

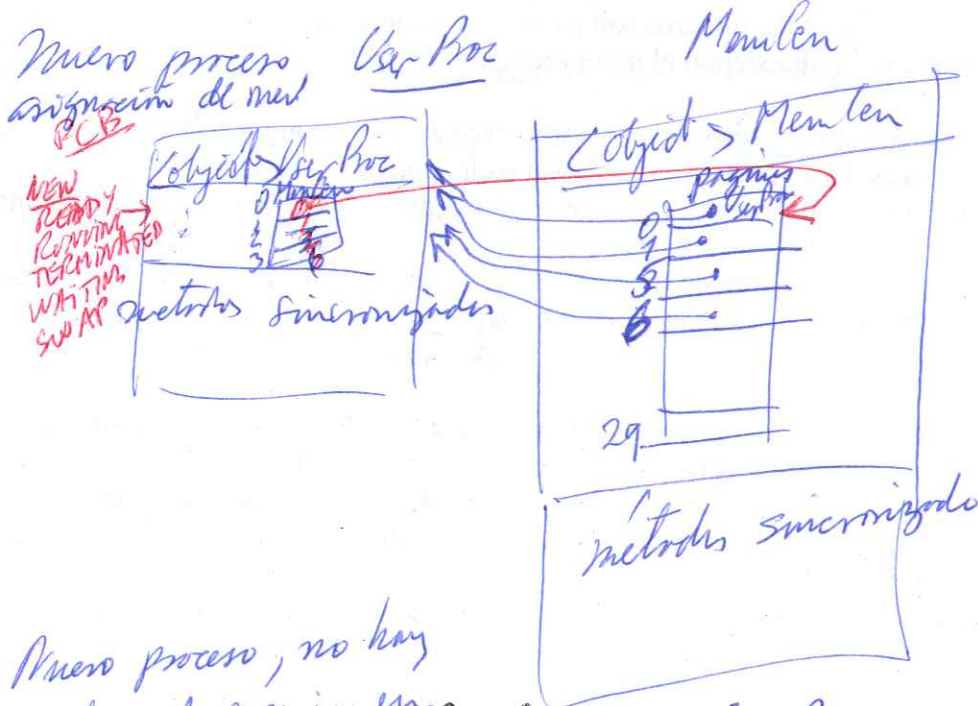
Dado que el sistema operativo es una aplicación concurrente, significa que muchas actividades llevándose a cabo "al mismo tiempo" o sea concurrentemente, debe evitar el tener grandes actividades de manera seriada. Además de lo solicitado en la práctica pasada debe observar:

- Que el sistema no debe terminar hasta que el último programa haya sido ejecutado.
- Cada vez que haya un cambio de estado deberá imprimirse tanto en la terminal como en el archivo de log, dicho cambio.
- Cada vez que un proceso sea asignado a memoria, o desasignado, deberán indicarse las páginas asignadas o las páginas desasignadas, imprimiendo el PageTable del proceso.

Bien, seguimos diciendo que parece que los cinco estados pudieran ser THREADS, que se arrancan y se mantienen según FCFS. También tenemos una serie de objetos comunes como:

- La Cola de READY
- La referencia al archivo de log.
- Algún otro objeto común que tenga el estado del sistema que permita al *main* sólo terminar cuando corresponda.

A lo mejor estos objetos comunes hay que aplicarles en algún momento *Exclusión Mutua* donde las secciones críticas deben estar en los métodos del objeto común. También se pudieran ayudar de los métodos para *Deadlocks*, pero esto déjenlo como última solución.



Lista de memoria libre

0	0
1	1
2	2
...	...
29	29

Nuevo proceso, no hay suficiente espacio en memoria

