

Prueba de evaluación a distancia 2 (PED2)

Sistemas Operativos



Asignatura:

Sistemas Operativos

Curso 2011-2012

Fecha de entrega: 16/01/2012

Alumno:

Diego Burgaleta Jiménez

D.N.I.: 78.746.258-P

Email: diego.burgaleta@gmail.com

Teléfono: +34 649 033 066

Centro Asociado de Pamplona

Ejercicio 1:

Explique razonadamente si las siguientes afirmaciones son verdaderas o falsas:

I) (1 p) La sobrepaginación aumenta el porcentaje de uso del procesador.

II) (1 p) Se denomina buffering de páginas a la estrategia consistente en cargar un cierto número de páginas de un proceso antes de iniciar o continuar su ejecución.

Respuestas:

- I) Falso. En un entorno con sobrepaginación se estarán provocando fallos de página constantemente, por lo que el sistema necesitará una gran cantidad de tiempo para el intercambio de páginas y se ralentizará. Como consecuencia el porcentaje de uso del procesador disminuirá.
- II) Falso. A la estrategia de carga de páginas antes de iniciar o continuar la ejecución de un proceso se le llama paginación por adelantado o prepaging. El término buffering de páginas se refiere a la estrategia de mantener un marco de página en la lista de marcos libres en lugar de vaciarla, con el fin de que en caso de se vuelva a necesitar esa página se ahorre tiempo al no tener que volver a leerla de la memoria secundaria.

Ejercicio 2:

(2 p) Un sistema con memoria virtual mediante demanda de páginas utiliza el algoritmo LRU para la sustitución de páginas. Un proceso genera la siguiente secuencia de referencias a páginas de memoria:

1 3 2 4 1 5 7 4 3 2 8 9 4 5 4 9 1 8 3 2

a) Determinar cuántos fallos de página se producen cuando se dispone de 4 o 5 marcos de página para este proceso.

b) Explicar razonadamente si mejoraría la tasa de fallos de página si se aumentase el número de marcos de página a N , siendo $N > 5$.

Respuesta:

a)

- Intercambio con 4 marcos de página:

1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2		
				4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2	Pila
			2	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	
		3	3	3	2	4	1	5	7	4	3	2	8	9	9	5	4	9	1	8	
	1	1	1	1	3	2	4	1	5	7	4	3	2	8	8	8	5	4	9	1	
F	F	F	F	A	F	F	A	F	F	F	F	F	F	A	A	F	F	F	F		

	1	1	1	1	1	1	1	1	3	3	3	3	4	4	4	4	4	3	3		Marco i
		3	3	3	3	5	5	5	5	2	2	2	2	5	5	5	5	8	8	8	Marco j
			2	2	2	2	7	7	7	7	8	8	8	8	8	8	1	1	1	1	Marco k
				4	4	4	4	4	4	4	4	9	9	9	9	9	9	9	9	2	Marco l

Se ha representado el estado de la pila en cada referencia de páginas y los valores que contienen los marcos i, j, k y l en cada momento. De un total de 20 referencias, con 4 marcos de página disponibles se producen **16 fallos de página**.

- Intercambio con 5 marcos de página:

1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2	
						5	7	4	3	2	8	9	4	5	4	9	1	8	3	2
				4	1	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3
			2	2	4	4	1	5	7	4	3	2	8	9	9	5	4	9	1	8
		3	3	3	2	2	4	1	5	7	4	3	2	8	8	8	5	4	9	1
	1	1	1	1	3	3	2	2	1	5	7	4	3	2	2	2	8	5	4	9
F	F	F	F	A	F	F	A	F	F	F	F	A	F	A	A	F	A	F	F	

	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	Marco i
		3	3	3	3	3	7	7	7	7	7	9	9	9	9	9	9	9	9	Marco j
			2	2	2	2	2	2	3	3	3	3	3	5	5	5	5	3	3	Marco k
				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	Marco l
					5	5	5	5	5	5	8	8	8	8	8	8	8	8	8	Marco m

En esta ocasión, de un total de 20 referencias, con 5 marcos de página disponibles se producen **14 fallos de página**.

b) El objetivo de aumentar el número de marcos de página disponibles es que al tener mayor número de páginas referenciadas, el número de fallos disminuya, lo que cumpliría el principio de que a mayores recursos mejor comportamiento. Sin embargo hay casos en los que este principio de que a mayores recursos mejor comportamiento no se cumple, con algunos algoritmos se sufre lo que se llama la anomalía de Belady, en los que a mayores recursos a veces aumenta el número de fallos.

Sin embargo el algoritmo LRU, como los que se basan en pila, no sufre de la anomalía de Belady y podemos concluir que **para $N > 5$ la tasa de fallos se mantendría igual o mejoraría**.

