

Relación de Problemas 2

1

No se podría reducir el espacio de ninguna, ya que mientras este activo es necesario tenerlo en la lista de procesos y su información de contexto para organizar los procesos.

Además, los datos se pueden volver a necesitar y las funciones definidas en el código también.

2

Porque al acceder a una dirección de memoria fuera del área asignada puede cambiar cosas pertenecientes a otro programa o información importante para el SO. El Sistema Operativo se encarga de gestionar la interrupción.

3

Sí, siempre tiene sentido un sistema de 5 estados para los procesos.

4

c) Finalizado. Al no puede ejecutarse correctamente no tiene sentido cambiar a ningún otro estado.

5

No tiene sentido ya que no es necesario gestionar varios procesos al ejecutarse solo uno.

6

Si solo se ejecutara un único proceso no sería necesario el planificador, pero en cuanto hubiera más de 1 sí, por lo que es una situación bastante irreal.

7

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

8

Para acceder a todos los recursos de la máquina y gestionar interrupciones, excepciones o llamadas al sistema es necesario el modo kernel independientemente del número de usuarios.

9

- a) Sí, es necesario para poder borrar ese programa de memoria e indicar que se puede ejecutar uno nuevo, en sistemas monoprogramados también.
- b) No, ya que puede producirse solo un cambio de modo.

- c) El SO es el que determina que debe esperar bloqueado y el que cambia su estado.
- d) Si hay algún proceso preparado este pasará a ejecutarse.
- e) No provoca ningún cambio de estado en otro proceso.

10

El tiempo usado en leer y escribir datos no podía ser usado para ejecutar otros programas.

11

Al no ser parte del SO este puede modificarse independientemente de él.

12

- a) Si tiene más prioridad que el que se está ejecutándose actualmente provocaría un cambio de contexto.
- b) Si ese proceso se estaba ejecutando y hay otro listo implicaría un cambio de contexto.
- c) Si hay otro listo implicaría un cambio de contexto.
- d) Si hay otro proceso ejecutándose provocaría un cambio de contexto.
- e) Si un proceso listo obtiene más prioridad que el que está ejecutándose se daría un cambio de contexto.

13

Tiene sentido en el caso de que los procesos estén esperando que un mismo recurso este libre.

14

Porque la memoria física esta dividida en potencias de dos.

15

- a) El correspondiente al desplazamiento. 2^{10} bytes
- b) $2^{24} \text{ bytes} / 2^{10} \text{ bytes} = 2^{14}$ páginas y marcos de página.
- c) Para representar 2^{14} marcos de página necesitas 14bits.
- e) $103 * 2^{10} \text{ bytes} / 2^{10} \text{ bytes} = 103$ páginas.
Tamaño de la Tabla de Páginas = $103 * 15\text{bits}$.

16

Desplazamiento físico = $n^{\circ}\text{marco} * \text{tamaño página} + \text{desplazamiento}$.

- a) $999/1024 = 0$ resto 999 – $n^{\circ}\text{virtual} = 0$, $DF = 4 * 1024 + 999 = 5095$

b) $2121/1024 = 2$ resto 73 – n.º virtual = 1, DF = $1024 + 73 = 1097$

c) $5400/1024 = 5$ resto 280 – n.º virtual = 0, DF = 280

17

Fragmentación interna ya que si no se rellena la página entera se dejan huecos sin usar en memoria. Haciendo las páginas más pequeñas se soluciona este problema pero la tabla de páginas ocuparía más memoria.

18

a) $2453 / 1024 = 2$ resto 405 . N.º de página virtual = 2. Desplazamiento = 405.

b) $9322 / 1024 = 9$ resto 106 . No sería posible ya que aunque el número de marco fuera el mismo el desplazamiento es diferente.

19

a)

Si utilizamos 4K bytes :

- Proceso A –

$20480 / 4K = 5$ páginas.

$14288 / 4K = 3.58$ (4 páginas).

$10240 / 4K = 2.5$ (3 páginas).

Total : 12 páginas

- Proceso B –

$16384 / 4K = 4$ páginas.

$8200 / 4K = 3$ páginas.

$8192 / 4K = 2$ páginas.

Total : 9 páginas

- Proceso C –

$18432 / 4K = 5$ páginas.

$13288 / 4K = 4$ páginas.

$9216 / 4K = 3$ páginas.

Total : 12 páginas

Total todos los procesos : 33 páginas * 4096 bytes = 135168 bytes.

Si utilizamos 1/2K bytes :

- Proceso A : $20480 / 1/2K + 14288 / 1/2K + 10240 / 1/2K = 88$ páginas

- Proceso B : $16384 / 1/2K + 8200 / 1/2K + 8192 / 1/2K = 65$ páginas.

- Proceso C : $18432 / 1/2K + 13288 / 1/2K + 9216 / 1/2K = 80$ páginas.

Total todos los procesos : $233 \text{ páginas} * 512 \text{ bytes} = 119296 \text{ bytes}$.

Solución : La opción más apropiada sería la de 512 bytes ya que ocupa menos espacio.

b) 1bit de protección + 9 bits de desplazamiento ($512 \text{ bytes} = 2^9$) + 8 bits de direcciones de marco (memoria total/tamaño de marco = n.º de marcos, $131072 \text{ B} / 512 \text{ B} = 256$, $256 = 2^8$) = 18 bits.

c) Una tabla por proceso (3 tablas) y una entrada por página (A – 88 entradas, B – 65 entradas y C – 80 entradas)

20

Una tabla de páginas.

21

$31.566 / 2048 = 15.41$ páginas. $0.58 \text{ páginas sin usar} * 2048 \text{ bytes} = 1202 \text{ bytes de fragmentación}$.

$18432 / 2048 = 9$ páginas. No provoca fragmentación interna.

22

a) $219 + 430 = 649$

b) $2300 + 10 = 2310$

c) $1327 + 400 = 1727$

d) 112 es mayor que 96 por lo que no se puede traducir.

23

Las hebras de un mismo proceso comparten memoria, al cambiar de contexto no será necesario consultar la ubicación de otro proceso, por lo que se producirá más rápido.