

Università di Roma Tor Vergata
Corso di Laurea triennale in Informatica
Sistemi operativi e reti
A.A. 2016-17

Pietro Frasca

Lezione 5

Martedì 25-10-2016

Definizione di processo

- Esiste una distinzione concettuale tra programma e processo. Un programma è l'implementazione di un algoritmo, il processo è un'entità astratta che rappresenta l'attività del calcolatore e del SO relativa all'esecuzione di un programma.
- Il programma è un'entità statica mentre il **processo è un'entità dinamica**.
- Quando un programma sorgente è compilato, genera il programma eseguibile (entità statica). Se il programma, ad esempio un gioco, è eseguito più volte, con diversi valori di input, è evidente che l'esecuzione sarà ogni volta diversa (entità dinamica).
- Un processo è composto da un **descrittore**, dal codice del programma che esegue, da un'area di memoria contenente i dati inizializzati e le variabili globali da un area per

l'allocazione dinamica di dati (heap) e dallo stack, un'area di memoria di lavoro necessaria per salvare i record di attivazione delle funzioni e le variabili locali. L'insieme di queste componenti è detta ***immagine*** del processo.

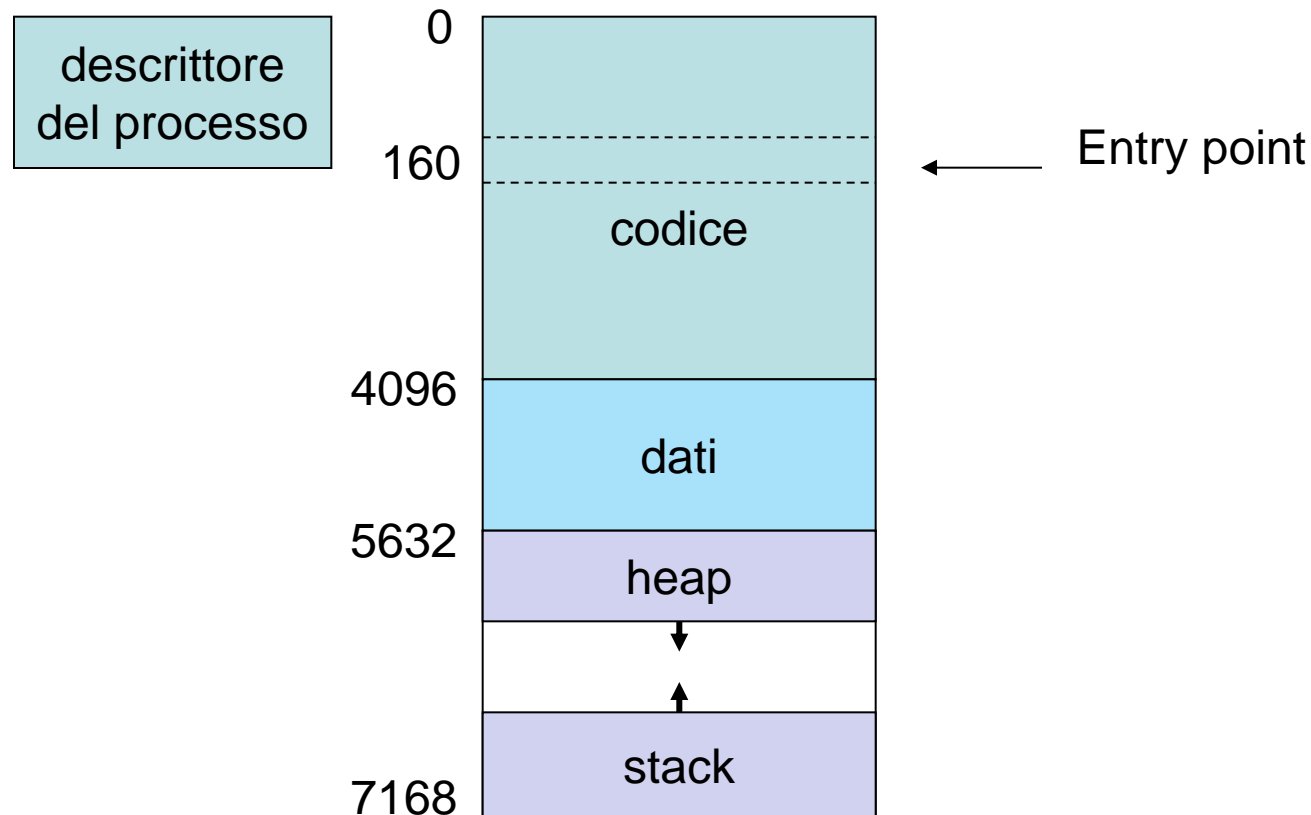


Immagine di un processo

Stati di un processo

- Un processo durante la sua esecuzione, esegue transizioni di stato che sono causate sia dall'esecuzione delle istruzioni del processo stesso, come ad esempio quando esso esegue operazioni di I/O, sia da eventi esterni asincroni con la sua esecuzione, come ad esempio la ricezione di segnali.