Ejercicio 3:

(2 p) Explique razonadamente las funciones que realizan las capas de software de E/S del núcleo de un sistema operativo.

Respuesta:

El software de E/S del núcleo de un sistema operativo se puede organizar en tres capas:

- Subsistema de E/S:

Se encarga de efectuar las tareas de E/S comunes a todos los dispositivos e independientemente de los mismos:

- Asignación y liberación de dispositivos dedicados, que son aquellos que sólo pueden ser utilizados simultáneamente por un solo proceso. El subsistema de E/S se encarga de aceptar o rechazar las llamadas de un proceso a uno de estos dispositivos.
- Bloqueo de procesos que solicitan una operación de E/S, en caso de que un proceso haya hecho una llamada a un dispositivo de E/S que no está disponible.
- Planificación de la E/S, para distribuir equitativamente el uso de los dispositivos de E/S entre los procesos que los requieren.
- Invocación del driver de dispositivo apropiado, el subsistema de E/S referencia a la posición de la memoria donde se encuentra el driver apropiado a cada dispositivo.
- Almacenamiento temporal de datos de E/S o buffering, para mejorar la eficiencia en la transferencia de datos entre los procesos y los dispositivos de E/S.
- Proporcionar un tamaño de bloque uniforme a los niveles superiores de software para que puedan trabajar con direcciones de memoria lógicas independientemente de la memoria física utilizada por los dispositivos de E/S.
- Gestión de errores y de las acciones a realizar en la llamada a un dispositivo de E/S.

- Drivers de dispositivos de E/S:

Es el código específico que necesita el sistema operativo para controlar un determinado dispositivo de E/S y es independiente para cada uno. Determina las funciones que se pueden realizar sobre el dispositivo e interactúa con el subsistema de E/S al que le facilita una interfaz con las acciones que puede realizar el dispositivo.

El driver se encarga de:

- Comprobar que los parámetros facilitados por el subsistema de E/S son válidos.
- Traducir esos parámetros a los específicos del dispositivo.
- Controlar si el dispositivo está disponible.
- Generar y cargar en el controlador del dispositivo las órdenes adecuadas a la petición solicitada.
- Esperar a la ejecución de las órdenes y bloquearse si fuera necesario hasta que se produzca una interrupción.
- Comprobación de si se han producido errores.
- Examinar las peticiones pendientes y atenderlas.

- Manejador de interrupciones :

Se encarga de generar una interrupción para desbloquear el driver de un dispositivo de E/S cuando este dispositivo ha terminado una operación. También puede encargarse de transferir datos desde un registro del controlador del dispositivo a un buffer del núcleo y viceversa según sea una operación de lectura o de escritura en el dispositivo de E/S.

Ejercicio 4:

En un computador con una capacidad de memoria principal de 64 kibipalabras se utiliza gestión de memoria mediante segmentación. La tabla de segmentos (todos los datos numéricos están en decimal) es la siguiente:

<i>Nº de segmento</i>	<i>Base</i>	<i>Longitud</i>
<i>0</i>	<i>0</i>	<i>7230</i>
<i>1</i>	<i>16384</i>	<i>8191</i>
<i>2</i>	<i>32768</i>	<i>1024</i>
<i>3</i>	<i>8192</i>	<i>356</i>
<i>4</i>	<i>24576</i>	<i>4200</i>

Se pide:

- a) (1 p) Supuesto que una dirección lógica tiene el mismo tamaño en bits que una dirección física y que consta de los campos [nº de segmento, desplazamiento], determinar el tamaño en bits de cada uno de estos campos.
- b) (1 p) Determinar a qué direcciones físicas expresadas en decimal corresponden las siguientes direcciones lógicas expresadas en hexadecimal: i) 11AE₁₆, ii) 6190₁₆,

Respuestas:

- a) Como no se ha especificado nada, suponemos que la unidad direccionable es la palabra y que la tabla de longitudes de los segmentos están en palabras.

La capacidad de la memoria principal es de 64 kibipalabras, por lo que tenemos:

$64 * 1024 = 2^6 * 2^{10} = 2^{16}$ palabras, lo que significa que tenemos direcciones físicas de 16 bits.

El número de segmentos $N_s = 5$, por lo que para representar 5 segmentos aplicamos la desigualdad:

$$\min_s \{N_s \leq 2^s\}$$

Se obtiene que $5 \leq 2^3$ por lo que para representar el **segmento** necesitamos un mínimo de **3 bits**.

El tamaño del campo de desplazamiento se calcula utilizando la siguiente desigualdad:

$$\min_d \{S_s \leq 2^d\}$$

S_s es la mayor longitud de la tabla de segmentos, en nuestro caso para el segmento 1, su longitud es 8191

Resolviendo la desigualdad $8191 \leq 8192 \Rightarrow 8191 \leq 2^{13}$ y esto significa que para representar el **desplazamiento** necesitamos **13 bits**.

