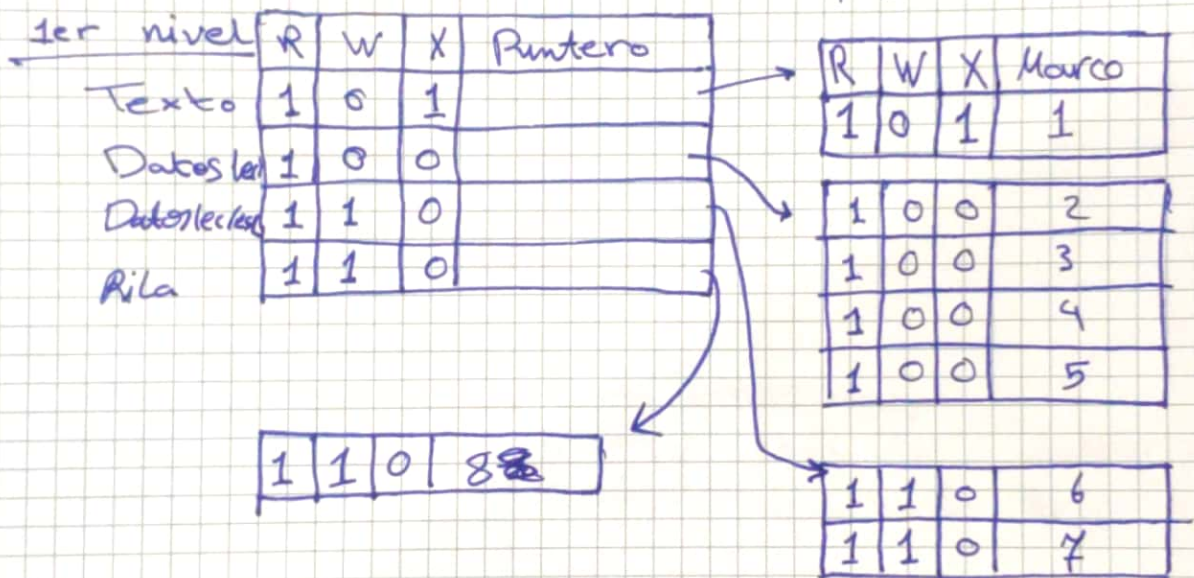


Problemas de Gestión de Memoria:

Problema 3.1:

a) Supongamos una tabla de páginas de dos niveles. En el primer nivel distinguiremos los diferentes segmentos y en el segundo las páginas de cada:

- Texto $3,4 \text{ kiB}$ 1 página
- Pila 200 B 1 página
- Datos lectura 14 kiB 4 páginas
- Datos lectura / escritura 6 kiB 2 páginas



b) Al iniciar la ejecución, no hay páginas en memoria principal.

Cuando ejecutamos el proceso, se dará un fallo de página y se traerá una página de memoria principal. La página del texto y a continuación, se irán produciendo más fallos de página que irán trayendo las otras páginas a los marcos.

c) La dirección que corresponde al push pertenece a una ~~buena~~ página inexistente y al intentar traerla detecta que se sale del rango y entonces la MMU genera una excepción por violación de memoria. El SO trata la excepción y aquí hay dos posibilidades, si hay espacio disponible, aumenta en una página el tamaño de la pila y si no, mata el proceso.

Problema 3.2:

a) ~~12~~

N° mágico
CP inicial
Tabla de secciones
Código
DCVI
DSVI
Tabla de símbolos lib. so biblio dinámica

	Desplaz.	Tam.
Código	1k	5k
DCVI	6k	6k
DSVI	—	2k
Tabla Smb.	12k	—

b)

8KB	Código
8KB	DCVI
4KB	DSVI
	⋮
4KB	Pila

2 págs } ~~Direcciones~~
 2 págs } según los datos de 1
 1 págs } apartado 2

1 págs

Direcciones virtuales de 24 bits

Espacio de direccionamiento virtual: $2^{24} = 16\text{MB}$

Las páginas tienen tamaño de $2^{12} = 4\text{KB}$

c) Es una fragmentación interna, ya que tendremos espacios ~~que~~ inútil en las páginas de cada región.

