



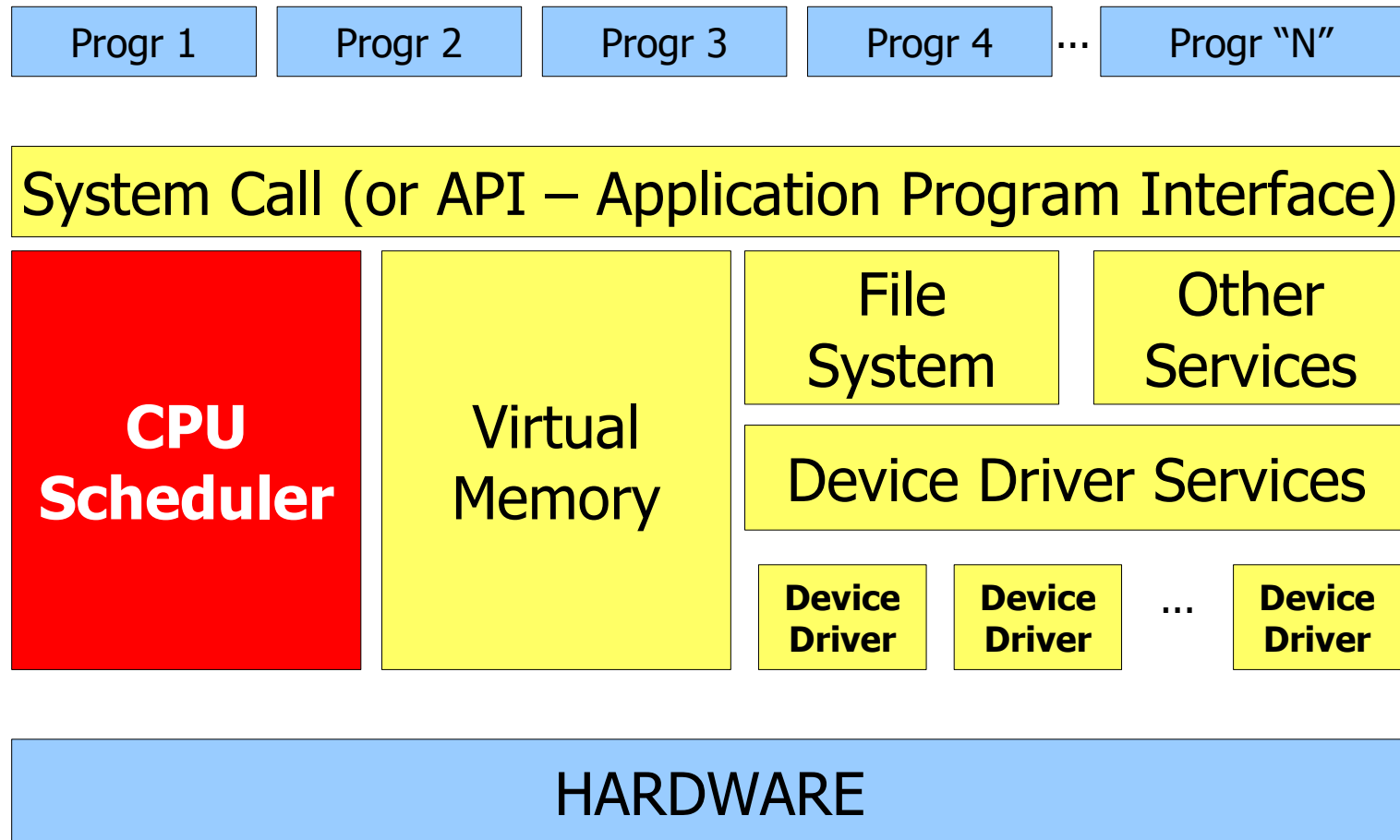
Scheduling della CPU

Introduzione ai Sistemi Operativi
Corso di Abilità Informatiche
Laurea in Fisica

prof. Ing. Corrado Santoro

A.A. 2010-11

Architettura di un sistema operativo



Il Concetto di "Processo"



- Un **processo** è un'istanza di **programma in esecuzione**
- E' caratterizzato da:
 - Il codice (le istruzioni) del programma
 - La memoria usata dal programma
- In un certo istante di esecuzione, un processo è caratterizzato anche da:
 - Lo **stato** della CPU

