\*process sheduling

Multiprograming: 讓CPU隨時都在執行程式

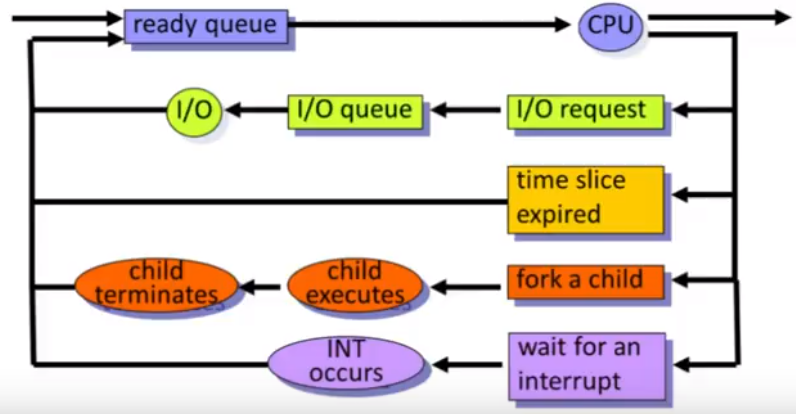
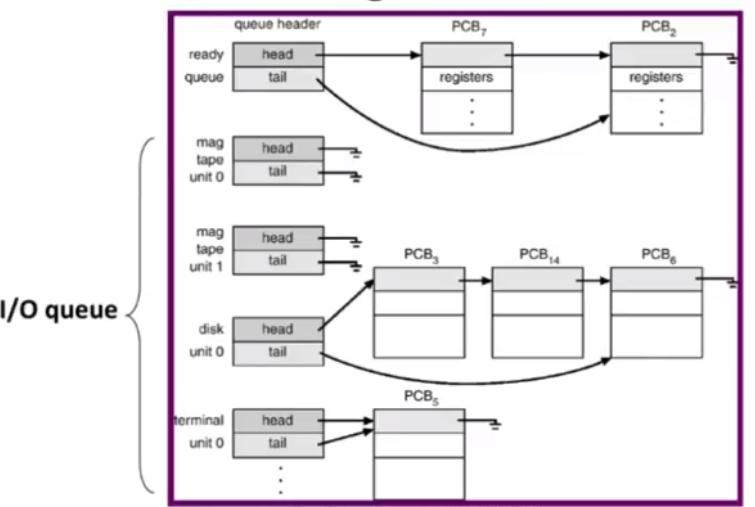
Time sharing: 讓user能interact

\*Process schedule Queue種類

Job queue (new state) (排隊進入memory)

Ready queue (ready state) (排隊進入 CPU) (最主要的)

IO queue(waiting queue) (不同device有不同queue)



(PCB linked list) (fork 就是開個新process)

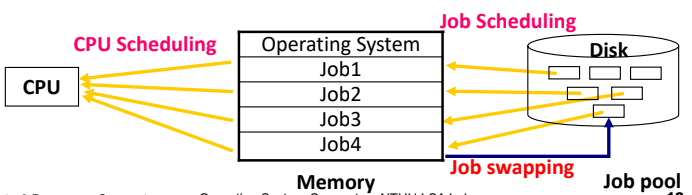
Scheduler種類

Long term(job scheduler): (job queue; new->ready)，決定誰要進入記憶體 (近年memory夠大，有些不需要此scheduler了)

Short term(cpu scheduler): (ready queue; ready->run)，決定誰進入CPU

Midium term ( ready -> wait)， 從memory放回disk，也就是swap

(vitural memory)



\* Long term scheduler

控制multiprogramming的程度，太低CPU會idle，太高會一直在做IO，

(要送入工作)平衡到CPU與IO同時都在工作

現代Unix:這個工作已經由Midium term scheduler取代 (memory夠大，可以一次load到memory沒關係)

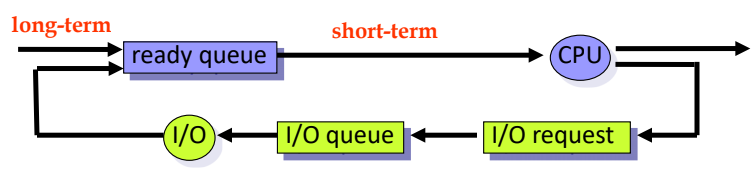
\*short term scheduler:

使用率非常高(time sharing)

演算法必須要有效率

Def overhead = 選A程式時間 / (選A時間+執行A時間)

同時也必須注重平均，不能某些程式等很久



\*midium term

負責swap out/swap in between Disck and Memory

目的: 配置記憶體空間，管理mutiprogramming比例

