

SISTEMAS OPERATIVOS

2º Curso – Dobles grados

Preguntas del Tema 3

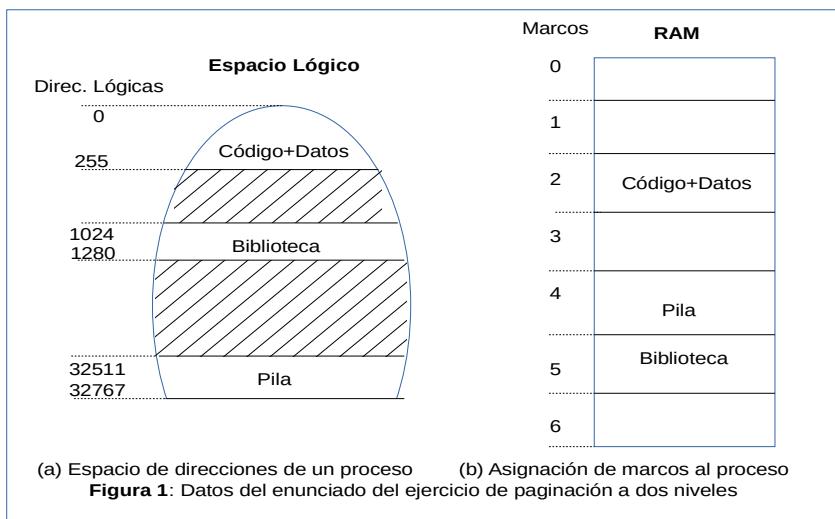
Esbozo de las soluciones:

Tema 3:

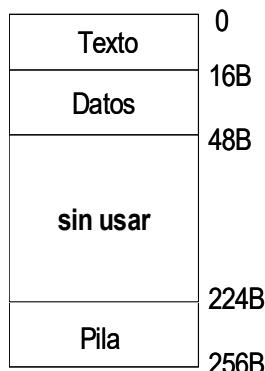
1. Suponga un espacio virtual de direcciones de 16 bits (3 bits para la tabla de páginas de primer nivel, 3 bits para las tablas de páginas de segundo nivel y 10 bits para el desplazamiento). El tamaño de página es 1Kbyte y cada palabra de memoria ocupa 1Byte. El mapa de memoria de un proceso es el siguiente: 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 esta 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

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 las estructura que sean posibles y suponiendo que el archivo ejecutable de dicho proceso reside en un archivo denominado “./mi_programa”.
 - 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.
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?
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).



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.
 (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).
6. En un sistema Linux, qué 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.
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 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**.



Ejercicios tema 3

1.- 3 bits para TP de 1-N $\Rightarrow 2^3 = 8$ PTE de 1-N

3 bits para TP de 2-N $\Rightarrow 2^3 = 8$ PTE de 2-N

10 bits para desplazamiento $\Rightarrow 2^{10} = 1K$ direcciones con 1Byte/direcc.

Cada PTE de 1-N direcciará $8 \cdot 1K = 8K$ bytes

Texto: $35E8 < 8K \Rightarrow$ PTE 0 de 1-N $\Rightarrow 35E8 / 1024 : \text{Cociente} = 3, \text{Resto} = 506$

Terminará en la PTE 3 de 2-N

Datos: Comenzará en la página siguiente al texto, PTE 0 de 1-N, 4 de 2-N

$4096 < 8K \Rightarrow 4096 / 1024 = 4$. Ocupará 4 páginas, luego de las 4 a 7

Pila: $5300 / 1024$: Cociente = 5 Resto = 380 \Rightarrow Ocupará las últimas 6

páginas del espacio código direccinado (PTE 7 de 1-N, 7, 8, 9 de 2-N)

Marcos 0

	Bit Validez	Marcos
0	1	1
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
7	1	2

Marcos 1

	Bit Validez	Marcos
0	1	10
1		11
2	1	12
3		13
4		14
5		15
6	1	16
7	1	17

Marcos 2

	Bit Validez	Marcos
0	0	
1	0	
2	1	23
3	1	22
4	1	21
5	1	20
6	1	19
7	1	18

Traducciones:

• 5000: $5000 < 8K \Rightarrow$ PTE 0 de 1-N. Bit Validez = 1, luego continuamos

$5000 / 1024$: Cociente = 4: PTE de 2-N, Resto = 904: Desplazamiento

En la TP del marco 1, el PTE 4 tiene el bit = 1, luego direcciónamiento correcto. Marco = 14

Direcc. Física = $14 \cdot 1K + 904 = 15240$

• 14200: $14200 / 8K$: Cociente = 1: PTE de 1-N. Su bit de validez = 0, luego direcciónamiento inválido

• 64000: $64000 / 8K$: Cociente = 8: PTE de 1-N, Resto = 6656

$6656 / 1024$: Cociente = 6 Resto = 512 \Rightarrow Bit Validez = 1 \Rightarrow Correcto
Marco = 19 \Rightarrow Direcc. Física = $19 \cdot 1K + 512 = 19968$