Processi e scheduling

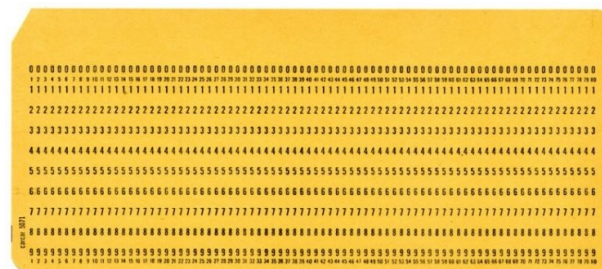


- In un moderno PC è possibile eseguire **diversi processi contemporaneamente**
- Tuttavia la CPU è una sola
- Compito dello **scheduler** è permettere il parallelismo assegnando la CPU ai processi
- Obiettivi dello **scheduler**
 - **Elevato Throughput**
 - **Fairness**

Uno sguardo alla storia: il batch processing



- Negli anni '70, si usava il modello “batch” (a lotti)
- Le applicazioni principali erano quelle di calcolo:
 - Programma di elaborazione
 - Sequenza di dati in input
 - Output dei dati calcolati
- I programmi da eseguire, insieme ai dati in input, erano inseriti in una coda
- Lo scheduler ne prendeva uno per volta e lo eseguiva fino alla fine
- NO PARALLELISMO!

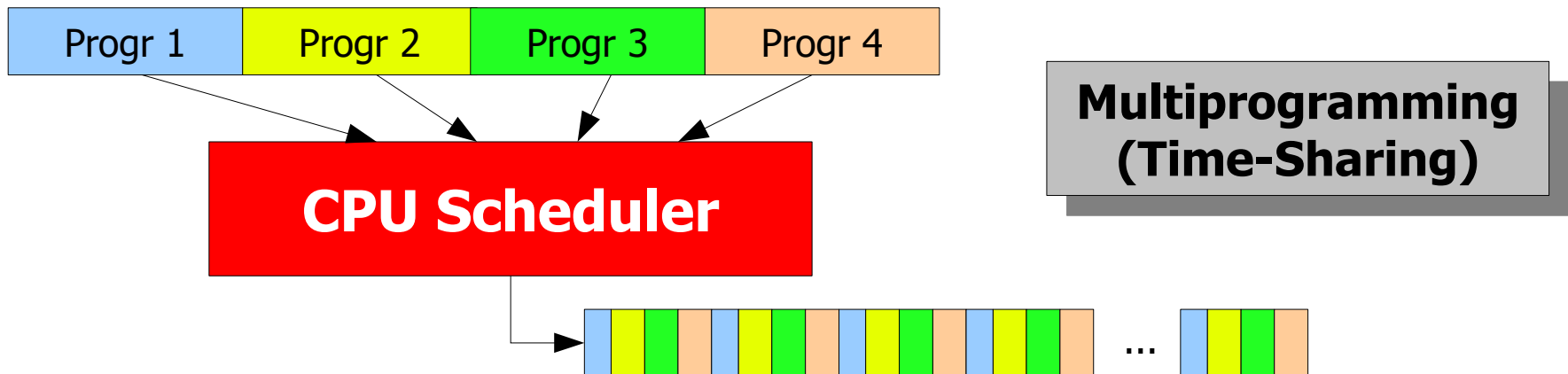
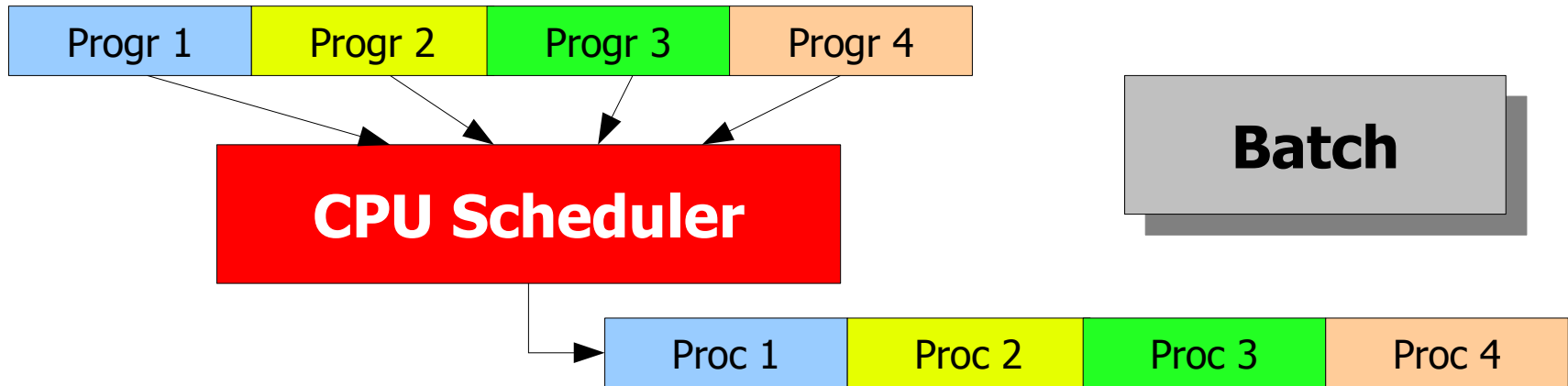


