

FONDAMENTI DI INFORMATICA

Ing. Davide PIERATTONI

Facoltà di Ingegneria
Università degli Studi di Udine

Gestione dei processi

Nota di Copyright

Questo insieme di trasparenze (detto nel seguito slide) è protetto dalle leggi sul copyright e dalle disposizioni dei trattati internazionali. Il titolo ed i copyright relativi alle slides (ivi inclusi, ma non limitatamente, ogni immagine, fotografia, animazione, video, audio, musica e testo) sono di proprietà degli autori prof. Pier Luca Montessoro e ing. Davide Pierattoni, Università degli Studi di Udine.

Le slide possono essere riprodotte ed utilizzate liberamente dagli istituti di ricerca, scolastici ed universitari afferenti al Ministero della Pubblica Istruzione e al Ministero dell'Università e Ricerca Scientifica e Tecnologica, per scopi istituzionali, non a fine di lucro. In tal caso non è richiesta alcuna autorizzazione.

Ogni altro utilizzo o riproduzione (ivi incluse, ma non limitatamente, le riproduzioni su supporti magnetici, su reti di calcolatori e stampe) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte degli autori.

L'informazione contenuta in queste slide è ritenuta essere accurata alla data della pubblicazione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, reti, ecc. In ogni caso essa è soggetta a cambiamenti senza preavviso. L'autore non assume alcuna responsabilità per il contenuto di queste slide (ivi incluse, ma non limitatamente, la correttezza, completezza, applicabilità, aggiornamento dell'informazione).

In ogni caso non può essere dichiarata conformità all'informazione contenuta in queste slide.

In ogni caso questa nota di copyright e il suo richiamo in calce ad ogni slide non devono mai essere rimossi e devono essere riportati anche in utilizzi parziali.

Processi

- Chiameremo **PROCESSO**, **TASK** o **JOB** un

programma in esecuzione

- Un processo è identificato da un PID (Process Identifier - un numero)
- Ad ogni processo è associato un Process Control Block (PCB)

Process Control Block

- Contiene tutte le informazioni necessarie al funzionamento del processo:
 - identificatore del processo e del proprietario
 - stato di avanzamento
 - contesto (registri e flag della CPU)
 - priorità e altre informazioni per lo *scheduling*
 - memoria allocata
 - informazioni contabili (es. CPU usata, limiti, ecc.)
 - risorse allocate (dispositivi hw, file, ecc.)
 - ...

Programmi in memoria

- **Monoprogrammazione:** un solo programma alla volta può occupare la memoria
- **Multiprogrammazione:** più programmi possono essere presenti contemporaneamente in memoria

Esecuzione dei programmi

- **Monotasking**: un solo programma in esecuzione per volta
- **Multitasking**: più programmi in esecuzione simultaneamente

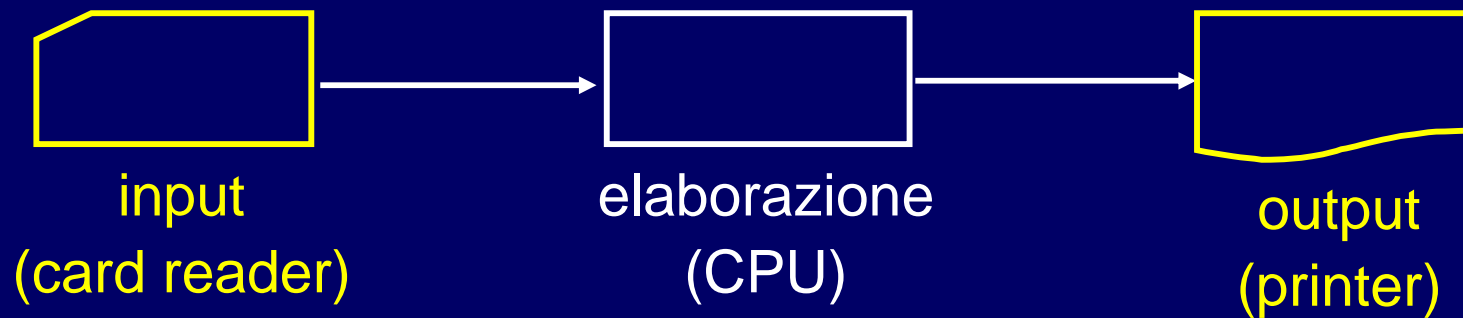
Utenti

- **Monoutente** (“single user”): un singolo utente per volta
- **Multiutente** (“multiuser”): più utenti possono lavorare contemporaneamente

