso que estaba ejecutándose ha finalizado para que pueda pasar el siguiente a ejecución. En sistemas monoprogramados ocurre lo mismo.

b)

c) No, ya que no tiene permisos para hacerlo pues se debe hacer en modo kernel.

d) Sí, pues si un proceso pasa de ejecutándose a bloqueado, el procesador puede pasar algún otro proceso de la lista de preparados a ejecutarse.

e) En este caso el que pase de bloqueado a ejecutable no provocará un cambio de estado en otros procesos, pues hasta que el proceso que esté en ejecución no finalice, el proceso que ha pasado a ejecutable no tendrá opción de empezar a ejecutarse.

### Ejercicio 10

Pues la principal implicación era la disminución del rendimiento, pues cada byte de datos necesitaba pasar por el procesador, lo que reducía en gran cantidad la eficiencia del mismo.

### Ejercicio 11

Son varias las razones:

- La primera es que esto permite una dependencia entre el shell y el SO a la hora, por ejemplo, de actualizarse. En el caso de que formara parte del SO, tendríamos que modificar también el SO para actualizar el shell.
- Otra razón es que, si formara parte del SO, la ejecución del shell se haría en modo kernell, lo que podría llevar al usuario a acceder a posiciones de memoria reservadas y restringidas.

### Ejercicio 12

a) Crear un proceso.

No, pues solamente se crea pero no pasa a ejecutándose.

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

No como tal, pero al pasar a finalizado, otro proceso pasará a ejecutándose y por tanto se necesitará el contexto del nuevo proceso. Esto en caso de que estuviese ejecutando, si estaba bloqueado no.

c) Suspender o bloquear un proceso.

Igual que b)

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

En este caso sí directamente, necesitamos el contexto del proceso que se

e) Modificar la prioridad de un proceso.

No.

### Ejercicio 13

Sí, al igual que la cola de procesos preparados

### Ejercicio 14

Para que el direccionamiento lógico sea transparente al programador/ensamblador/montador y le sea fácil al hardware del procesador realizar la traducción de la instrucción

### Ejercicio 15

Memoria Real : 16 MB

Dirección lógica: 32 bits

22 bits para el número de página y 10 bits para el desplazamiento

a) Tamaño de la página =  $2^{10}$  palabras de memoria =  $2^{10}$  bytes (suponiendo que 1 palabra = 1 byte)

b) N° marcos de página =  $16 \text{ MB} * \frac{2^{10} \text{ KB}}{1 \text{ MB}} * \frac{1 \text{ página}}{1 \text{ KB}} = 2^{14}$  marcos

c) 14 bits:  $2^{14}$  páginas

d)  $103 \text{ KB} * \frac{1 \text{ página}}{1 \text{ KB}} = 103$  páginas

Por tanto, la tabla de páginas deberá tener 103 entradas para dicho proceso

### Ejercicio 16

a) 999

Dirección virtual: 0011 1110 0111

Número de pág virtual Desplazamiento

Número de marco: 4

01 0011 1110 0111

b) 2121

Dirección virtual: 1000 0100 1001

Número de pág virtual Desplazamiento

Número de marco: 1

00 0100 0100 1001

c) 5400

Dirección virtual: 1 01 01 0001 1000

Número de pág virtual Desplazamiento

Número de marco: 0

00 0001 0001 1000

### Ejercicio 17

quieres trabajar en Wuolah??

TE  
BUSCAMOS

sin ánimo de lucro, chequea esto:



tú puedes ayudarnos a llevar  
**WUOLAH** al siguiente nivel  
(o alguien que conozcas)

### Ejercicio 18

Páginas de 1024 palabras  $\Rightarrow$  10 bits de desplazamiento

$$a) 2453_{10} = 1001\ 1001\ 0101_2$$

**10 01 1001 0101 Desplazamiento**

Nº de página: 10

Desplazamiento: 01 1001 0101

$$b) 9322_{10} = 10\ 0100\ 0110\ 1010_2$$

Desplazamiento: **00 0110 1010**

No puede traducirse pues el desplazamiento no es igual

### Ejercicio 19

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.

