

Juan Valentín Guerrero Cano 45338112Y

$$h=8 \quad L=1 \quad M=1 \quad N=2$$

Tema 3

① a) dirección de 10 / 10 / 12 bits.

Código: INICIO: 4096 · 2 ~~8192~~ FIN: 12287
8192

DATOS: INICIO: 12288 FIN: 18481

PILA HERBA: BASE: $2^{30} - 2^{13} + 12 \cdot 2^{10} - 1$ Tam: $2048 + 1024$

PILA MAIN: $2^{30} - 1$ tam: $1204 \cdot 8 + 4096$

PILA HERBA: INICIO: $2^{30} - 2^{13} + 12 \cdot 2^{10} - 1 - 2048 - 1024$

FIN: $2^{30} - 2^{13} + 12 \cdot 2^{10} - 1$

PILA MAIN: INICIO: $2^{30} - 1 - 1204 \cdot 8 - 4096$

FIN: $2^{30} - 1$

b)

TP 1°N → 2° entornos (1014)

TP 2°N → " " "

TP 2°N (0)

RAM	
0	TP 2°(0)
1	
...	
1023	TP 2°(1023)
1024	CÓDIGO
1025	DATOS 1
1026	DATOS 2
1027	PILA HEBRA
	P.
1028	MAIN 1
1029	MAIN 2
1030	MAIN 3
1031	MAIN 4

TP 1°N	
0	
1	
...	
1023	

0	
1	
2	CÓDIGO
...	
1023	

TP 2°N (A)	
0	
1	
...	
1023	

TP 2°(1023)	
0	
1	
...	
1023	

TAM.
PÁGINA: 4KB

Luego → código : 1 página

DATOS : 2 páginas

PILA HEBRA : 1 página

PILA MAIN : 4 página

Luego

TP 2°N (0):

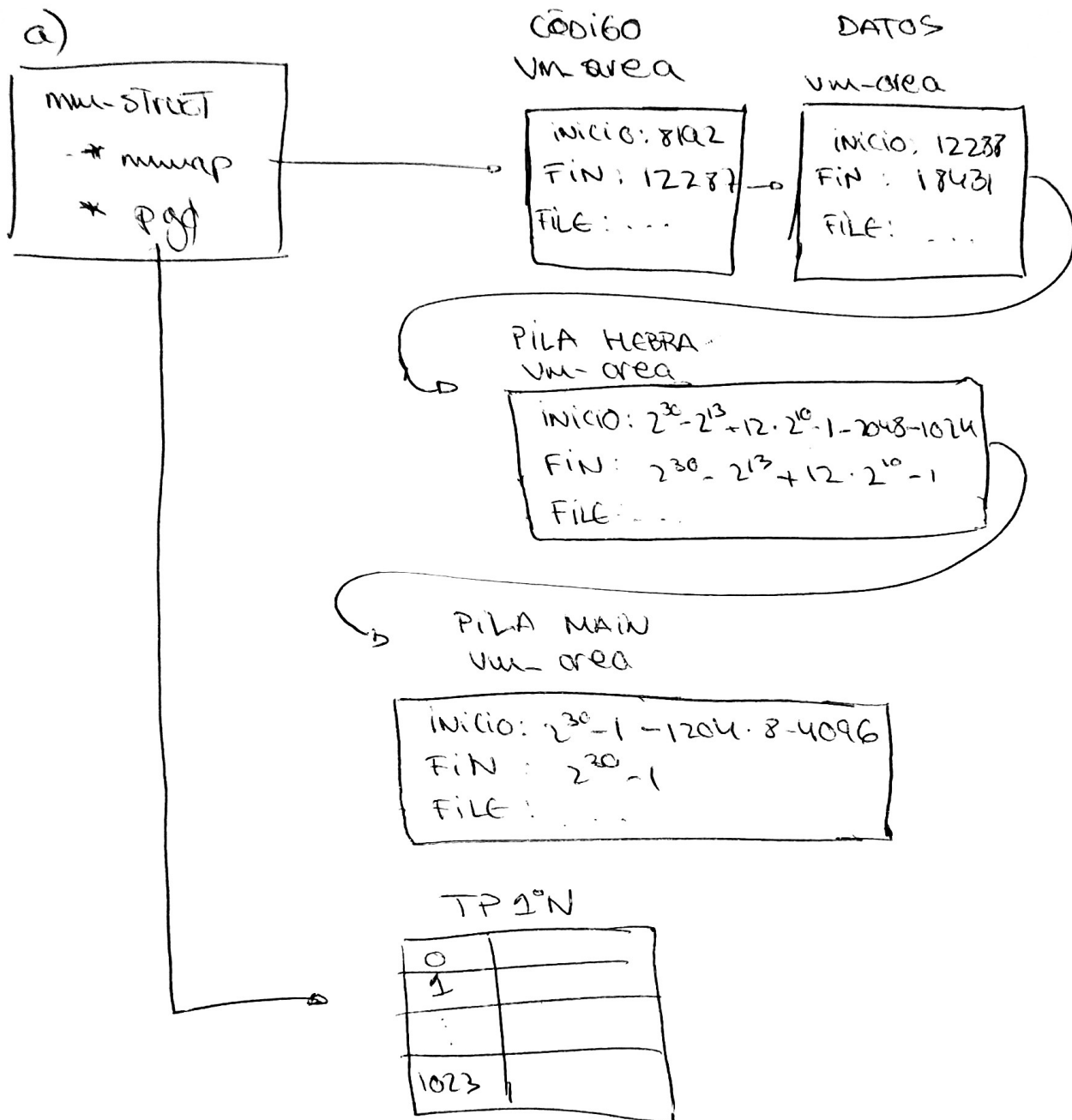
0	
1	
2	CÓDIGO
3	DATOS 1
4	DATOS 2
5	
...	
1023	