2 - a) Por un lado, el mm_struct guardaría:

- Ptr. a lista de vm-áreas
- Número de vm-áreas (3)
- Número de páginas mapeadas (64)
- Dirección de inicio y final del código [0-35EEI]
- " " " " " " datos [4096-8192]
- Dirección de la base de la pila (64K-1)

Las vm-áreas que guardaría son:

• Texto/código:

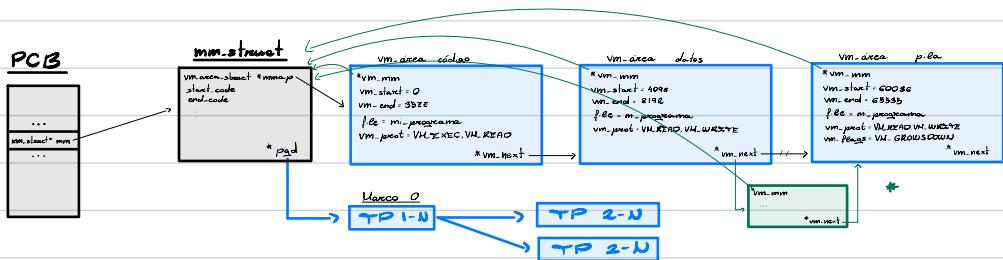
- Dirección inicio y final [0-35EEI]
- Indicadores: VM_EXEC, VM_READ
- Información de enlace
- Ptr. a operaciones
- Ptr. a "/miprograma" y desp. hasta zona código

• Datos:

- Dirección inicio y final [4096-8192]
- Indicadores: VM_READ, VM_WRITE
- Información de enlace
- Ptr. a operaciones
- Ptr. a "/miprograma" y desp. hasta zona de datos

• Pila:

- Dirección inicio y final [64K-1, 64K-1-5500=60035I]
- Indicadores: VM_READ, VM_WRITE, VM_GROWSDOWN
- Información de enlace
- Ptr. a operaciones
- Ptr. a "/miprograma"



b) Comprobando su dirección de inicio y tamaño, vemos que no se solapa con ninguna vm-área ya presente, luego podemos llevar acabo la proyección, se creará un vm-área para representarla:



3- a) Rutina de tratamiento falta de página:

- 1- Comprueba si la referencia es válida si no lo es, aborta. *
- 2- Busca un marco libre en memoria principal. **
- 3- Carga la página en el marco encontrado. **
- 4- Activa su bit de validez.
- 5- Retorna la instrucción causante de la falta.

b) Sirven para ver si una dirección es válida, mirando el start y end de las vm-áreas para comprobar en cuál está.

- *. Hija si es válida dependiendo del SO. En Linux, con vm-áreas
- ** Si no hay marcos libres, el SO utilizará algún algoritmo de reemplazo para liberar alguno. En Linux, LRU
- ** Actualiza TP y, en Linux, vm-áreas

6.- Ante la excepción de falta de página, el Kernel responderá con la rutina correspondiente a ésta, que se encargará de:

- 1.- Comprobar si la referencia es válida consultando los campos start y end de las mmáreas para ver en cuál está. Si no lo es, aborta.
- 2.- Buscar un marco libre o, si no quedan, liberaré alguno según el algoritmo LRU.
- 3.- Carga la página al marco y actualiza los TP y las mmáreas, en este caso, la de código.
- 4.- Retorna la instrucción causante de la falta.

Muy 2 errores en el enunciado: Biblioteca termina 1289 y Pila empieza 32312

4.- 4 bits para TP de 1-N $\Rightarrow 2^4 = 16$ PTE de 1-N

4 bits para TP de 2-N $\Rightarrow 2^4 = 16$ PTE de 2-N

8 bits para desplazamiento $\Rightarrow 2^8 = 256$ direcciones/PTE de 2-N

Cada PTE de 1-N direccionalá $2^4 \cdot 2^8 = 2^{12} = 4K$ direcciones

Supondré que los TP se guardan en los primeros marcos libres

Código + Datos: comienza en 0 (PTE 0 de 1-N, 0 de 2-N) y termina en la 255, osea que ocuparé la primera página

Biblioteca: comienza $1024 < 4K \Rightarrow$ PTE 0 de 1-N $\Rightarrow 1024 / 256 = 4$ (Resto = 0)

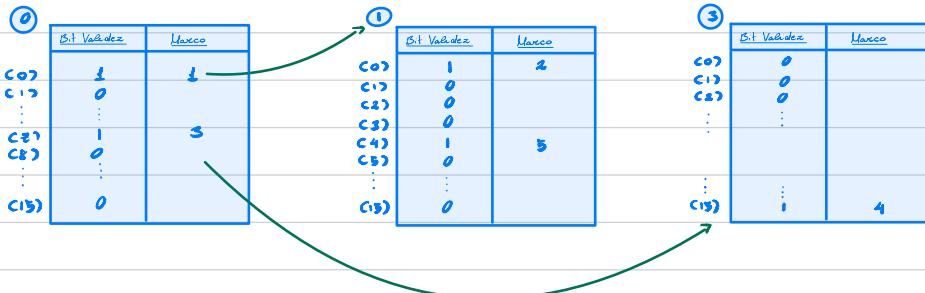
termina $1289 / 256$: Cociente = 4 = PTE de 2-N, Resto = 255