I Sistemi Multiprogrammati



- **Multiprogrammazione:**
 - Possibilità di eseguire più processi contemporaneamente
 - Implicazioni su: CPU Scheduling e gestione della memoria
- **Pseudo-parallelismo (**time-sharing**):**
 - si basa sull'eseguire i processi “un pezzetto per volta”
 - si assegna la CPU ad un processo per volta, ma per un tempo limitato

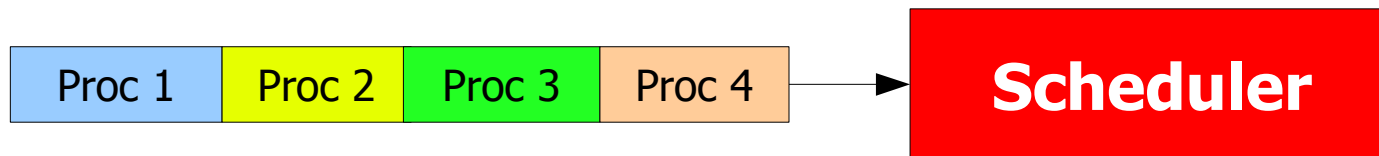
Batch vs. Multiprogramming



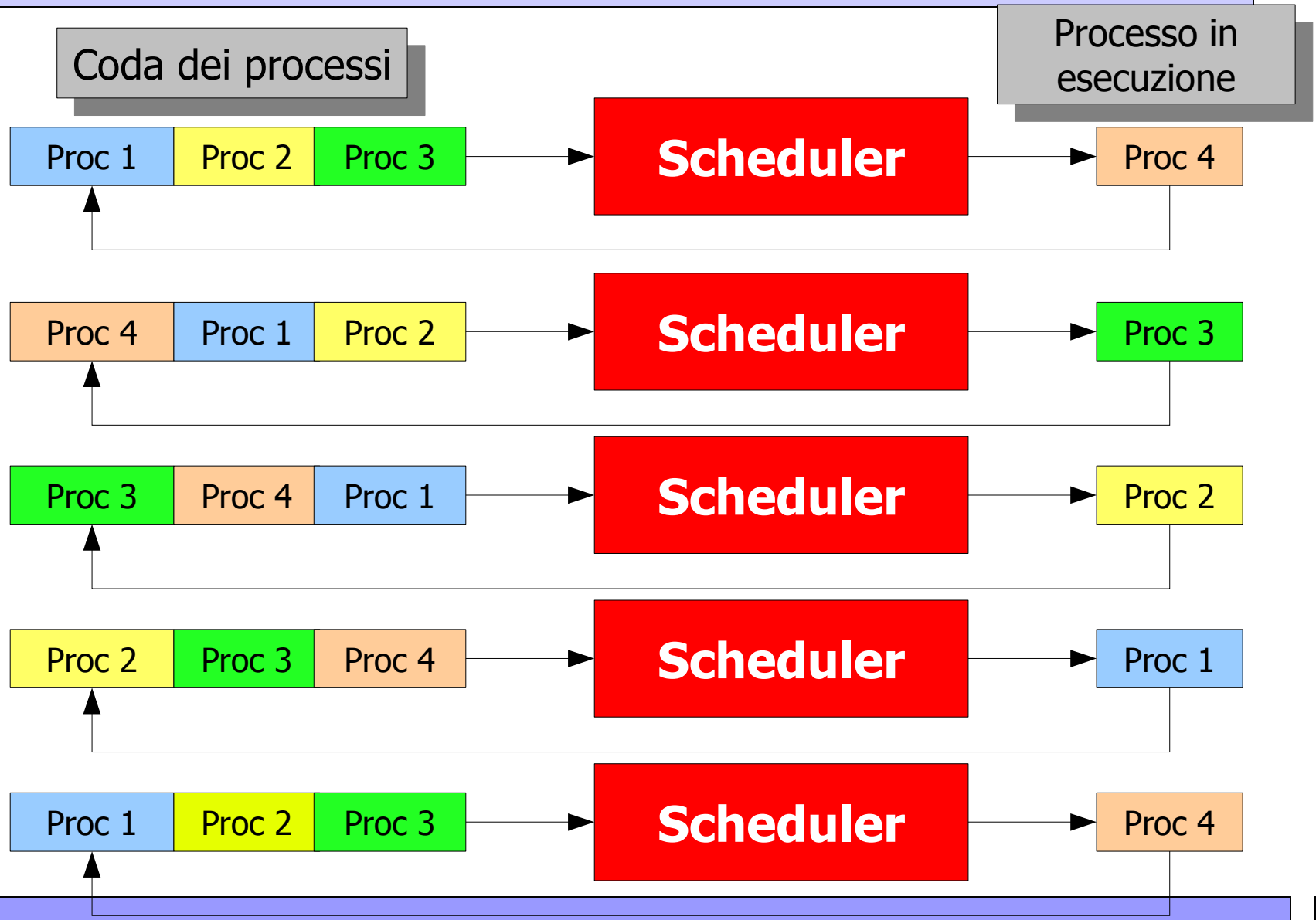