HABRÍA DE SITUAR

LA PILA HEBRA y LA PILA MAIN (NO ME DATIENPO).

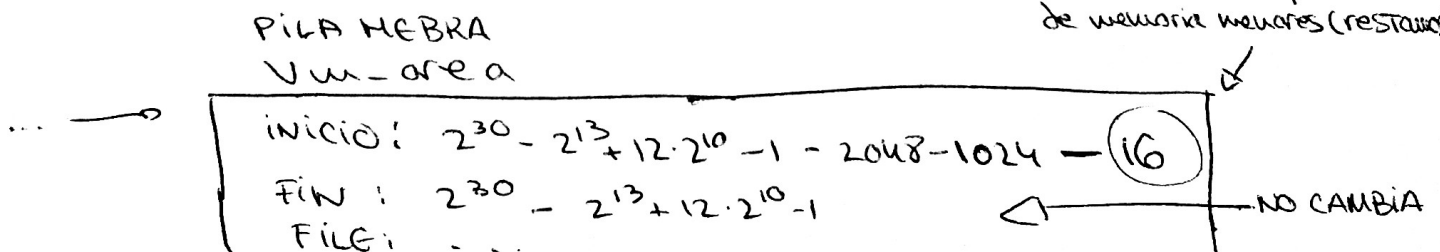
②

a)



b) Si la hebra empuja su pila 16 B eso corresponderá a aumentar el Tam de la pila 16 B luego el vmm-area de la pila de la hebra quedaría:

como la pila crece hacia posiciones de memoria menores (restamos)



Tema 4

Juan Valentin Guerrero Cano

45338112Y

K=8 L=1 M=1 N=2

3

a)

I1 = fd1 = open(..., O_RDWR);

I2 = fd2 = open(..., O_RDWR);

I3 = read(fd2, buffer2, M);

I4 = read(fd2, buffer2, M);

I5 = fork();

I6 = write(fd1, buffer2, 5);

b)

DESCRIPTORES
ARCHIVO

0	STDIN
1	STDOUT
2	STDERR
3	fd1
4	fd2

Hijo

0	STDIN
1	STDOUT
2	STDERR
3	fd1
4	fd2

TABLA
ARCHIVOS ABIERTOS

CURSOR	REFERENCIAS	PERMISOS
10	2	RW-
28	2	RW-

Tabla
dentro

datos	INODO
datos CSV	700
datos TXT	600

Tabla inodos

inodo	metadatos
700	...
600	...

Juan Valentin Guerrero Cano 453331124

④ FAT se basa en la asignación
no contigua mediante enlazamiento.
En nuestra memoria tendremos
un bloque que almacena nuestra Tabla FAT

EXT2 se basa en la asignación
por indexación, es decir, índices que contendrán
información acerca de los metadatos del
archivo y los bloques donde se almacena

↳ Tabla de
indexación combinada