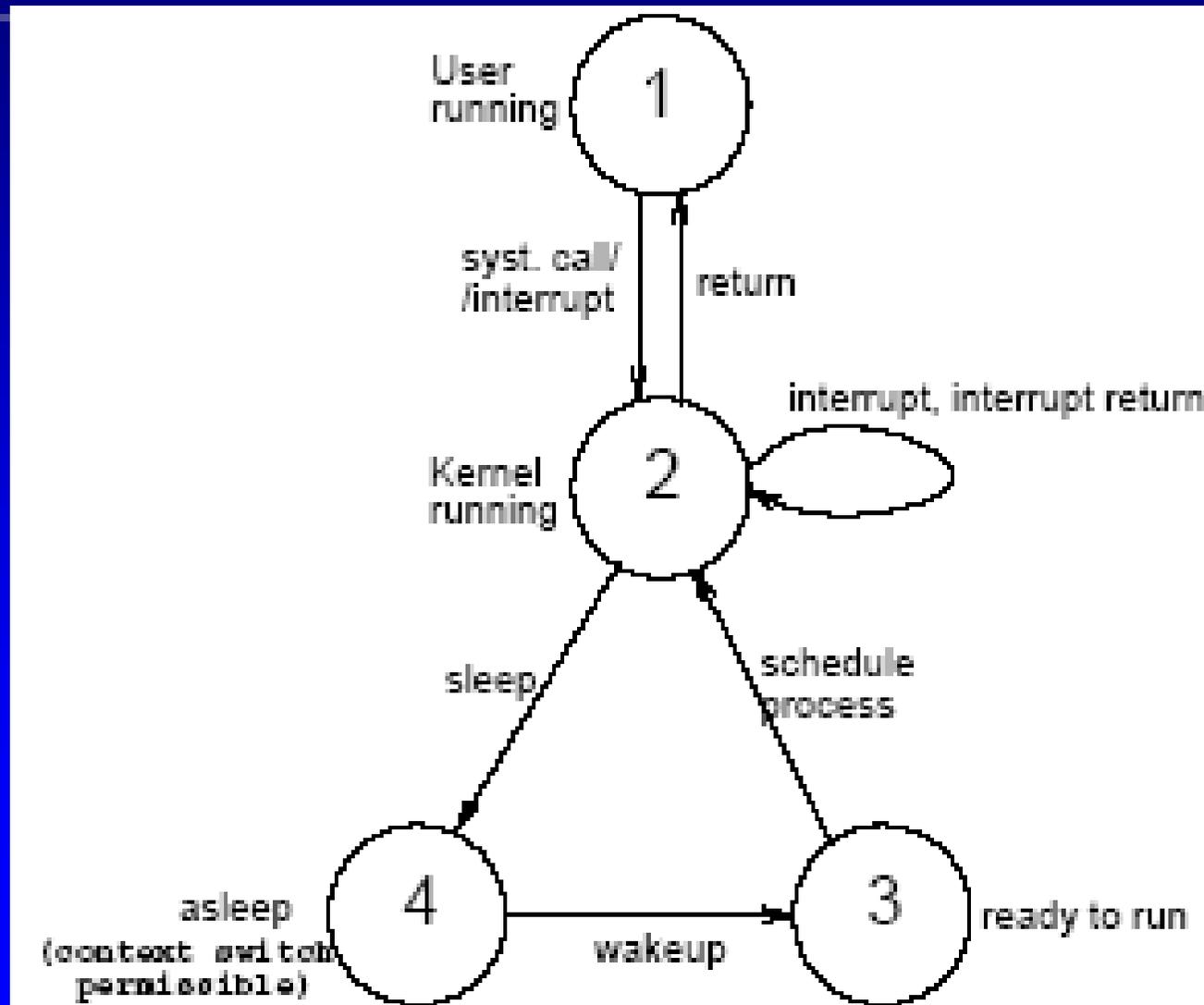


Processi in UNIX

I processi

- Un processo è un programma in esecuzione
- Un processo è una sequenza di byte che la CPU interpreta come istruzioni (text e dati)
- Caratteristiche
 - Organizzazione gerarchica
 - PID (assegnato dal sistema)
 - Priorità (assegnata dal sistema)
- Evolve attraverso un certo numero di stati

Lo stato dei processi



Lo stato dei processi

- Il comando **ps** permette di analizzare lo stato di un processo
- Numerose opzioni con vari livelli di informazioni
- Output base

PID	TTY	TIME	CMD
3490	pts/3	00:00:00	bash
3497	pts/3	00:00:00	ps

PID Process Identifier

TTY terminale da cui il processo è eseguito

TIME tempo totale di esecuzione

CMD comando eseguito corrispondente

Lo stato dei processi

- Opzioni principali
 - e: visualizza tutti i processi (tutti gli utenti)
 - x: visualizza anche i processi in background
 - u: visualizza info sulle risorse
- Stati di un processo:

R	In esecuzione/esequibile
T	Stoppato (es. ^Z)
S	Addormentato
Z	Zombie
D	In attesa I/O non interrompibile

Zombie e daemon

- **Zombie**: processo che ha terminato o è stato ucciso, ma non riesce a segnalare l'evento al padre
- **Daemon** (demone): processi che girano persistentemente in background e forniscono servizi al sistema (es: la posta elettronica o la gestione delle risorse)
 - Sono disponibili in qualunque momento per servire più task o utenti

Gestione dei processi

- I processi normalmente eseguono in **foreground** e hanno tre canali standard connessi al terminale
- I processi attivati con **&** eseguono in **background** e sono provi di stdin
- Un processo in foreground può essere sospeso con **^Z**

Gestione dei processi

- I processi sospesi possono essere continuati sia in foreground che in background
- I processi in background possono essere riportati in foreground
- Il comando **at** permette di lanciare e controllare processi batch

Gestione dei processi – Comandi

- **jobs [-l]**
elencare i job in background o sospesi
- **bg [job-id]**
eseguire i job specificati in background
- **fg [job-id]**
eseguire i job indicati in foreground
- **kill [-signal] process-id**
- **kill [-signal] %job-id**
manda un segnale al processo/job indicato (i più comuni sono 1 HUP e 9 KILL)
- **kill -l**
elencare tutti i segnali disponibili