

AVVIO DI UN SISTEMA OPERATIVO

• BOOT STRAP

Sequenza di processi $P_1 \rightarrow P_2 \rightarrow \dots$

P_1 : **BIOS** (basic i/o system), salvato in memoria nella MB

P_2 : **BOOT LOADER** che cerca il sistema operativo sul disco ed eseguirne il codice di avvio o far scegliere all'utente quale OS avviare tra quelli installati

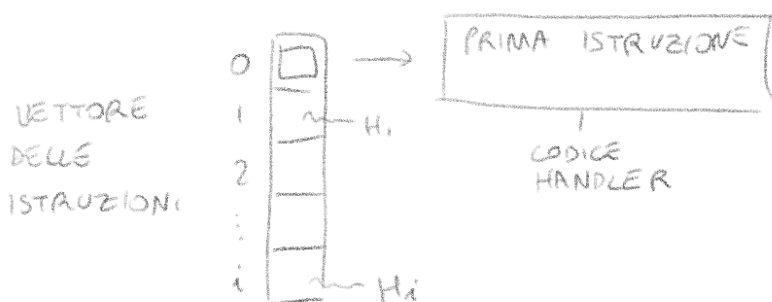
OS: programma **REATTIVO** e non **PROATTIVO**

EVENTO: notifica che è occorsa qualcosa (es. click del mouse)

EVENTI HW: INTERRUPT

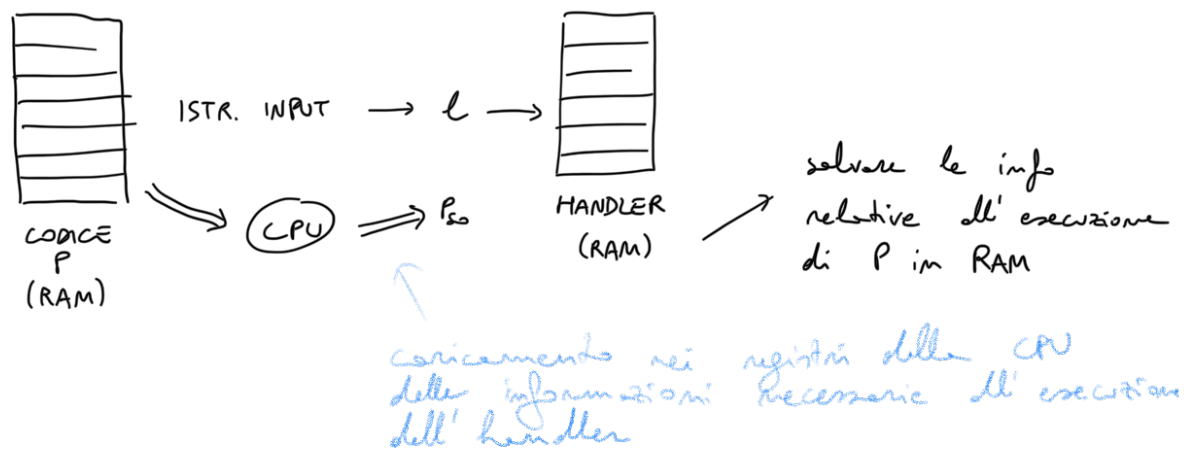
EVENTI SW: TRAP

ad ogni evento è associato un **HANDLER**



DISPATCHER usa l'indice del vettore per identificare l'handler corretto da usare

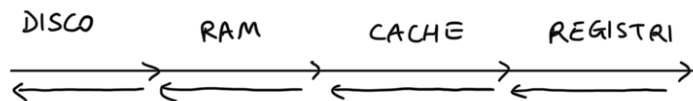
PROCESSO P : n



CONTEXT SWITCH :

II RISORSA GESTITA DAL OS

MEMORIA RAM



III RISORSA

DEVICE (DISPOSITIVI)

Il collegamento tra computer e device è gestito da:

- controller (HW)
- driver di dispositivo (SW)

È compito dell'OS:

Identificare **ASTRAZIONI** per ogni tipo di elemento da gestire

PROCESSI

È necessario RAPPRESENTARLI con

IDENTIFICATORE
CODICE
STACK

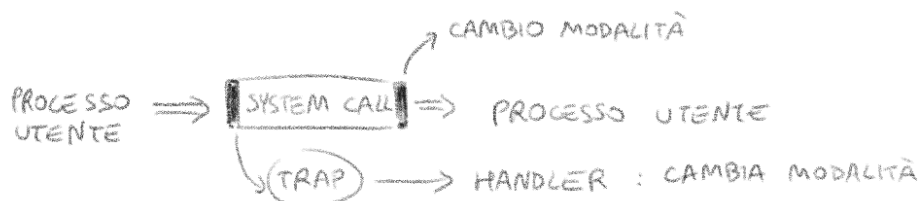
DUAL MODE

INSTRUCTION	SET
ESEGUIBILE DA TUTTI	ESEGUIBILE SOLO DALL'OS

L'architettura deve fornire un bit di modalità, e secondo del suo valore si può essere in:

- MODALITÀ UTENTE
- MODALITÀ KERNEL

Per invocare un'istruzione alla quale un programma non può accedere direttamente chiama una **SYSTEM CALL** che cambia il bit di modalità



SYSTEM CALL

1. CONTROLLO DEI PROCESSI
2. GESTIONE DEI FILE
3. GESTIONE DEI DEVICE
- 4.
5. COMUNICAZIONE TRA PROCESSI
- ...