Monitor

- Per terminali intelligenti e console di controllo di elettrodomestici ed apparecchiature
 - semplice
 - monotasking, monoutente
 - legge comandi dalla tastiera e invia messaggi sul video o sul display
 - normalmente residente in ROM

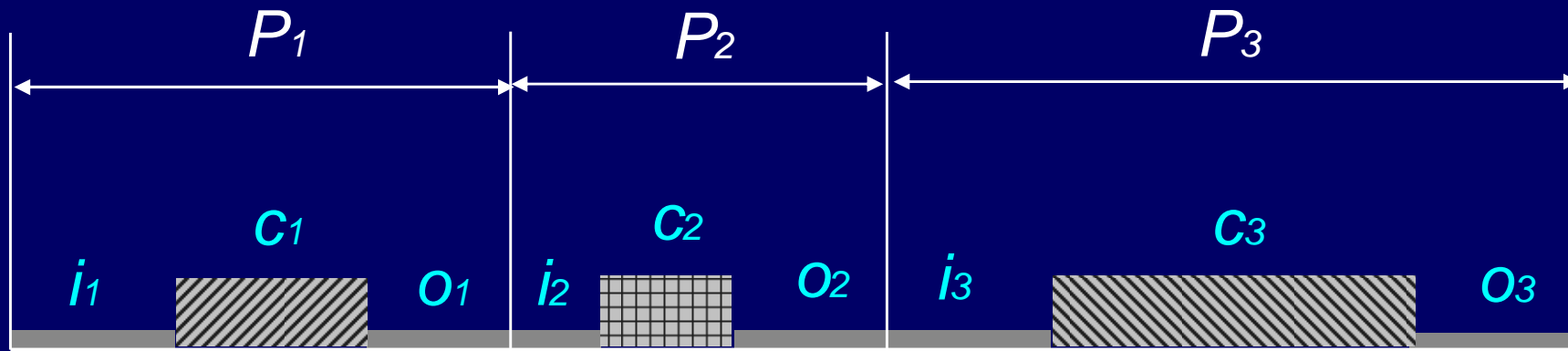
Batch



Ancora in uso per applicazioni particolari, in cui il multitasking non è possibile.

