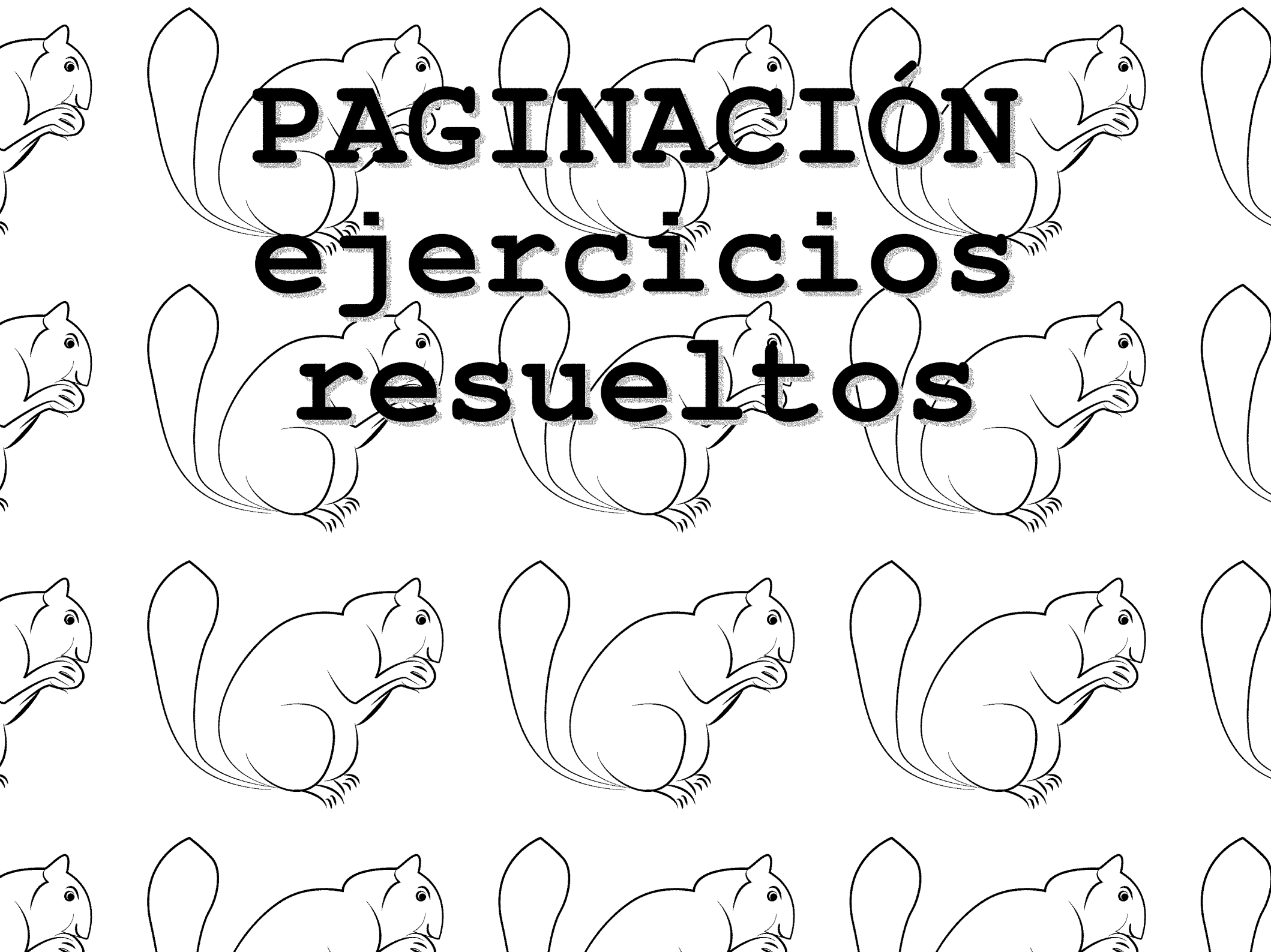


PAGINACIÓN

ejercicios

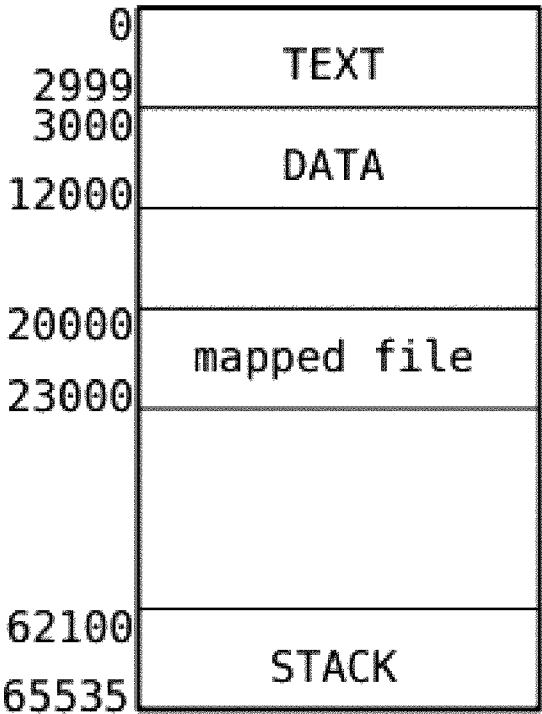
resueltos



Ejercicio de examen



Suponga un sistema que utiliza paginación a dos niveles. Las **direcciones** son de **16 bits** con la siguiente estructura: **3 bits en la tabla de páginas de primer nivel, 3 bits en la tabla de páginas de segundo nivel y 10 bits para el desplazamiento**). El espacio de direccionamiento virtual de un proceso tiene la estructura del dibujo, en la cual se especifican las direcciones virtuales de comienzo y final de cada área válida. Represente gráficamente las tablas de páginas y sus contenidos, suponiendo que cada entrada de la tabla de páginas ocupa: 2 bytes para la dirección base de marco y 1 byte para la codificación del resto de información (V/P, Mod., Prot.); y que todas las páginas están cargadas en memoria principal consecutivamente a partir del marco 10 de RAM. La memoria principal tiene un tamaño de 64KB. Dado este mapa de memoria traduce las siguientes direcciones virtuales: 7428, 19425.



Ej. Dirección lógica:
 [0 1 0 1 1 1 0 0 1 0 0 0 0 1 1 1]
 1^{er} lvl 2^{do}lvl offset(desplazamiento)

Lo primero que haremos es averiguar cuanto ocupa cada página, dado que conocemos que se usan 10bits para el desplazamiento dentro de cada página sabemos que tienen un tamaño de 2^10=1KB