**4096 bytes:**

$$A: 5 + 4 + 3 = 12 \text{ páginas}$$

$$B: 4 + 3 + 2 = 9 \text{ páginas}$$

$$C: 5 + 4 + 3 = 12 \text{ páginas}$$

Memoria total =  $(12 + 9 + 12) \cdot 4096 \text{ bytes} = 135168 \text{ bytes} > 131072 \text{ bytes}$ , que es la memoria física del sistema, por lo que se excede. Por tanto esta no es una opción.

**512 bytes:**

$$A: 40 + 28 + 20 = 88 \text{ páginas}$$

$$B: 32 + 17 + 16 = 65 \text{ páginas}$$

$$C: 36 + 26 + 18 = 80 \text{ páginas}$$

Memoria total =  $(88 + 65 + 80) \cdot 512 = 98816 \text{ bytes} < 131072 \text{ bytes}$ , por tanto esta es la única opción posible.

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.

Necesitaríamos 9 bits para indicar el desplazamiento, y 8 bits para indicar el marco de página (pues hay 131072/512 marcos de página), por tanto el formato sería el siguiente:

X XXXX XXXX XXXX XXXX

También se puede añadir un bit de protección, por lo que el formato ocuparía un total de 18 bits.

c) ¿Cuántas Tablas de Páginas habrá en este sistema? ¿Cuántas entradas hay en cada tabla de páginas (filas)?  
Habrá tres tablas de páginas, una por cada proceso, y cada tabla tendrá tantas filas como páginas necesite el proceso.

### Ejercicio 20

### Ejercicio 21

2048 bytes/página

8 MB de memoria física.

**1 proceso: 31566 bytes**

**otro proceso: 18432 bytes**

Si cada página ocupa tiene 2048 posiciones, calculemos el número de páginas que necesitaremos para cada proceso:

- ☐ Todos los apuntes que necesitas están aquí
- ☐ Al mejor precio del mercado, desde **2 cent.**
- ☐ Recoge los apuntes en tu copistería más cercana o recíbelos en tu casa
- ☒ Todas las anteriores son correctas

$$31566 \text{ bytes} * \frac{1 \text{ página}}{2048 \text{ bytes}} = 15,4131 \text{ páginas} \Rightarrow \text{Necesitamos } 16 \text{ páginas} \Rightarrow \text{Se fragmentan } 16 * 2048 - 31566 = 1202 \text{ bytes}$$

$$18432 \text{ bytes} * \frac{1 \text{ página}}{2048 \text{ bytes}} = 9 \text{ páginas} \Rightarrow \text{Necesitamos } 9 \text{ páginas} \Rightarrow \text{No se fragmentan}$$

Luego con esta paginación se fragmentarían 1202 bytes.

También debemos probar si se pueden cargar ambos programas a la vez en memoria, es decir, que la suma de las páginas de los dos procesos no supere la cantidad de memoria real total:

$$(16 + 9) \text{ páginas} * \frac{2048 \text{ bytes}}{1 \text{ página}} * \frac{1 \text{ MB}}{2^{20} \text{ bytes}} = 0,0488 \text{ MB} < 8 \text{ MB}$$

Luego sí que se pueden cargar ambos programas en memoria.

## Ejercicio 22

a)

Desplazamiento:  $430 < 600$  (longitud del segmento 0), luego es correcta

$$\text{Nº segmento: } 0 \Rightarrow \text{Dirección física} = 219 + \text{Desplazamiento} = 219 + 430 = \mathbf{649}$$

b)

Desplazamiento:  $10 < 14$  (longitud del segmento 1), luego es correcta

$$\text{Nº segmento: } 1 \Rightarrow \text{Dirección física} = 2300 + \text{Desplazamiento} = 2300 + 10 = \mathbf{2310}$$

c)

Desplazamiento:  $400 < 580$  (longitud del segmento 3), luego es correcta

$$\text{Nº segmento: } 3 \Rightarrow \text{Dirección física} = 1327 + \text{Desplazamiento} = 1317 + 400 = \mathbf{1717}$$

d)

Desplazamiento:  $112 > 96$  (longitud del segmento 4), luego es **incorrecta**

## Ejercicio 23

a) El producido entre dos hebras del mismo proceso

Este cambio de contexto tardará menos porque las hebras del mismo proceso tienen recursos comunes a todas las hebras, tales como el PCB del proceso y el espacio de direcciones del usuario.

Imprimir