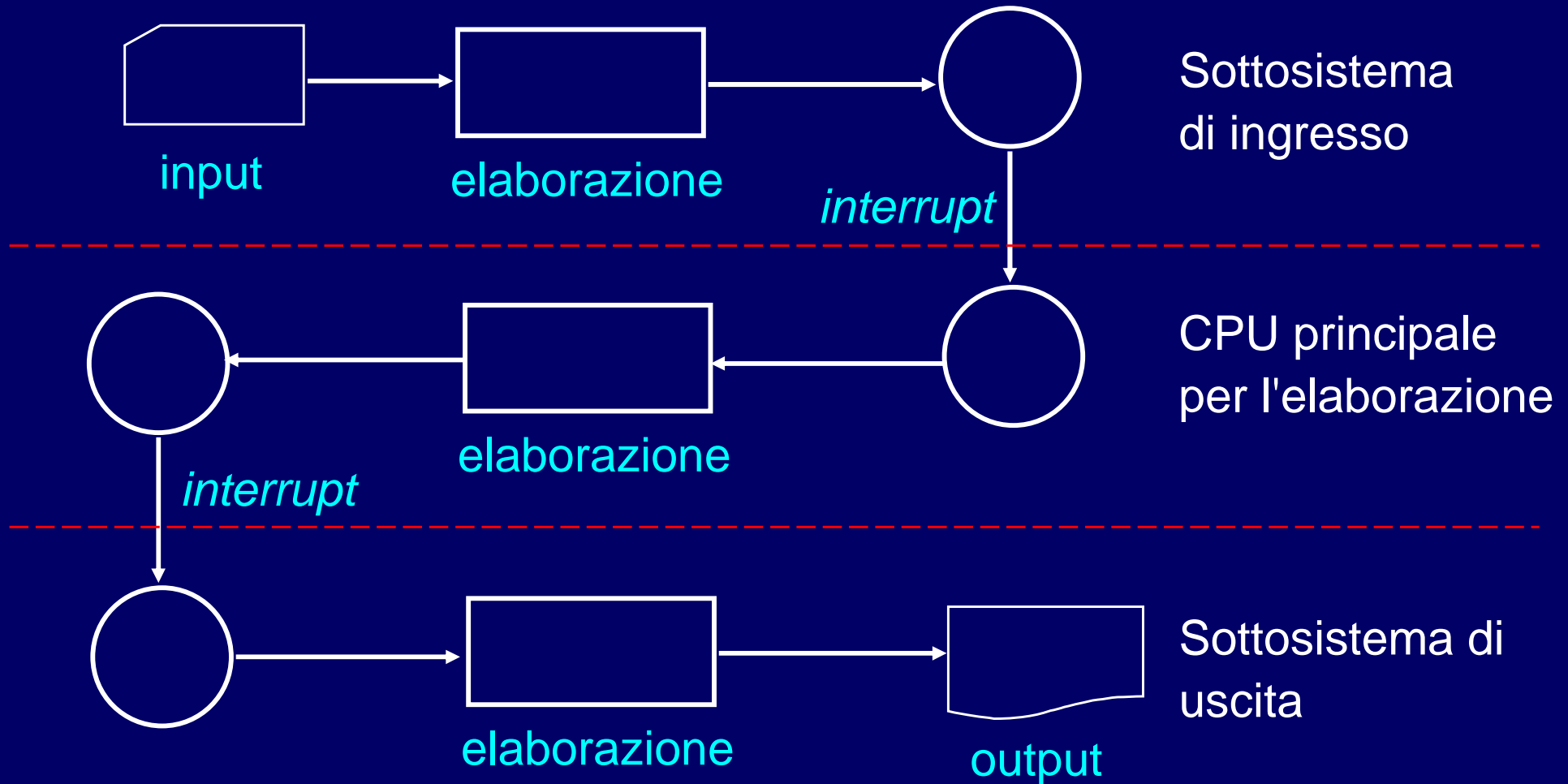
Quando le CPU erano molto costose e non esisteva ancora la tecnologia per il multitasking, permetteva di massimizzare l'utilizzo dei calcolatori