Tam_pag = 1024B = 1KB

Tam_pag = 1024B = 1KB

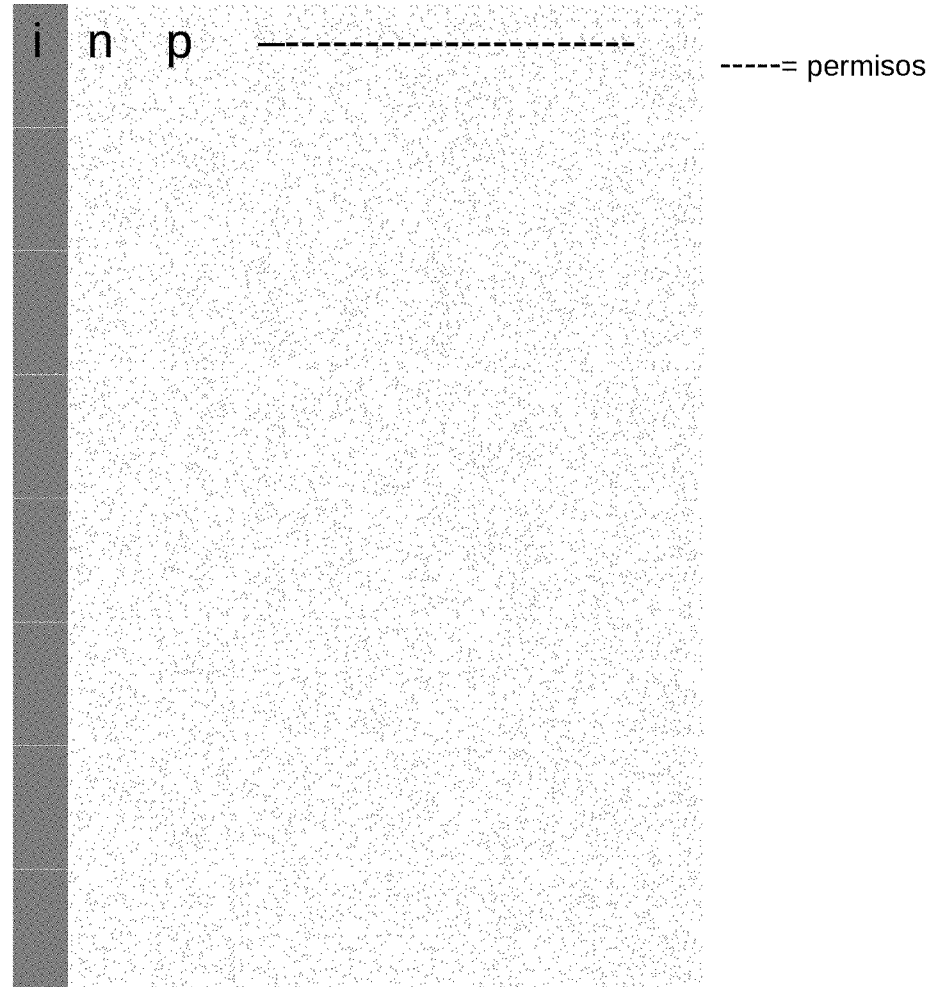
Tam_tabla_1^{er} = 8

Tam_tabla_2^{do} = 8

Igual con el tamaño de cada tabla de páginas, como ambas se direccionan con 3 bits, tendrán $2^3 = 8$ entradas. Cada entrada será del tamaño de una pagina.

Y quedarían tal que así:

l= indice; *n=num marco/tabla; p = bit presencia



* si es la tabla de primer nivel será el num de la tabla de segundo nivel. Si es la de segundo nivel, será el num de marco en MP

Tam_pag = 1024B = 1KB
Tam_tabla_1^{er} = 8
Tam_tabla_2^{do} = 8

Ahora calculamos cuanto ocupa cada sección del programa.

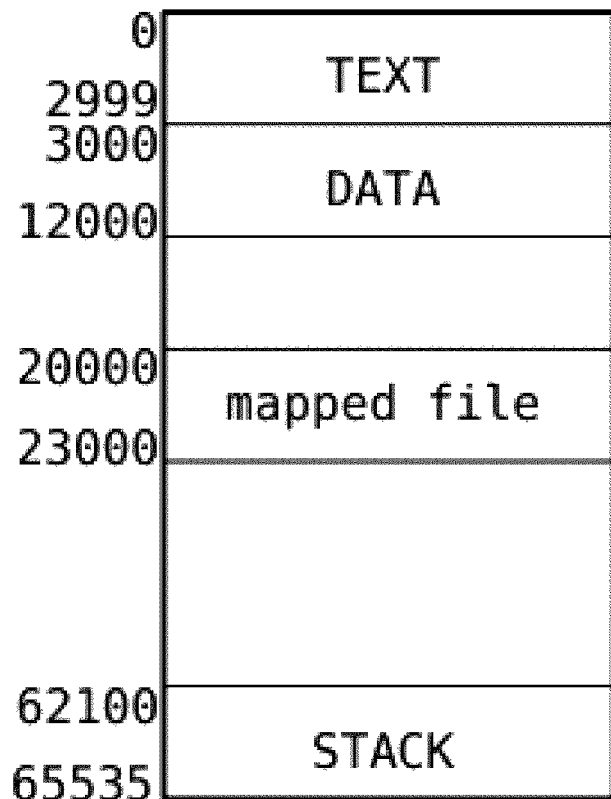
TEXT: 2999-0 = 2999; ocupa 2999B y cada página puede albergar 1024B, por tanto text, ocupará $2999/1024 = 3$ páginas.*

DATA: 12000-3000 = 9000; $9000/1024 = 9$ páginas.

mapped file: 23000-20000 = 3000; $3000/1024 = 3$ páginas.

STACK: 65535-62100 = 3435; $3435/1024 = 4$ páginas.

También sabemos la dirección de comienzo de cada sección.



