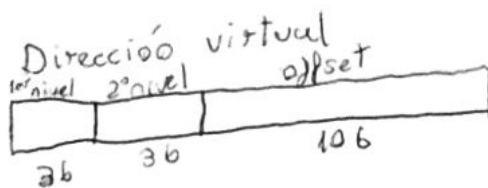
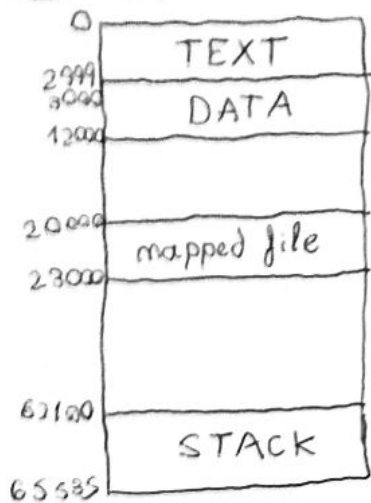


Ejercicios propuestos

Francisco Javier Bolívar Expósito

②

Mapa de memoria



tamaño página/marco = $2^{10} B = 1KB$

entradas por tabla = 2^3

páginas cargadas en MP a partir de la dirección $10 \cdot 2^{10} = 10240$

tamaño por tabla = $2^3 \cdot 3B = 24B$

Tabla de páginas

| Dir. base de marco | V/P | Prot. | Mod. |
|--------------------|-----|-------|------|
| 11264 | 1 | | |
| 12288 | 1 | | |
| 13312 | 1 | | |
| - | 0 | | |
| - | 0 | | |
| - | 0 | | |
| - | 0 | | |
| 14336 | 1 | | |

| Dir. base de marco | V/P | Prot. |
|--------------------|-----|-------|
| 0 | 1 | r-X |
| 1 | 1 | r-X |
| 2 | 1 | rwx |
| 3 | 1 | rW- |
| 4 | 1 | rW- |
| 5 | 1 | rW- |
| 6 | 1 | rW- |
| 7 | 1 | rW- |

El mapa de memoria no debería usar la misma página para distintas regiones.

| Dir. base de marco | V/P | Prot. |
|--------------------|-----|-------|
| 8 | 1 | rW- |
| 9 | 1 | rW- |
| 10 | 1 | rW- |
| 11 | 1 | rW- |
| 12 | 0 | - |
| 13 | 0 | - |
| 14 | 0 | - |
| 15 | 0 | - |

| PV | Dir. base de marco | V/P | Prot. |
|----|--------------------|-----|-------|
| 56 | - | 0 | - |
| 57 | - | 0 | - |
| 58 | - | 0 | - |
| 59 | - | 0 | - |
| 60 | 31744 | 1 | rW- |
| 61 | 32768 | 1 | rW- |
| 62 | 33792 | 1 | rW- |
| 63 | 34816 | 1 | rW- |

| Dir. base de marco | V/P | Prot. |
|--------------------|-----|-------|
| 16 | 0 | - |
| 17 | 0 | - |
| 18 | 0 | - |
| 19 | 0 | - |
| 20 | 1 | rW- |
| 21 | 1 | rW- |
| 22 | 1 | rW- |
| 23 | 0 | - |

Traducción de direcciones virtuales

Procesamiento de Datos Externos

$$7428 = 000|111|0100000100$$

$$\text{dir.fis.} = 22528 + 260 = 22788$$

$$19425 = 010|010|1111100001$$

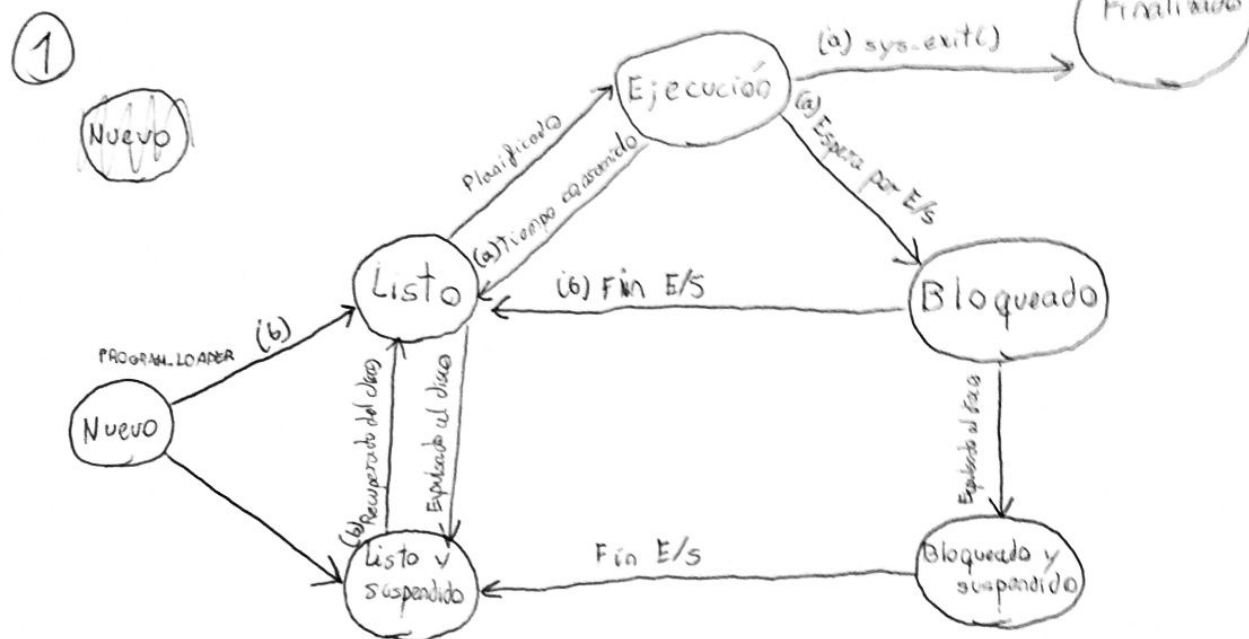
$$\text{dir.fis.} = 24576 + 998 = 25569 \text{ Dirección no válida}$$

$$45000 = 101|011|1111001000$$

En la TP podemos ver que esta dirección no es válida

$$63500 = 1111110|0000001100$$

$$\text{dir.fis.} = 33792 + 12 = 33804$$



- Las transiciones de $L \rightarrow SL$ y $B \rightarrow SB$ las realiza el swapper cuando decide sacar o cargar un programa de/en memoria.
- Un proceso puede pasar de ejecutándose a bloqueado por una E/S provocada por `read()` o `write()` o por sincronización padre/hijo provocada por `wait()`.
- Las transiciones de $B \rightarrow L$ y $SB \rightarrow SL$ suceden cuando termina el evento o suceso por el que esperaba bloqueado con una RSI.
- Por cualquier razón por la que el proceso termine se acaba ejecutando `sys_exit()` y se saca de ejecución.
- Cuando se termina el quantum del proceso pasa de $E \rightarrow L$.
- En todas las transiciones marcadas con (a), al dejar de ejecutarse un programa, se llama a `context_switch` que hace `schedule()` y `dispatch()`.
- En todas las transiciones marcadas con (b), se hace CPU preemption y si hay otro proceso con mayor prioridad pasa a ejecutar y el que estaba en E pasa a Listo. En caso contrario se encola en listas.