

Scheduling di breve termine in Unix tradizionale

(fino a 4.3BSD e SVR3)

- a code multiple, round-robin
- ogni processo ha una priorità di scheduling; numeri più grandi indicano priorità minore
- Feedback negativo sul tempo di CPU impiegato
- Invecchiamento dei processi per prevenire la starvation
- Quando un processo rilascia la CPU, va in *sleep* in attesa di un *event*
- Quando l'evento occorre, il kernel esegue un *wakeup* con l'indirizzo dell'evento e il processo *asleep* in testa alla coda sull'evento viene messo nella coda di ready (*)
- I processi che erano in attesa di un evento in modo kernel rientrano con priorità *negativa* e non soggetta a invecchiamento

248

- (*) Nota:
 - In Unix tradizionale gli eventi sono mappati in indirizzi del kernel
 - Eventi diversi possono essere mappati nello stesso indirizzo kernel (ad es. attesa su un buffer e attesa di completamento di I/O vengono mappati sull'indirizzo del buffer)
 - Più processi possono essere asleep sullo stesso indirizzo kernel
 - Il kernel non tiene traccia di quanti processi sono in attesa
 - A seguito di una *wakeup tutti i processi in attesa* su un evento vengono risvegliati e spostati nello stato Ready
 - Tuttavia molti di essi torneranno subito asleep