Round-robin scheduling



- I processi sono inseriti in una **coda**
- Un timer hardware genera un segnale che causa l'attivazione dello **scheduler**
- Lo scheduler
 - **sospende** il processo corrente e lo inserisce alla fine della coda
 - estrae il processo successivo dalla testa della coda e lo manda in **esecuzione** fino al prossimo segnale del timer



Round-robin scheduling

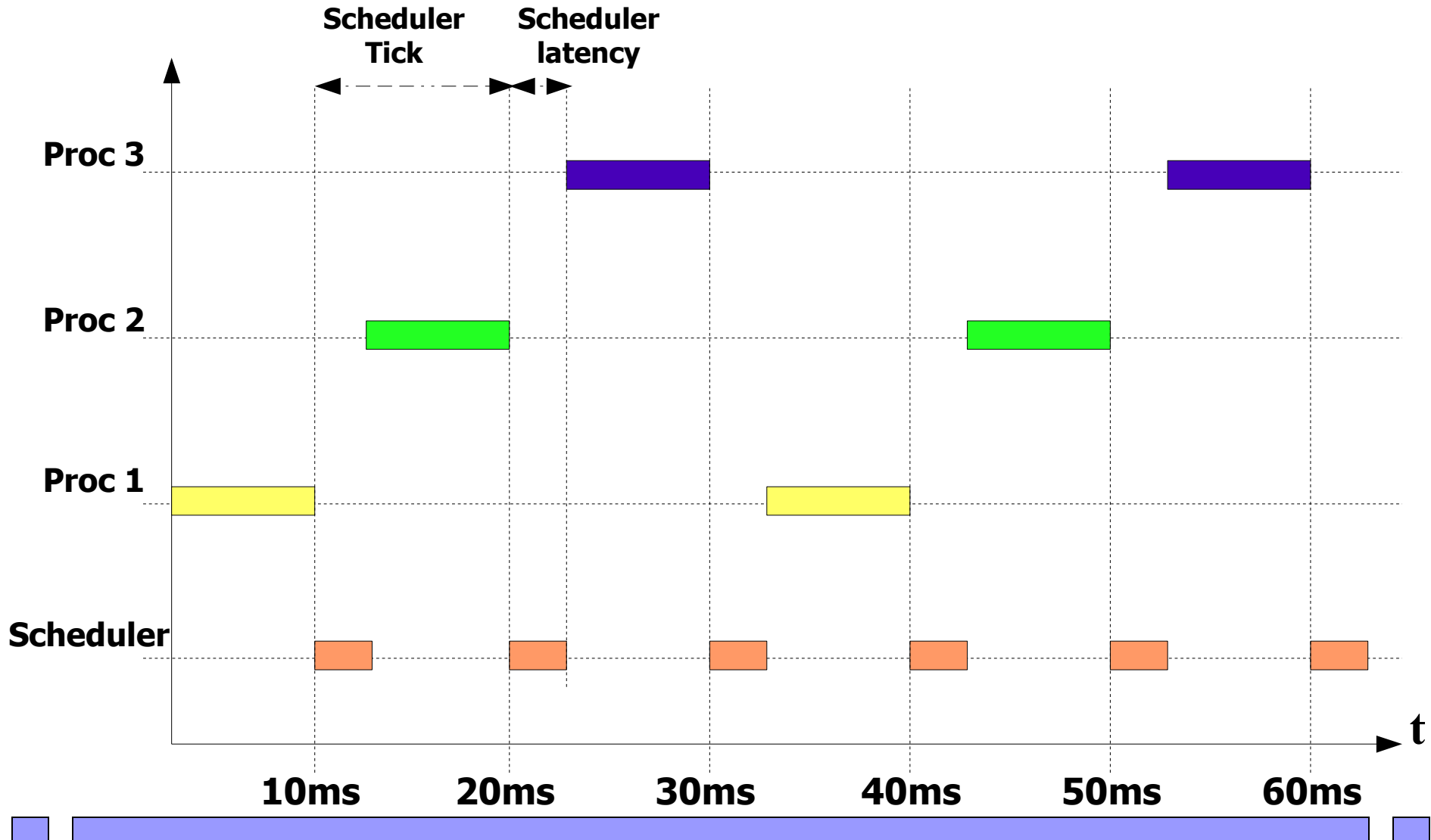


Un paio di termini...