En este caso se cumple que la dirección lógica tiene el mismo número de bits que la dirección física (16 bits), divididos en 3 bits para el segmento y 13 bits para el desplazamiento.

$$\begin{aligned} \text{b) i) } 11AE_{16} &= 1 \cdot 16^3 + 1 \cdot 16^2 + A \cdot 16^1 + E \cdot 16^0 \\ &= 1 \cdot 4096 + 1 \cdot 256 + 10 \cdot 16 + 14 \\ &= 4096 + 256 + 160 + 14 \\ &= \mathbf{4526_{10}} \end{aligned}$$

$$\begin{aligned} \text{ii) } 6190_{16} &= 6 \cdot 16^3 + 1 \cdot 16^2 + 9 \cdot 16^1 + 0 \cdot 16^0 \\ &= 6 \cdot 4096 + 1 \cdot 256 + 9 \cdot 16 + 0 \\ &= 24576 + 256 + 144 + 0 \\ &= \mathbf{24976_{10}} \end{aligned}$$

Ejercicio 5:

La política de gestión de memoria de un cierto sistema es del tipo demanda de página. El tamaño de una página es de 1 KiB, el tamaño máximo de la memoria virtual es de 4 MiB y el tamaño de la memoria física es de 1 MiB. Se pide:

a) (1 p) Determinar el tamaño de cada uno de los campos de una dirección virtual y de una dirección física.

b) (1 p) Determinar la capacidad mínima que debe tener la tabla de páginas del proceso de mayor tamaño que se puede ejecutar en el sistema. ¿Qué tanto por ciento de la memoria principal ocuparía dicha tabla?

Respuesta:

- a) Igual que en el ejercicio anterior, suponemos que las palabras son de 1 B (8 bits) y que la unidad de direccionamiento es la palabra.

En el enunciado se nos da la capacidad de la memoria principal (C_{MP}) que es 1 MiB. Para la dirección física resolvemos la desigualdad:

$$\min_n \{C_{MP} \leq 2^n\}$$

De aquí obtenemos que $1\text{MiB} = 1.048.576 \text{ bytes} \leq 2^{20}$ por lo que la **direcciones físicas son de 20 bits.**

Para el cálculo del desplazamiento, tenemos que tener en cuenta el tamaño de la página S_p que en nuestro caso es 1 KiB. Para el cálculo de bits necesarios resolvemos la desigualdad:

$$\min_d \{S_p \leq 2^d\}$$

Y de aquí obtenemos que $1024 \leq 2^{10}$, por lo que el **campo de desplazamiento** es de **10 bits**.

Para el cálculo del tamaño del campo de los marcos de página se restan estos dos valores, por lo que el **campo de marco de página** es $f = n - d = 10$ bits.

Para la memoria virtual, sabemos que tiene un tamaño C_A de 4MiB y para calcular el tamaño en bits m de una dirección virtual tenemos que resolver la desigualdad:

$$\min_m \{C_A \leq 2^m\}$$

Y despejando $4 \cdot 2^{10} = 2^2 \cdot 2^{20} \leq 2^{22}$ por lo que una **dirección virtual** se representa mediante **22 bits**.

El **campo de desplazamiento** tiene que ser igual que en el caso anterior, es decir, **10 bits**, por lo que el **campo del número de página** es de **12 bits**.

- b) Como mínimo la tabla de páginas usadas en paginación por demanda contienen un bit de presente/ausente más que en las de paginación simple, por lo que según el enunciado el tamaño E de una entrada es de $10+1 = 11$ bits.

Para el cálculo del número máximo de páginas de un proceso que se puede cargar en este sistema N_P se calcula:

$$N_P = \text{ceil} \left(\frac{C_A}{S_p} \right) = \text{ceil} \left(\frac{4 \text{ MiB}}{1 \text{ KiB}} \right) = \text{ceil} \left(\frac{2^{22}}{2^{10}} \right) = 2^{10} \text{ páginas}$$

Por lo tanto la tabla de páginas debe tener 2^{10} entradas.

El tamaño de la tabla de páginas C_{TP} del proceso se calcula:

$$C_{TP} = E \text{ (bits/entrada)} * N_P \text{ (entradas)} = 11 \cdot 2^{10} = 11264 \text{ bits} = 1408 \text{ B}$$

Y el porcentaje de la memoria principal que ocupa esta tabla es:

$$P = \frac{C_{TP}}{C_{MP}} * 100 = \frac{1408}{2^{20}} * 100 = 0,13428 \%$$