TEXT: 0

DATA: 3000

mapped file: 20000

STACK: 62100

* es 3 porque se aproxima la unidad por exceso

Tam_pag = 1024B = 1KB
Tam_tabla_1^{er} = 8
Tam_tabla_2^{do} = 8

TAM
TEXT: 3 páginas
DATA: 9 páginas
mapped file: 3 páginas
STACK: 4 páginas

DIR_COMINEZO
TEXT: 0
DATA: 3000
mapped file: 20000
STACK: 62100

Tabla 1er nivel

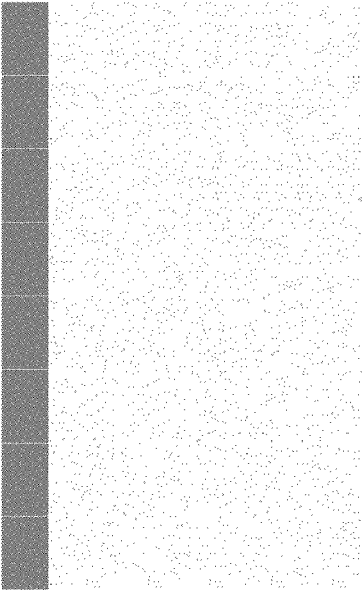


Tabla 2do nivel

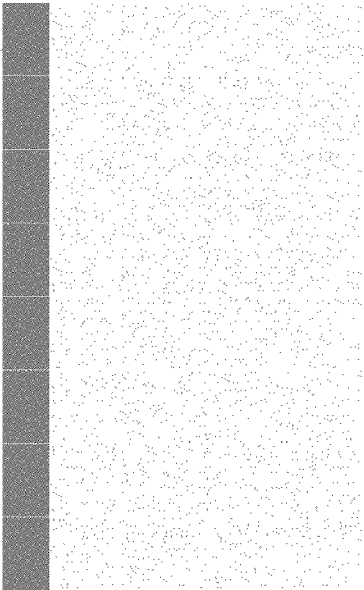
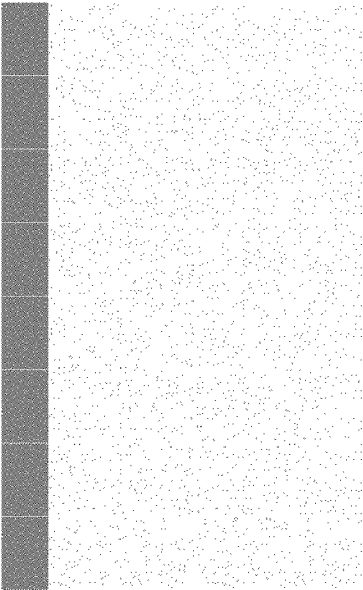


Tabla 2do nivel



Para saber donde va cada cosa...

Bueno pues para saber donde va cada cacho de trozo de programa basicamente hay que:

- 1) Saber donde empieza el cacho; ej 3000. Una vez sepas donde comienza tienes que dividir la dir_comienzo entre el $\text{tam_tabla_2}^{\text{er nivel}} * \text{tam_pag} = C + R$.
- 2) **$\text{DIR_COMIENSO} / (\text{tam_tabla_2}^{\text{er nivel}} * \text{tam_pag}) = C + (R)$.**
- 3) Esto es asi porque cada entrada de la tabla de primer apunta a una tabla de 2do nivel que almacena x paginas, por eso $x * \text{tam_pag}$ es el tamaño maximo que puede asignarse a una entrada de primer nivel.
- 4) El Cociente(**C**) es el indice de la tabla de primer nivel donde ha caido. Ahora dividimos el resto(**R**) entre tam_pag , la parte entera del resultado será el indice de la tabla de 2do nivel donde empieza a asignarse nuestro cacho.
- 5) Ahora desde esa pagina usamos tantas páginas como ocupe nuestra mi**da cacho de trozo de programa.
- 6) Si por casualidad esa pagina ya está usada, pues la siguiente, y si también está usada, pues la siguiente y si.....blablablablabla.