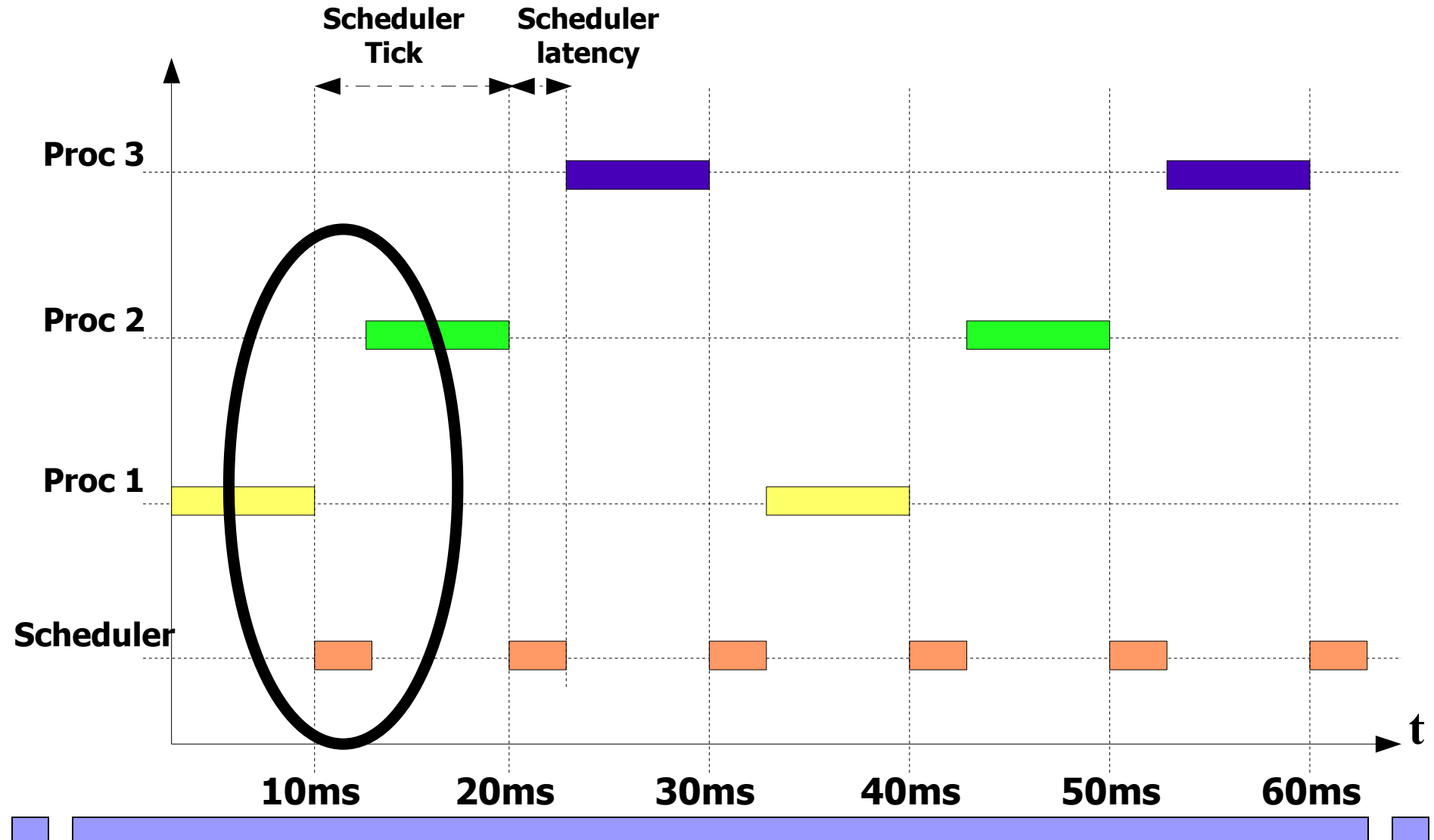


- **Scheduler Tick**
 - Intervallo di tempo di attivazione dello scheduler
 - In genere è 10ms
- **Context Switch**
 - Procedura di sostituzione del **processo corrente** con il **nuovo processo** estratto dalla coda
- **Scheduler Latency**
 - Durata della procedura di context switch
 - Deve essere molto minore dello scheduler tick

