

SISTEMAS OPERATIVOS

2º Curso – Dobles grados

Preguntas del Tema 3

Esbozo de las soluciones:

Tema 3:

1. Suponga un espacio virtual de direcciones de 16 bits (3 bits para la tabla de páginas de primer nivel, 3 bits para las tablas de páginas de segundo nivel y 10 bits para el desplazamiento). El tamaño de página es 1Kbyte y cada palabra de memoria ocupa 1Byte. El mapa de memoria de un proceso es el siguiente: 3578 Bytes de texto (código que comienza en la dirección virtual 0), 4096 Bytes de datos (a continuación del código) y 5500 Bytes de pila (cuya base está en la última dirección virtual). La Figura muestra las tres regiones en el espacio virtual. Construya las tablas de páginas necesarias para describir este espacio de memoria suponiendo paginación a dos niveles, y suponiendo que la primera página está cargada en el marco 10 y el resto de forma consecutiva. Las tablas de páginas están cargadas a partir del marco 0. A continuación, realice la traducción de las siguientes direcciones virtuales: 5000, 14200 y 64000 y ponga todos los cálculos utilizados.

| |
|-----------------------------|
| Texto(Código) 3578 Bytes |
| Datos 4096 Bytes |
| <i>heap</i> |
| Pila 5500 Bytes |

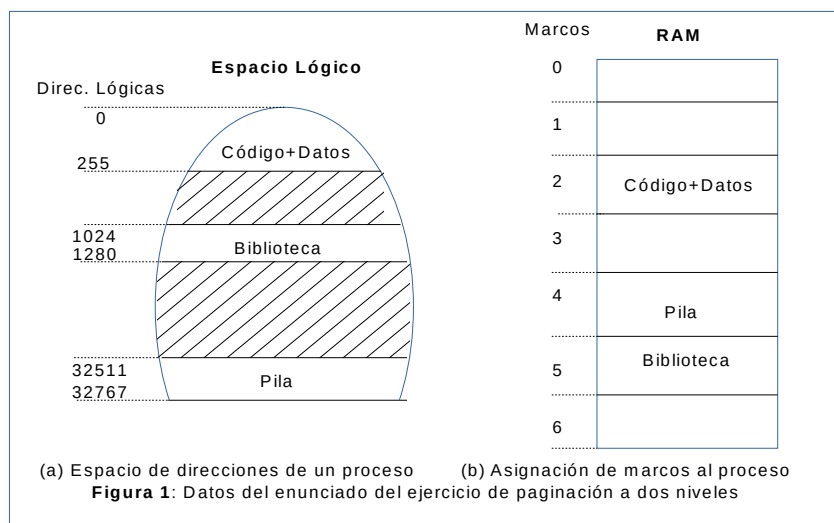
- Tablas 1N y 2N: $2^3 = 8$ PTE's. Tamaño página = 2^{10}
 - Cada PTE's de 1N direcciona: 8 entradas TP2N x 1024 bytes (2^{10}).
 - Cada PTE's de 2N direcciona: 1024 bytes.
 - Código y datos: 4 páginas cada una; Pila: 6 páginas:
 - Código y datos: usan la PTE 0 de 1N y una TP2N completa.
 - Pila: PTE 7 de 1N, y PTE's 2-7 de 2N.
 - Traducción de la DL 5000: esta dirección es alcanzable usando la PTE 0 de 1N y la PTE 4 2N y tiene un offset de 904 $\rightarrow DF = N \times 1024 + 904$.
 - Traducción de la DL 14200: no es válida (PTE 1 de 1N)
 - Traducción de 64000: se calcula de forma similar a la de 5000
2. A partir del espacio de direcciones virtuales descrito en el ejercicio anterior:
 - a) Representar las estructuras de datos (`mm_struct` y `vm_area`) que utilizaría Linux para describir dicho espacio, rellenando aquellos campos de la estructura que sean posibles y suponiendo que el archivo ejecutable de dicho proceso reside en un archivo denominado `./mi_programa`.
 - La `mm_struct` de teoría.
 - 3 `vm-areas`, una por cada región. Los datos son dirección lógica inicio, dirección lógica fin, protección, almacén de respaldo.
 - b) Qué modificaciones habría que realizar en las estructuras de datos antes citadas, si se realiza un proyección de un archivo de datos de 3500 bytes en la dirección virtual 16384.
Se crea una nueva región (`vm-area`) con inicio en la dirección lógica 16283 y fin de $16384 + 3500$

3. (a) Indicar los pasos que sigue un sistema operativo general en el tratamiento de una excepción de falta de página. (b) En el caso de Linux, ¿qué papel juegan las *vm-areas* en el tratamiento de la excepción del falta de página?

(a) aplicación de teoría

(b) teoría (ver manejador falta página en Linux y mecanismo COW).

4. Suponga un espacio virtual de direcciones de 16 bits en un sistema paginado a doble nivel que direcciona byte (4 bits para la tabla de páginas de primer nivel, 4 bits para la tabla de página de segundo nivel y 8 bits para el desplazamiento u offset). El mapa de memoria sobre el espacio lógico de un proceso y la asignación de memoria se muestra en la Figura 1. Las direcciones de (a) marcan el inicio y fin de la región, en (b) se muestra en qué marcos están cargadas cada una de las páginas del proceso. Construya las tablas de páginas y sus contenidos más relevantes (marco, bit de presencia/validez) para describir el espacio de la Figura 1(a). A continuación, realice la traducción de las siguientes direcciones virtuales: 2000 y 32600 (indicando todos los cálculos utilizados).



Similiar al Ejercicio 1

5. (a) Dibujar/describir las estructuras de datos que utiliza el kernel de Linux para representar el espacio de usuario del proceso descrito en el ejercicio anterior e indicar los contenidos más relevantes.

Tres vm-areas con los datos del Ejercicio.

(b) Supongamos que el proceso realiza una invocación de `malloc()` para reservar 500B de memoria adicionales para datos dinámicos. Como cambiaría la descripción realizada en el Apartado (a).

Se expande la vm-datos, es decir, la dirección de fin ahora es 255+500.

6. En un sistema Linux, que acciones toma el kernel cuando el proceso actual (resultado de la ejecución del programa que reside en el archivo ejecutable *mi_programa*) genera una falta de página en la región de código.

Aplicar la lógica del manejador de falta de pagina para paginación por demanda.

7. Suponed un sistema con una memoria principal de 160 Bytes y que utiliza **paginación a dos niveles**. Las direcciones son de **8 bits** con la siguiente estructura: **2 bits en la tabla de páginas de primer nivel, 2 bits en la tabla de páginas de segundo nivel y 4 bits para el desplazamiento**. El espacio de direccionamiento virtual de un proceso tiene la estructura del dibujo. Representad gráficamente las tablas de páginas de primer y segundo nivel y sus contenidos, suponiendo que que todas las páginas válidas de este proceso están cargadas en memoria principal en el orden que queráis (todos los marcos de página de memoria principal están libres). Dada esa asignación **traduce la dirección virtual 47 y la dirección virtual 250**.

Similar al ejercicio 1.

| | |
|-----------------|------|
| Texto | 0 |
| Datos | 16B |
| sin usar | 48B |
| | 224B |
| Pila | 256B |