Problema 3.3

2) Primer nivel

	Validez	Ptr		validez	presente	Modificado	Referenciado	CDW	RC	Rt	Protección	Nº de Marco
0	1	→	0	1	1	0	1	0	0	0	rx	En memoria Cód. 1
1	0		1	1	0	0	0	0	0	1	rx	En SWAP Cód. 2
2	0		2	1	0	0	0	0	0	1	rw	En SWAP DSV1
3	0		3	1	0	0	0	0	0	1	rw	En SWAP DSV2
...	1	0	0	0	0	1	0	rw	DSV1
63	0		63

0												
1												
...												
63	1	1	1	1	0	0	0	0	rw	En memoria Pila		

→ por las variables declaradas en el main

b) la primera vez que una página es expulsada a SWAP, hay que reservar un espacio, ya que estamos sin preasignación. Las siguientes veces, se usará el espacio que se reservó la primera vez.

El bit de modificado, se inicializa por el sistema operativo cuando llevamos una página de disco a MP y se modifica cuando la MMU accede en escritura a dicha página de memoria.

Problema 3.4:

a) Tendremos dos regiones ~~mas~~ en adición de lo visto en el apartado 3.2.b. Esas regiones son el texto de la biblioteca y una región de datos.

DSVI										2 páginas	
Texto lib.so										2 páginas	
Datos lib.so										3 páginas	

Validar	Presente	Modif	Letra	Can	PC	RF	Prote	Nº de	Marco	T P nivel 1	
0	1									Validar	Ptr
1	1										
...	...										
63	1										

4	1	1	1	1	0	0	0	rw	En memoria	DSVI
5	1	1	0	1	0	0	0	rx	En memoria	Texto lib 1
6	1	0	0	0	0	0	1	rx	En disco (SWAP)	texto lib 2
7	1	1	?	1	0	0	0	rw	En memoria	Datos lib 1
8	1	0	●	0	0	0	1	rw	En disco (SWAP)	Datos lib 2
9	1	0	●	0	0	0	1	rw	En disco (SWAP)	Datos lib 3

...
1	1	1	1	0	0	0	0	rw	En memoria		

b) En principio, ~~al~~ crear el hijo, las zonas privadas del padre se marcan poniendo el can a 1. En el momento de que un proceso, el padre o el hijo escriban en esas zonas, se duplicaría la región y se escribiría en la página correspondiente (la página de datos de la ~~librería~~ biblioteca dinámica).

Problema 3.5:

a)

Cabecera: nº mágico, valor inicial de los registros del procesador

Código: 3573 B + cadena de texto

DCVI

int b = 50 → 4 B

static int e = 3 → 4 B

Tablas para montaje dinámico

b)