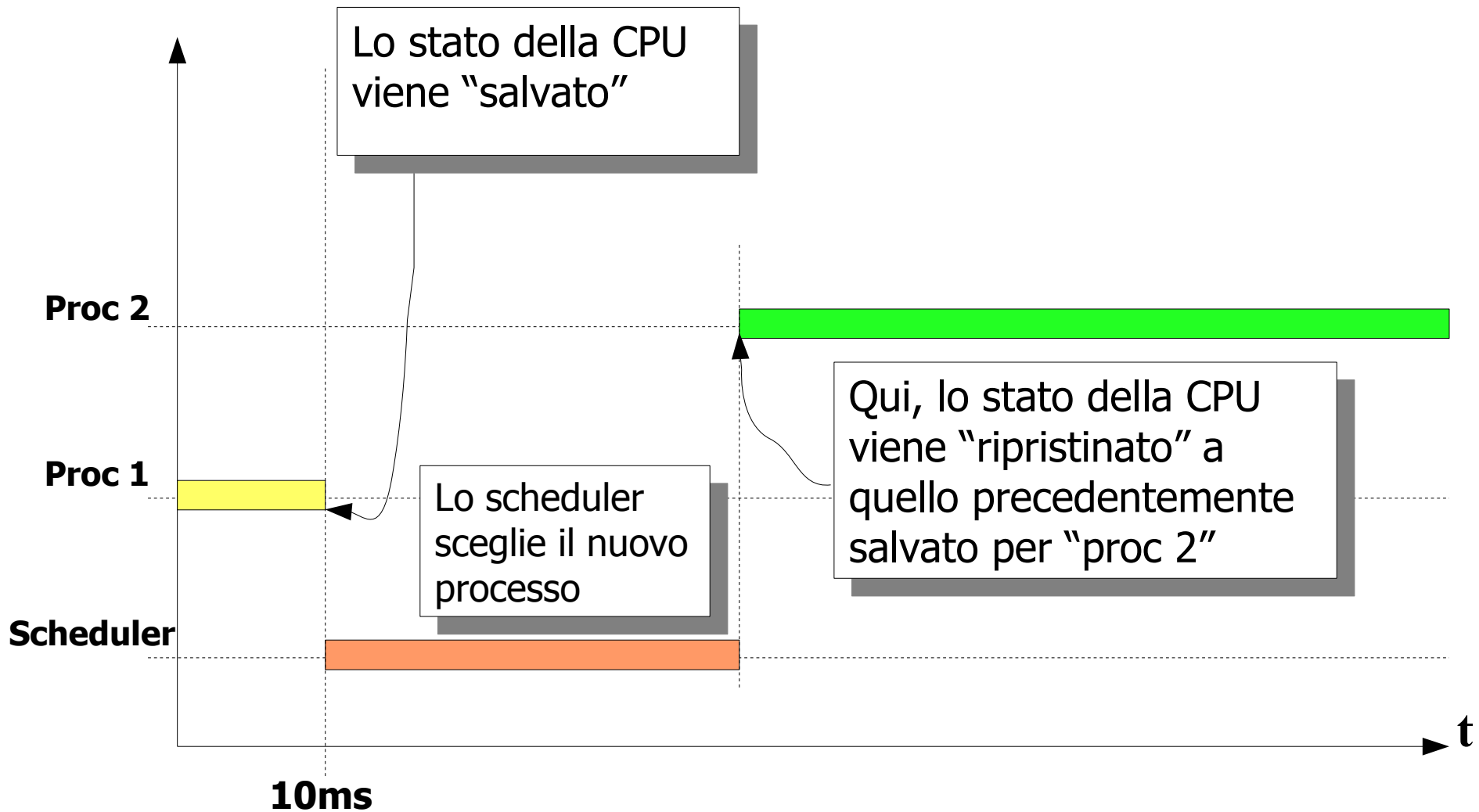
Sequenza temporale dello scheduling



Context Switch: come avviene?



Context Switch: come avviene?



Stati di "wait"



- Un processo, consuma sempre tempo di CPU?
- Se un processo sta **attendendo qualcosa** (es):
 - Dati che arrivano dalla rete
 - Il completamento di un'operazione di lettura o scrittura di un blocco dati
 - La pressione di un tasto da parte dell'utente
 - ...
- ... ha senso assegnare comunque la CPU?
- Sarebbe inutile! Il processo **non potrebbe proseguire perché è in attesa di qualcosa**

Stato di "wait"