Batch

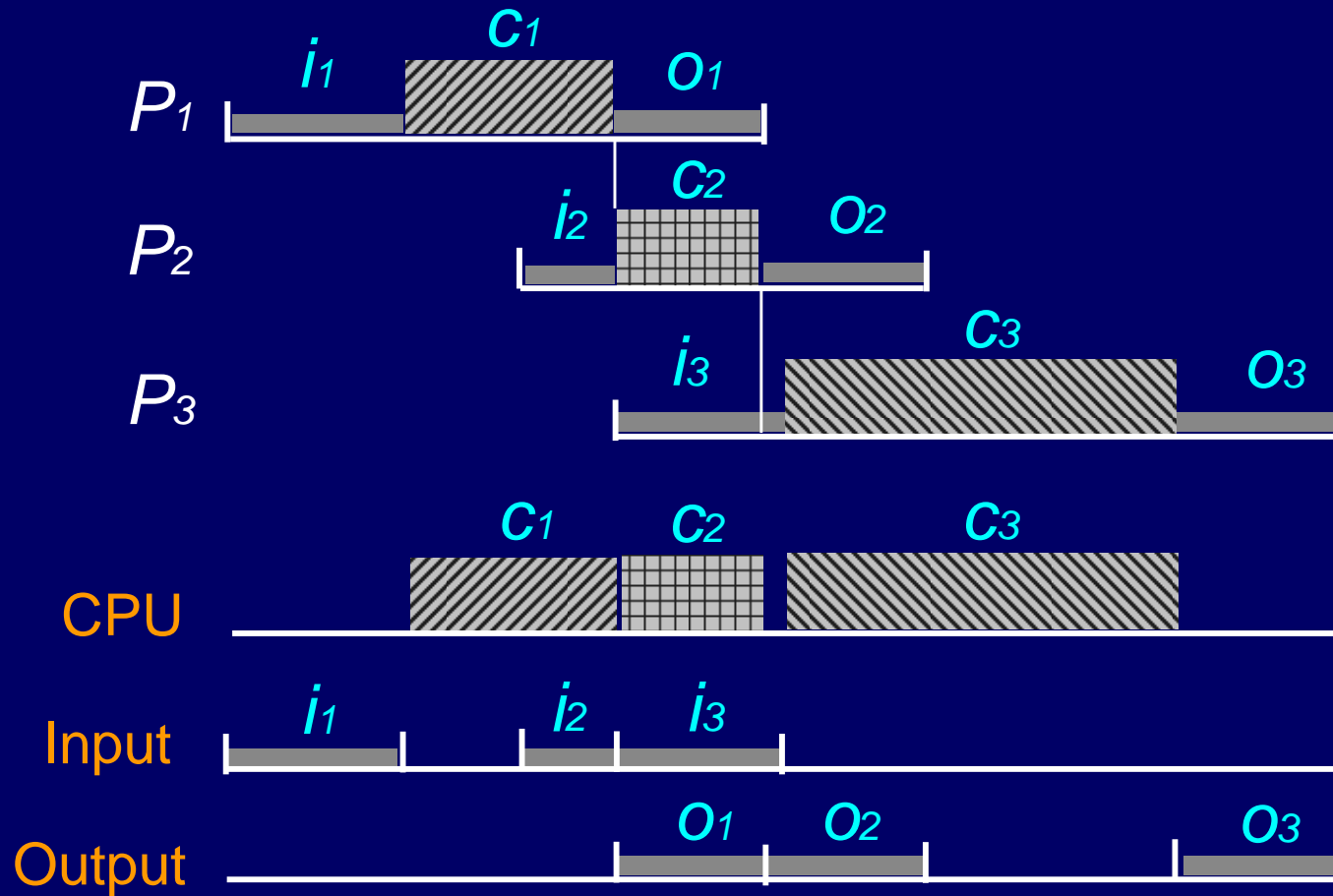


i input
c elaborazione
o output

Multiprogrammazione



Multiprogrammazione



Multitasking

- Consente l'interattività senza lasciare inattiva la CPU
- Si appoggia su:
 - multiprogrammazione (coesistenza di più programmi in memoria)
⇒ **memoria virtuale**
 - condivisione della capacità elaborativa della CPU (che può eseguire una sola istruzione alla volta)
⇒ **time-sharing**