- Un processo entra nello **stato nuovo** quando è creato, ad esempio quando si avvia un programma.
- Un processo passa dallo stato nuovo allo **stato di pronto** dopo che il SO ha verificato che esso può essere effettivamente eseguito (ad esempio il processo ha i diritti di esecuzione).
- Un processo passa dallo stato di **pronto** allo stato di **esecuzione** quando ad esso è assegnato il processore.
- Un processo è nello **stato attivo** quando si trova nello stato di **pronto** o di **esecuzione**.
- Un processo passa dallo **stato di esecuzione** allo **stato di bloccato** quando è in attesa di qualche evento.

- Il processo ritorna nello stato di pronto quando l'evento atteso si è verificato.
- La transizione dallo stato di esecuzione allo stato di pronto è chiamato **prerilascio** (preemption) o **revoca**.
- Il prerilascio può avvenire per vari motivi, come ad esempio, quando un processo ha esaurito il suo quanto di tempo (nei sistemi time-sharing), o è presente nella coda di pronto un altro processo con priorità più alta.
- Il passaggio dallo stato di pronto allo stato di esecuzione è gestito dallo scheduler, un componente del kernel che seleziona un processo cui assegnare la CPU, tra tutti i processi che si trovano nello stato di pronto. Lo **scheduler** ha il compito di garantire che tutti i processi pronti possano avanzare nella loro esecuzione.
- Il processo passa nello stato **terminato** quando ha terminato l'esecuzione del suo programma o quando si è verificato un'eccezione di vario tipo.