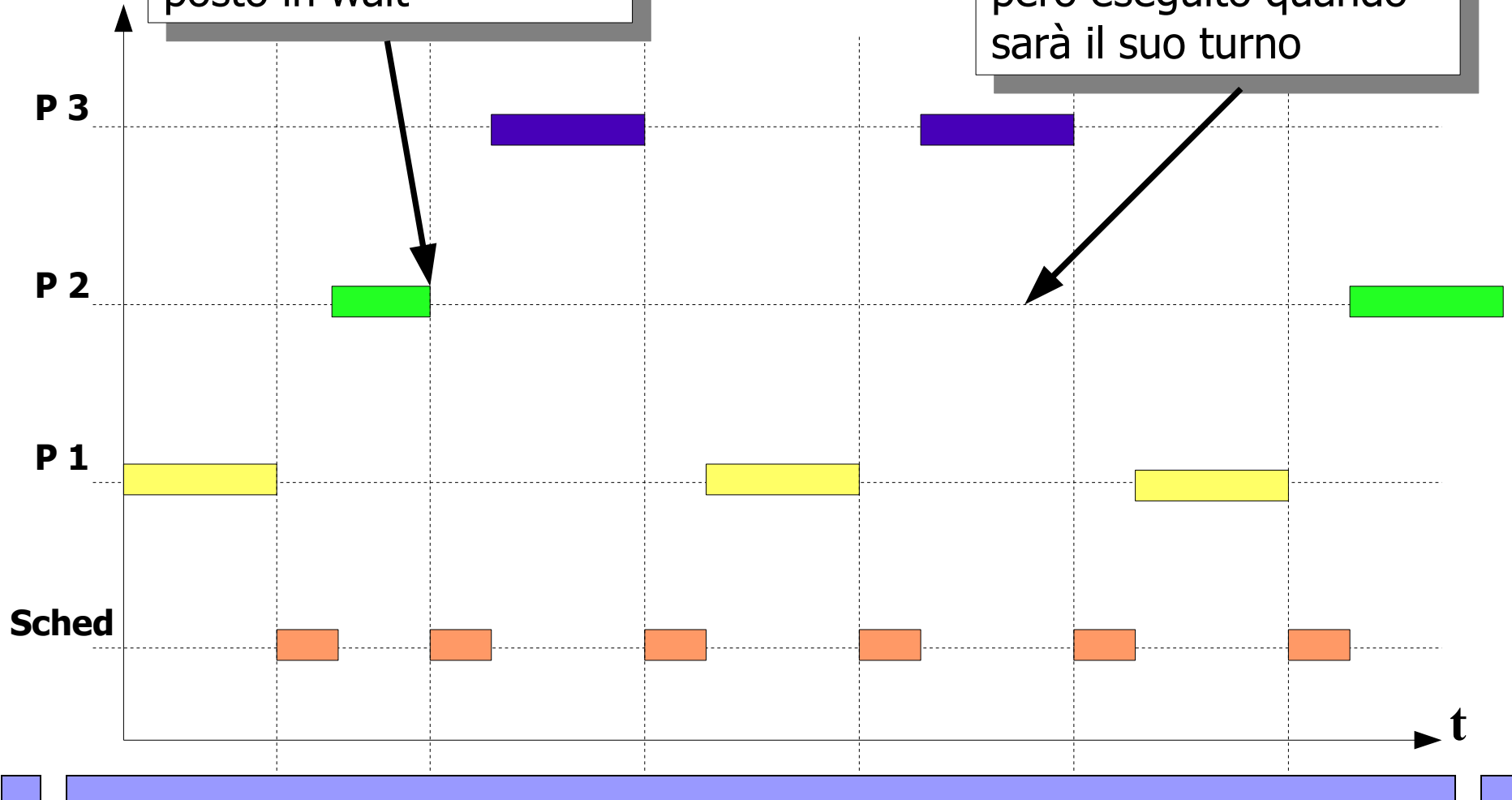
- Quando un processo necessita di un evento per poter proseguire (attende qualcosa) ...
- ... questa situazione è **segnalata** allo scheduler, il quale marca il processo in **stato di wait**
- **I processi in stato di wait non vengono attivati dallo scheduler**
- Quando arriva l'evento che sblocca l'attesa, il processo viene marcato come **pronto** (ready-to-run) e può essere attivato dallo scheduler

Scheduling e stati



P2 deve attendere qualcosa, viene dunque posto in wait

Accade l'evento che sblocca P2; P2 sarà però eseguito quando sarà il suo turno



Stati di un processo



- Possiamo allora definire uno **stato di esecuzione** di un processo che può assumere i valori
 - **RUNNING**: il processo è in esecuzione e sta usando la CPU
 - **READY-TO-RUN**: il processo non sta usando la CPU ma è pronto per l'esecuzione
 - **WAITING**: il processo sta attendendo un evento esterno