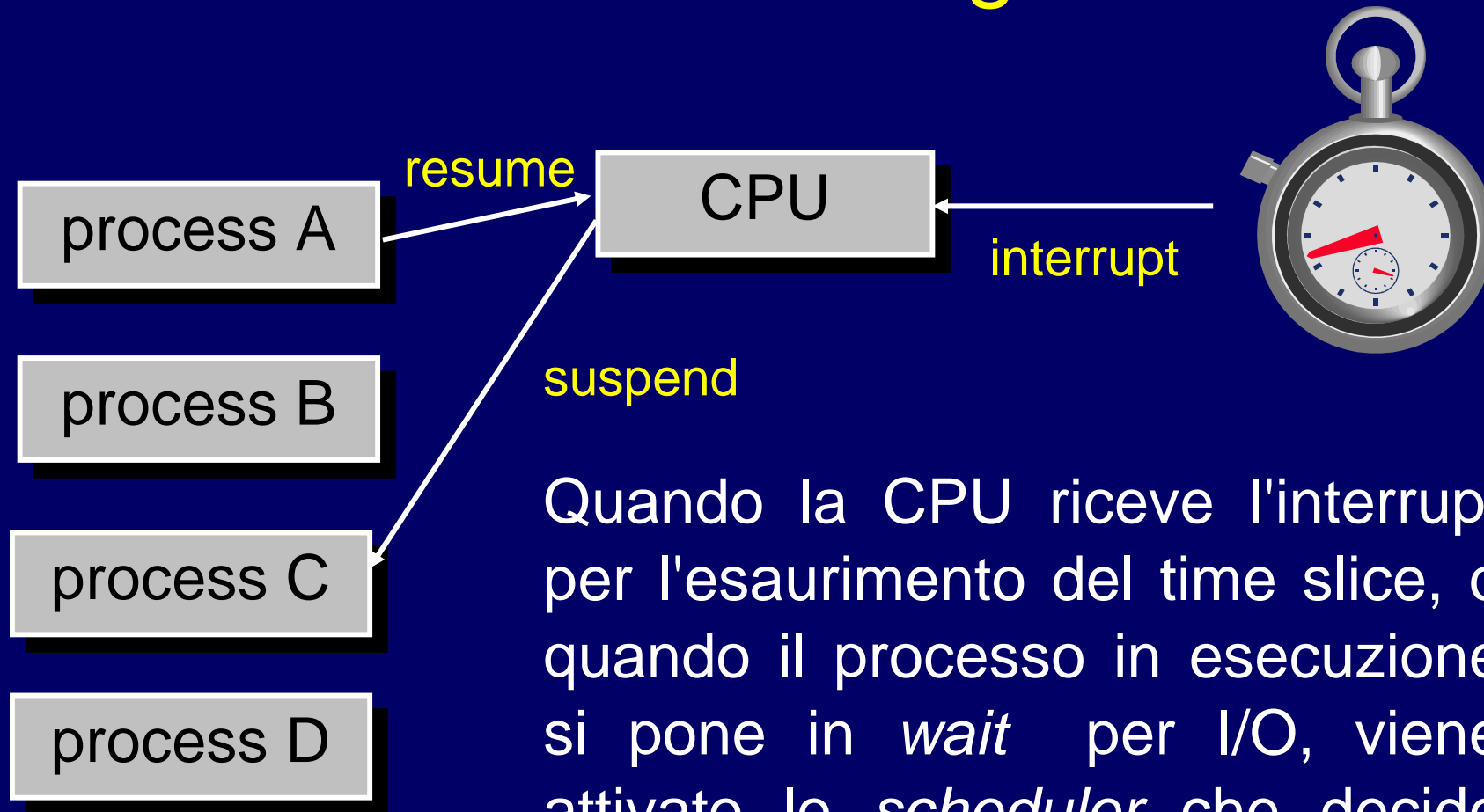
Come funziona il multitasking?

- Ogni processo “vede” una “macchina virtuale”:
 - set di istruzioni disponibili leggermente ridotto (alcune istruzioni sono riservate al sistema operativo)
 - memoria virtuale
 - accesso alle periferiche esclusivamente tramite librerie di sistema
 - protezioni e funzioni di lock nel file system
 - multiplexing della CPU (la velocità di elaborazione dipende quindi dal carico effettivo, cioè da quante macchine virtuali sono attive)