La biblioteca ocupa la página entera

Pila: comienza 32312 $\Rightarrow 32312 / 4K$: Cociente = 8 = PTE de 1-N, Resto = 3840

$3840 / 256 = 15$ (Resto = 0)

Ocuparé $32362 - 32312 + 1 = 256$ direc., la página entera



Traducciones:

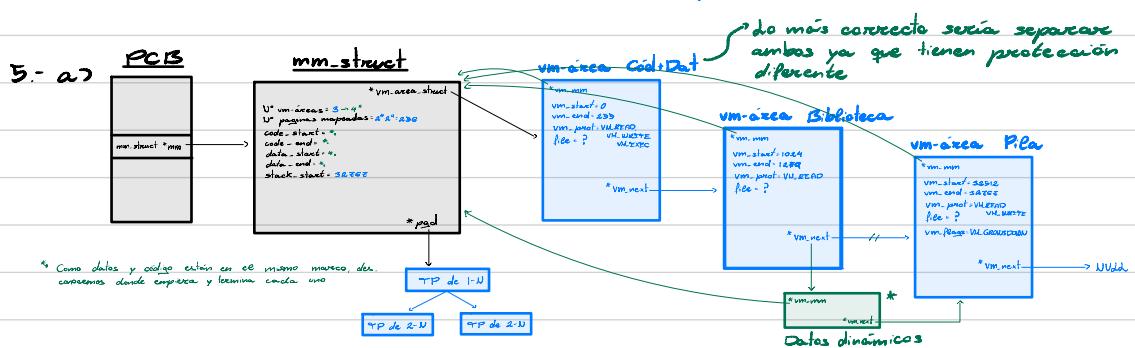
$2000 \rightarrow 2000 < 4K \Rightarrow$ PTE 0 de 1-N $\Rightarrow 2000 / 256$: Cociente = E = PTE de 2-N. Sin embargo, este PTE tiene el bit de validez desactivado, luego directamente inválida.

$32600 \rightarrow 32600 / 14K$: Cociente = E = PTE de 1-N. Resto = 3928

$3928/236$: Cociente = 16 = PYE de 2-N, Resto = 88 = Desplazamiento

Bit val: idea = 1 \Rightarrow Correcto \Rightarrow Marco = 4

$$\text{Dirac. Física} = 256 \cdot 4 + \text{Desp.} = 1112 \text{ ym}$$



b) Suponiendo que malloc aloja los 500B entre la biblioteca y la pila en el espacio lógico, tendríamos que añadir una nueva VM-área entre las de estas.



2 bits TP 1-N \Rightarrow 4 PTE de 1-N

2 bits TP 2-N \Rightarrow 4 PTE de 2-N

4 bits desplazamiento $\Rightarrow 2^4 = 16$ direcciones / PTE de 2-N

Cada PTE de 1-N direccionaría $4 \cdot 2^4 = 64$ direcciones

Texto: comienza en 0 y ocupa 16 direcciones, luego la primera página entera.

Datos: comienza en 16 \Rightarrow PTE 0 de 1-N, 1 de 2-N, y ocupa 48-16=32 direcciones, luego ocupar 2 páginas (1,2)

Pila: comienza en 224 $\Rightarrow 224/64$: Cociente = 3 = PTE de 1-N Resto = 32
32/16 = 2 = PTE de 2-N (Resto = 0 = Desp.)

y ocupa 256-224=32 direcciones \Rightarrow 2 páginas (2,3)

Suponiendo que la distribución de marcos es

0	1	2	3	4	5	6	7
1P _{1,0}	1P _{2,0}	1P _{2,3}	Texto	Datos	1D _{2,0}	0D _{2,0}	0D _{2,1}

Los QP serán:

①

B1 Valores	Marcos
0	1
1	0
2	0
3	1

②

B1 Valores	Marcos
0	1
1	1
2	1
3	0

③

B1 Valores	Marcos
0	0
1	0
2	1
3	1

Traducciones

· 48 \rightarrow PTE 0 de 1-N, 48/16: Cociente = 2 = PTE de 2-N, Resto = 16 = Desp.

Bit Val = 1 \Rightarrow Marco = 5 \Rightarrow Direc. Física = $16 \cdot 5 + 16 = 96$,

· 250 \rightarrow 250/64: Cociente = 3, Resto = 58 \Rightarrow 58/16: Cociente = 3, Resto = 10

Bit Val = 1 \Rightarrow Marco = 6 \Rightarrow Direc. Física = $16 \cdot 6 + 10 = 106$,