Hardware ayuda a SO x interrupciones

Creación de procesos
- Procesos son creados x otros procesos.
- Árbol de procesos.

swapper → proceso padre (PID 0) lanza al 1er proceso como tal → init

init es padre de todos los procesos.

Al crear proceso
- se crea PCB, se le asigna PID (proceso : 0) → new
- se le da memoria → stack, text, datos → ready
- crear estructuras auxiliares de datos
  - fork → copia contexto, regiones de datos, text y stack.
    - se crea PCB con PID y se copian datos
    - P. hacer proceso hijo
  - a. padre se ejecute en simultáneo con el hijo
  - b. padre puede tener q' esperar al hijo para seguir

Hijo es duplicado del proceso padre (en unix) en windows se crea vacío

En unix → 2 system calls.
- fork → crea nuevo proceso igual al q' llama.
- execve() → carga nuevo programa en espacio de direc. → HACELO
  - carga el contexto, sustituye imagen del proceso padre x la del hijo.
  - se le pasa instrucción

windows → CreateProcess() → crea y carga
- 1 sys call.

Padre no tiene PID 0, la función fork devuelve 0 en el padre.

Con el exit se le devuelve control al SO.

wait cuando padre tiene q' esperar a hijos. → cuando hijo hace exit se sigue.

Padre puede matar a hijos con kill
- A veces se mata de una
- a veces se van terminando en cascada.

Procesos
- independientes: proc. no afecta ni es afectado x ejecución de otros → fork + execve
- cooperativos: se comunica con otro. → en general solo fork.

mainframe