Time-sharing

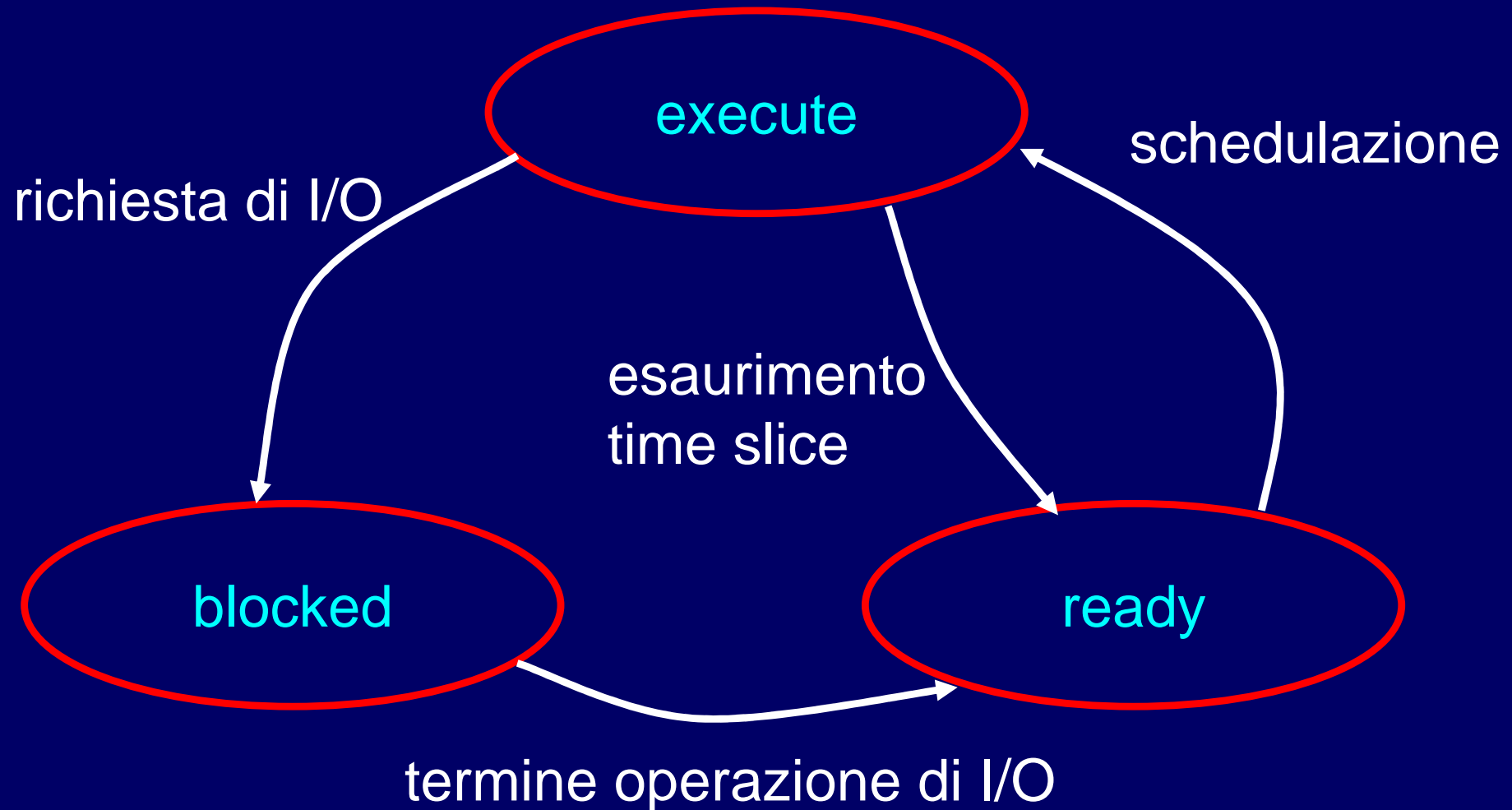
- Sistema multiprogrammato con multiplexing della CPU che consente il multitasking con una singola CPU
- Ogni processo viene eseguito finché:
 - non si esaurisce la sua time slice
 - non esegue un'operazione di I/O
- Quando un processo viene interrotto:
 - il S.O. ne salva il contesto (registri, ed in particolare la Processor Status Word - PSW)
 - il processo successivo in ordine di priorità viene riattivato (resumed)