4kB

Código → 2 páginas = $2 \times 2\text{kB} = 4\text{kB}$

Compartida → rx → fuente: fichero ejecutable

DCVI

int b = 50 → 4 B

static int e = 3 → 4 B

} 1 página = 2kB

Privada, rw, fichero

6kB

DSVI

int z → 4 B

} 1 página = 2 kB

Privada, rw, rellenar ceros

8kB

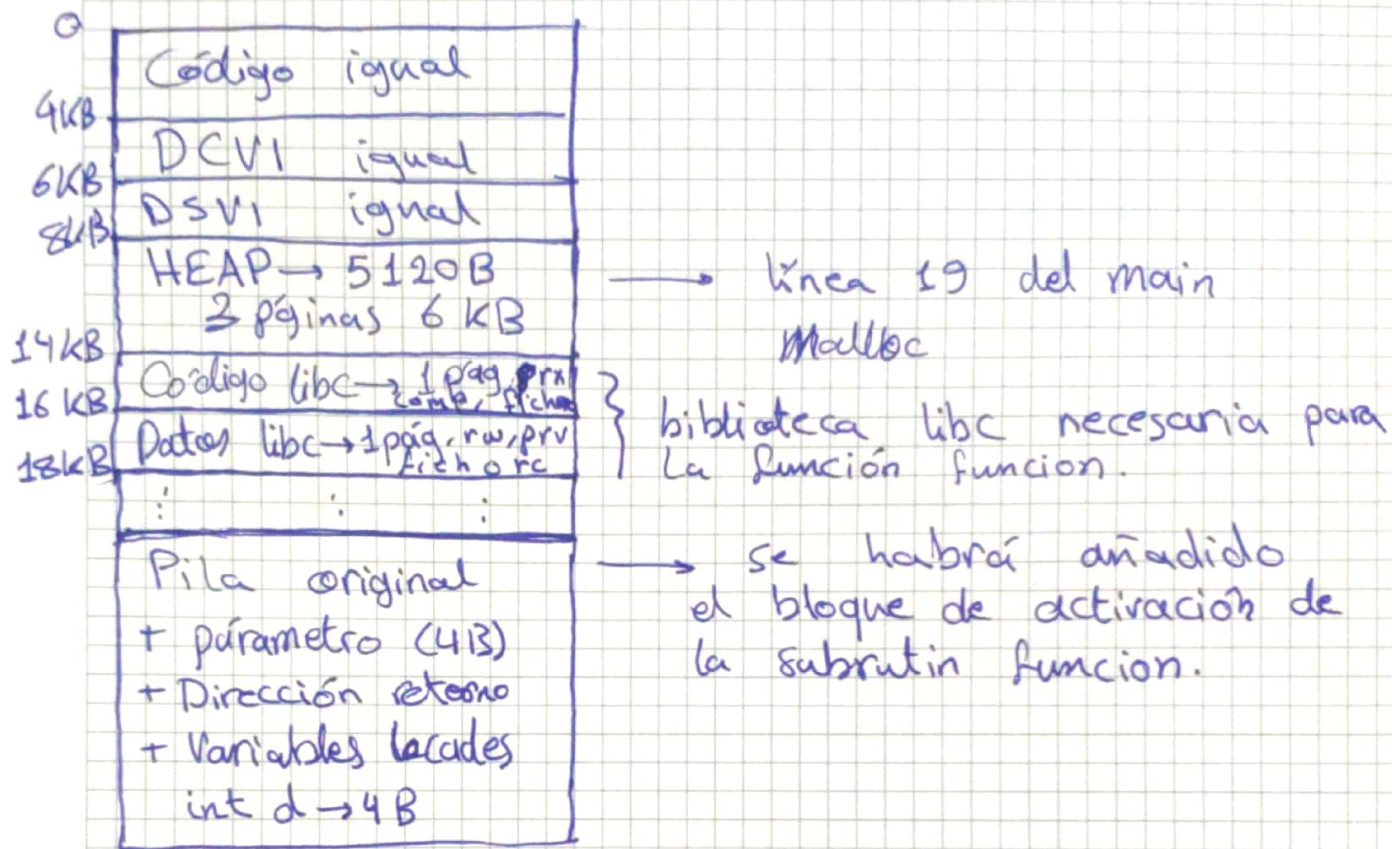
⋮

Pila → 1 página = 2 kB, Privada, rw,
rellenar con ceros, variables de entorno

Parámetros del main (no hay), dirección
de retorno, variables locales char*

Problema 3.6:

Línea 12: significa que se ejecutó el main hasta llamar a función:



Línea 21

En esa línea, ya se había terminado función y se ejecuta el free. La región de heap permanece igual de tamaño. En la pila se carga el bloque de activación de la subrutina función.

Problema 3.4:

- a) Texto compartido \rightarrow 7 páginas
Datos A \rightarrow 5 páginas
Datos B \rightarrow 5 páginas
Texto bib compartido \rightarrow 12 páginas
Datos bib A \rightarrow 3 páginas
Datos bib B \rightarrow 3 páginas.
Fichero M compartido \rightarrow 7 páginas.
Pila A \rightarrow 3 páginas
Pila B \rightarrow 7 páginas
En total so 52 ^{marcos} _{de} páginas.

b)