Vamos a hacer un ejemplo a ver si lo has pillao

Tam_pag = 1024B = 1KB

Tam_tabla_1^{er} = 8

Tam_tabla_2^{do} = 8

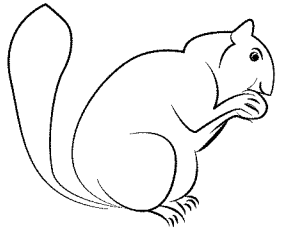


Tabla 1er nivel

0	1	1
1		
2		
3		
4		
5		
6		
7		

TAM

TEXT: 3 páginas

DATA: 9 páginas

mapped file: 3 páginas

STACK: 4 páginas

DIR_COMINEZO

TEXT: 0

DATA: 3000

mapped file: 20000

STACK: 62100

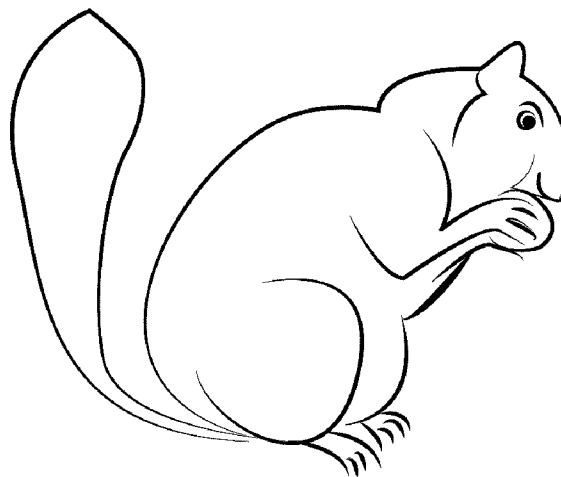


Ya lo sé, este ejemplo es una
pu** m**da

Tabla
2do nivel

0	10	1	rx-
1	11	1	rx-
2	12	1	rx-
3			
4			
5			
6			
7			

Hemos empezado
en el numero de
marco 10, porque
lo decía el
enunciado.



Para TEXT

$$0 / 8 * 1024 = 0 + (0) \text{ (ostia si?)}$$

$$0 / 1024 = 0 \text{ (xd)}$$

- ¿Cuanto ocupa?
- 3
- Pues pillas 3
- ok



VA,

OTRO EJEMPLO

Tam_pag = 1024B = 1KB

Tam_tabla_1^{er} = 8

Tam_tabla_2^{do} = 8

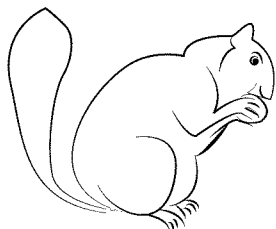


Tabla 1er nivel

0	1	1
1	2	1
2		
3		
4		
5		
6		
7		

TAM

TEXT: 3 páginas

DATA: 9 páginas

mapped file: 3 páginas

STACK: 4 páginas

DIR_COMINEZO

TEXT: 0

DATA: 3000

mapped file: 20000

STACK: 62100

0	10	1	rx-
1	11	1	rx-
2	12	1	rx-
3	13	1	r-w
4	14	1	r-w
5	15	1	r-w
6	16	1	r-w
7	17	1	r-w

0	18	1	r-w
1	19	1	r-w
2	20	1	r-w
3	21	1	r-w
4	0	0	
5	0	0	
6	0	0	
7	0	0	

Para DATA

$$3000 / 8 * 1024 = 0 + (3000)$$

$$3000 / 1024 = 2$$


- ¿Cuanto ocupa?
- 9
- Pues pillas 9
- ok

////////////////////////////////////

- oye, pero esque la dos de segundo nivel está usada por text
- Pues la siguiente!
- ok

////////////////////////////////////

Osea que desde la 2 de segundo palante es data, vaya



AHORA

ENTERA

Tam_pag = 1024B = 1KB
Tam_tabla_1^{er} = 8
Tam_tabla_2^{do} = 8

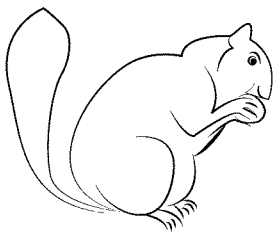


Tabla 1er nivel

0	1	1
1	2	1
2	3	1
3	0	0
4	0	0
5	0	0
6	0	0
7	4	1

TAM
TEXT: 3 páginas
DATA: 9 páginas
mapped file: 3 páginas
STACK: 4 páginas

DIR_COMINEZO
TEXT: 0
DATA: 3000
mapped file: 20000
STACK: 62100

1)

0	10	1	r-x
1	11	1	r-x
2	12	1	r-x
3	13	1	r-w
4	14	1	rw-
5	15	1	rw-
6	16	1	rw-
7	17	1	rw-

2)

0	18	1	rw-
1	19	1	rw-
2	20	1	rw-
3	21	1	rw-
4	0	0	
5	0	0	
6	0	0	
7	0	0	

3)