Time-sharing

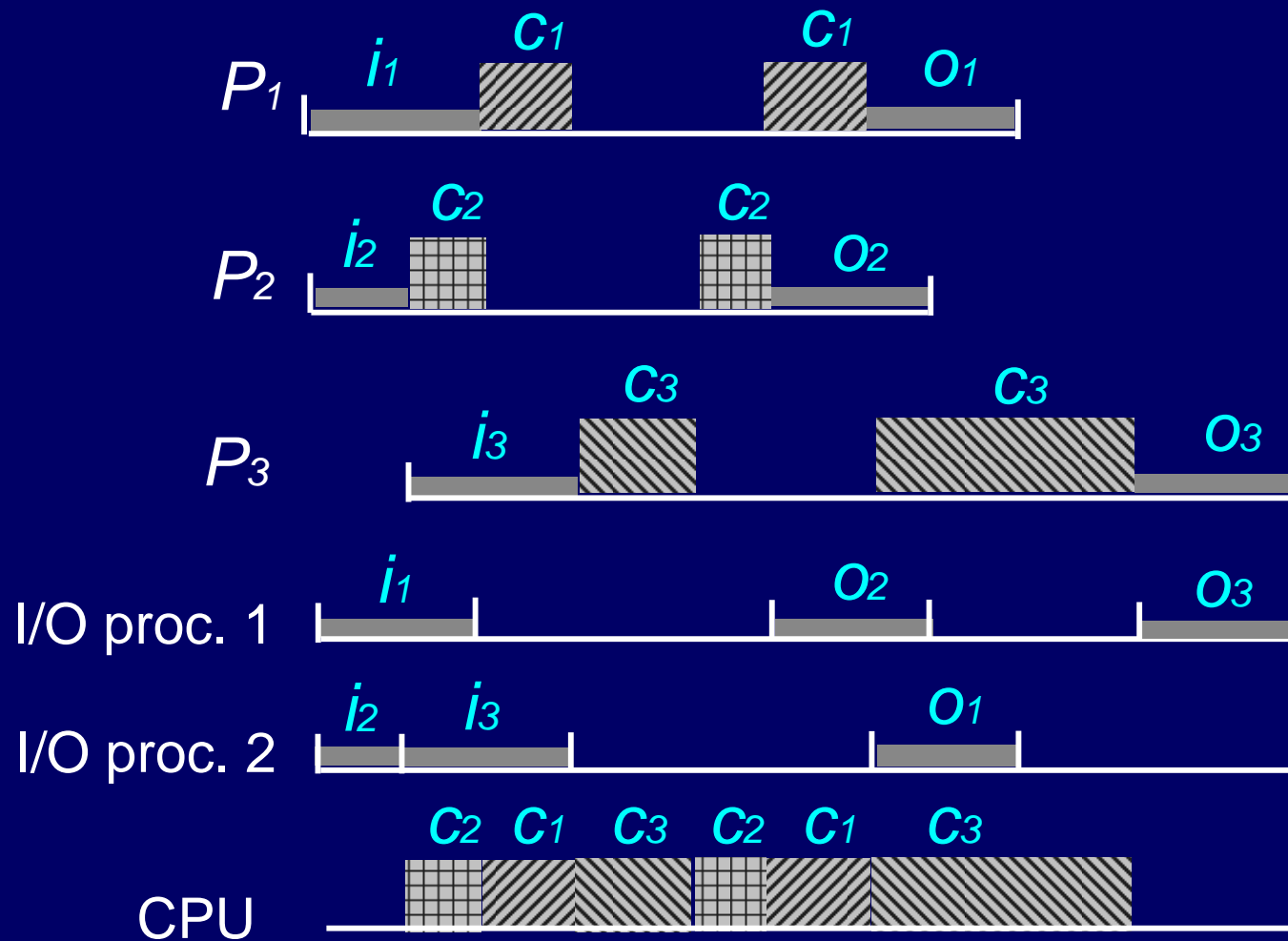


Quando la CPU riceve l'interrupt per l'esaurimento del time slice, o quando il processo in esecuzione si pone in *wait* per I/O, viene attivato lo *scheduler* che decide quale processo deve essere posto in esecuzione

Time-sharing



Time-sharing



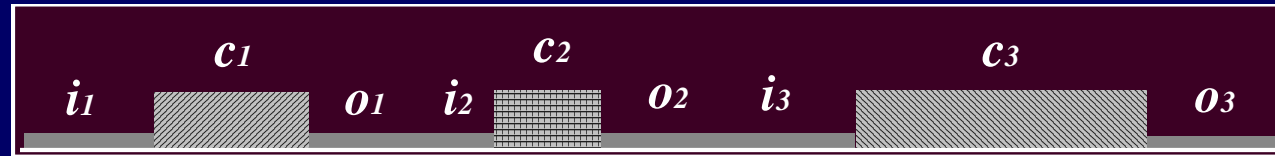
Elapsed time vs. CPU time

- Elapsed time
 - tempo “utente” trascorso tra l’attivazione del processo e il suo completamento
- CPU time
 - tempo totale per il quale il processo ha impegnato la CPU

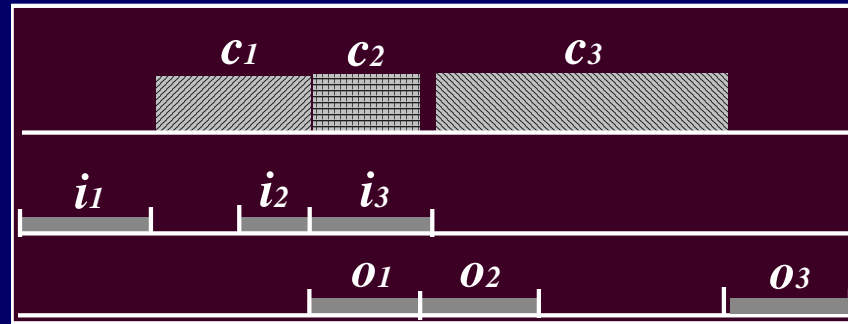
$$\text{CPU time} \leq \text{Elapsed time}$$