Diagramma a stati di un processo

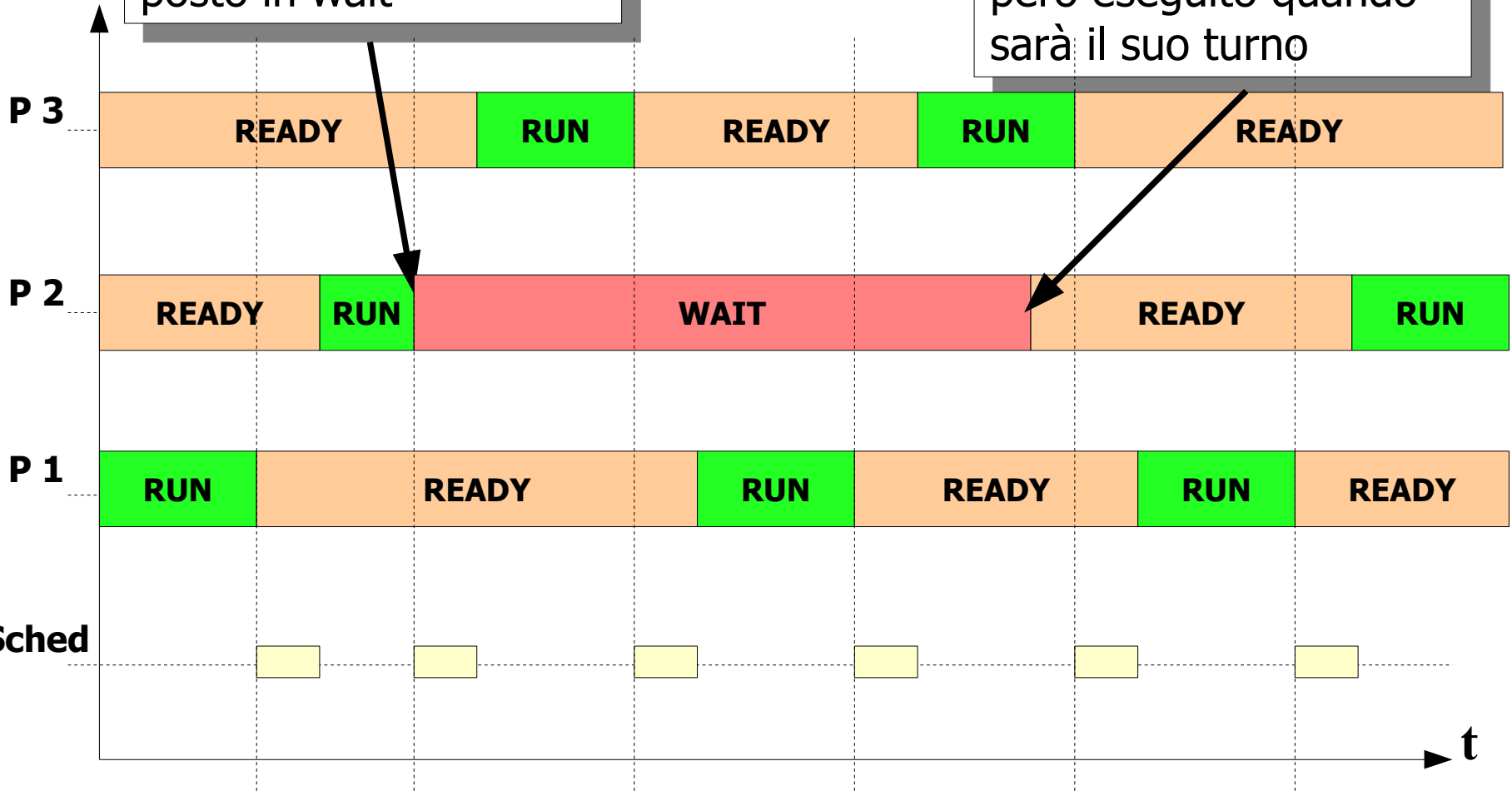


Scheduling e stati



P2 deve attendere qualcosa, viene dunque posto in wait

Accade l'evento che sblocca P2; P2 sarà però eseguito quando sarà il suo turno



Conclusioni



- Lo scheduler permette l'esecuzione di diversi processi in uno stesso computer: assegna la CPU in modo dinamico ai vari processi
- **Batch Scheduling:** un processo per volta fino alla fine dell'esecuzione
- **Time-sharing Scheduling:** un pezzetto di processo per volta (pseudo-parallelismo)
- **Round-robin:** organizzazione dei processi in una coda circolare
- **Context-switch:** procedura che salva il contesto del processo in esecuzione e ripristina il contesto (precedentemente salvato) del processo che dovrà essere eseguito