0	0	0	
1	0	0	
2	0	0	
3	22	1	rw-
4	23	1	rw-
5	24	1	rw-
6	0	0	
7	0	0	

4)

0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	25	1	rw-
5	26	1	rw-
6	27	1	rw-
7	28	1	rw-

Tam_pag = 1024B = 1KB
 Tam_tabla_1^{er} = 8
 Tam_tabla_2^{do} = 8



Tabla 1er nivel

0	1	1
1	2	1
2	3	1
3	0	0
4	0	0
5	0	0
6	0	0
7	4	1

TAM
 TEXT: 3 páginas
 DATA: 9 páginas
 mapped file: 3 páginas
 STACK: 4 páginas

DIR_COMINEZO
 TEXT: 0
 DATA: 3000
 mapped file: 20000
 STACK: 62100

1)

0	10	1	r-x
1	11	1	r-x
2	12	1	r-x
3	13	1	r-w
4	14	1	rw-
5	15	1	rw-
6	16	1	rw-
7	17	1	rw-

2)

0	18	1	rw-
1	19	1	rw-
2	20	1	rw-
3	21	1	rw-
4	0	0	
5	0	0	
6	0	0	
7	0	0	

3)

0	0	0	
1	0	0	
2	0	0	
3	22	1	rw-
4	23	1	rw-
5	24	1	rw-
6	0	0	
7	0	0	

4)

0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	25	1	rw-
5	26	1	rw-
6	27	1	rw-
7	28	1	rw-

RAM

Esta no xd

9	
10	TEXT
11	TEXT
12	TEXT
13	DATA
14	DATA
15	DATA
16	DATA
17	DATA
18	DATA
19	DATA
20	DATA
21	DATA
22	mapped_file
23	mapped_file
24	mapped_file
25	STACK
26	STACK
27	STACK
28	STACK

Así se vería en RAM...

Tam_pag = 1024B = 1KB
 Tam_tabla_1^{er} = 8
 Tam_tabla_2^{do} = 8

TAM
 TEXT: 3 páginas
 DATA: 9 páginas
 mapped file: 3 páginas
 STACK: 4 páginas

DIR_COMINEZO
 TEXT: 0
 DATA: 3000
 mapped file: 20000
 STACK: 62100



Tabla 1er nivel

0	1	1
1	2	1
2	3	1
3	0	0
4	0	0
5	0	0
6	0	0
7	4	1

1)

0	10	1	r-x
1	11	1	r-x
2	12	1	r-x
3	13	1	r-w
4	14	1	rw-
5	15	1	rw-
6	16	1	rw-
7	17	1	rw-

2)

0	18	1	rw-
1	19	1	rw-
2	20	1	rw-
3	21	1	rw-
4	0	0	
5	0	0	
6	0	0	
7	0	0	

3)

0	0	0	
1	0	0	
2	0	0	
3	22	1	rw-
4	23	1	rw-
5	24	1	rw-
6	0	0	
7	0	0	

4)

0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	25	1	rw-
5	26	1	rw-
6	27	1	rw-
7	28	1	rw-

Ahora traducir...

Pues pa' traducir, (casi)lo mismo, miras en que pagina cae tu dirección y el marco de ram que tenga asignado será su marco en MP.

Toneses con una vez que tengas tu marco de ram, como sabes cuanto ocupa cada marco(tam_pag) puedes calcular la dir_física de ese marco.

Después le sumas el offset y a correr.

- Que es el offset.
- Ahora lo vemos.
- ok.

Dir_lógica: 7428

$$7428 / (8 \cdot 1024) = 0 + 7428$$

$$7428 / 1024 = 7 + (\text{Resto}) \quad // \text{ este es el offset!}$$

El offset es 260 = 7428 % 1024, osea el resto

i_1er = 0

i_2do = 7

Offset = 260

Esto nos da el marco 17(mirar img)

Entonces la dir_física es:

$$\text{tam_marco} * \text{num_marco} + \text{offset}$$

$$\text{Osea, } 1024 \cdot 17 + 260 = 17668$$

La dirección física es 17668.

Bueno y con la otra dirección que te dice le enunciado, haces lo mismo. Cosas a tener en cuenta:

- Si en la tabla de páginas no esta cargada la página a la que hace referencia la dirección lógica, pues no se puede traducir, punto.
- Y nada más solo era eso xd.

FIN