Confronto dell'efficienza

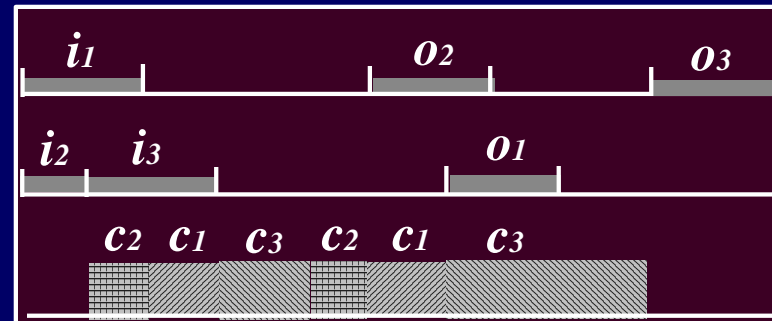
Batch



Multiprog.

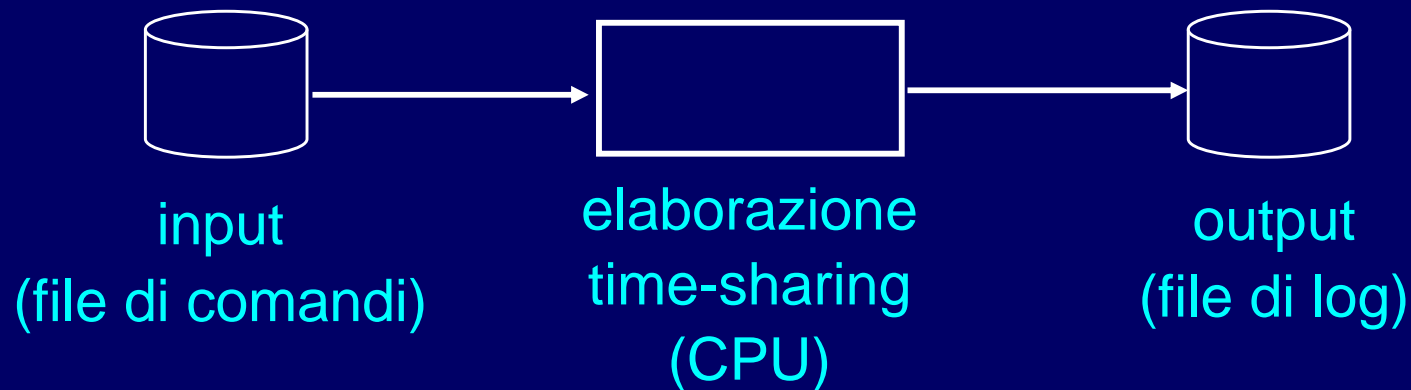


Time-sharing



Batch e time-sharing

- L'elaborazione batch esiste ancora nei sistemi time-sharing:



- Utile per elaborazioni lunghe (anche ore o giorni): non richiede alcun dispositivo di I/O (es. terminale) dedicato

Programmi in background

- Via di mezzo tra batch e interattivo:
 - permettono di continuare a lavorare in quanto non monopolizzano il terminale
 - utilizzano il terminale per l'output, e talvolta anche per l'input
 - sono normalmente dei sottoprocessi: la loro esistenza è legata all'esistenza del processo che li ha generati
 - comunissimi nei sistemi con interfaccia a window: il programma lanciato effettua l'I/O in una finestra indipendente da quella da cui è stato attivato

Real-time

- I dati devono essere elaborati entro tempi vincolanti, determinati dalla velocità di funzionamento degli elementi fisici esterni che interagiscono con il processore
- Applicazioni:
 - controllo di macchinari e servomeccanismi
 - processori ausiliari per sistemi di I/O ad alta velocità, es. dischi, grafica (anche giochi)
- È incompatibile con il time-sharing, in quanto non è ammissibile che un maggior carico del sistema allunghi i tempi di elaborazione dei dati