Transizioni di stato

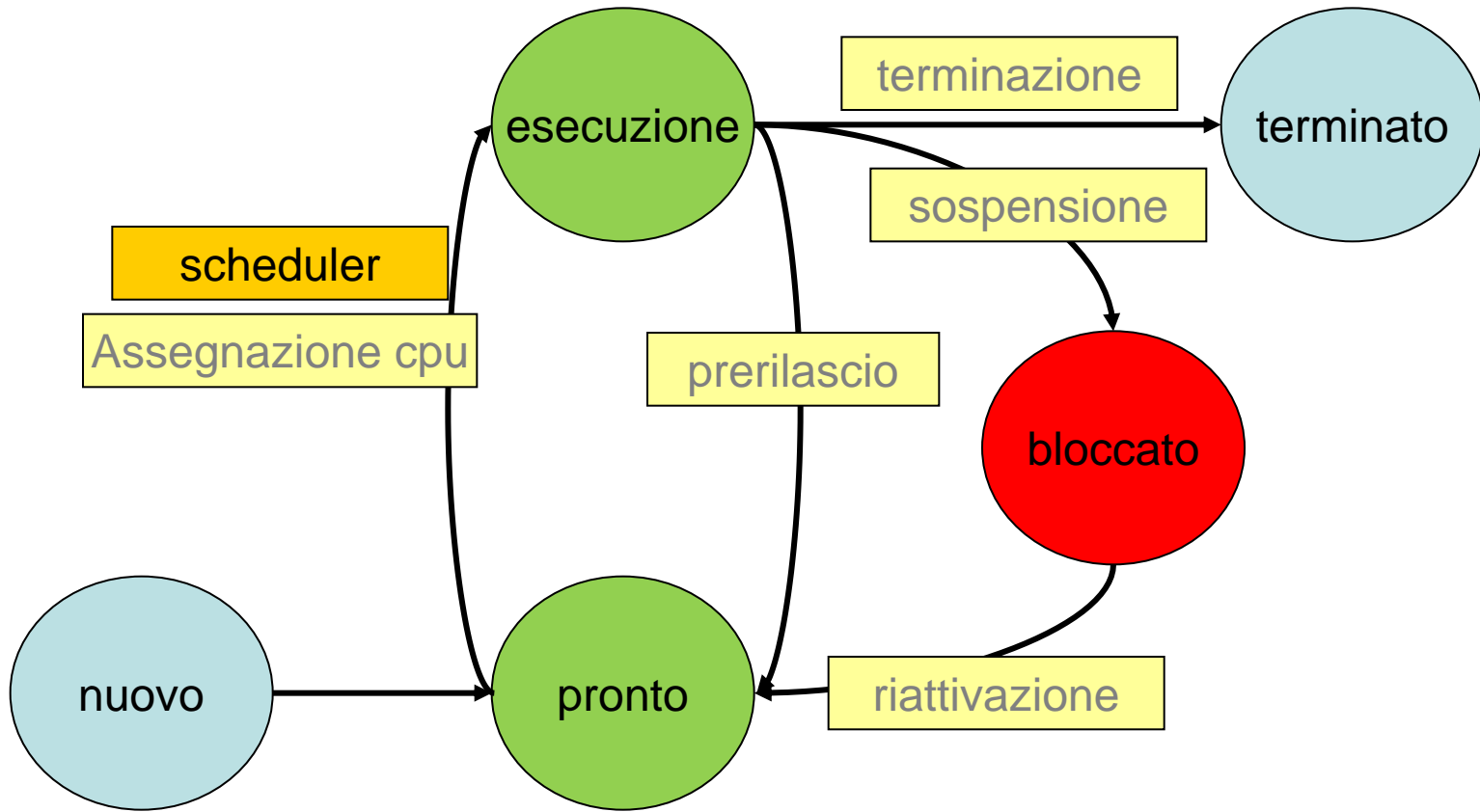


Diagramma di transizione a 5 stati

- In molti sistemi è previsto che un processo sia trasferito dalla memoria principale alla memoria secondaria (disco) in modo da ottenere spazio per altri processi. Tale operazione prende il nome di ***swapping (scambio)*** e lo stato relativo è detto ***swapped***.
- Vedremo un diagramma di transizione di stato con lo stato swapped quando parleremo della gestione della memoria.

Descrittore del processo

- Ogni processo è rappresentato da una struttura dati, detta **descrittore del processo (PCB Process Control Block)**.
- I descrittori dei processi sono memorizzati in una tabella (lista concatenata), detta **tabella dei processi**.
- I campi presenti nel descrittore di processo dipendono dal particolare sistema operativo e dall'architettura dell'hardware.
- Generalmente il descrittore del processo contiene le seguenti informazioni:
 - **Identificatore del processo**. Spesso si identifica il processo con un numero intero detto **PID**.
 - **Stato del processo**. Questo campo identifica lo stato in cui si trova il processo in un determinato istante. Può anche non essere presente nel descrittore ma può essere ottenuto implicitamente dall'appartenenza del descrittore ad una delle code gestite dal kernel.

- **Informazioni sullo scheduling di CPU.** La scelta del processo pronto a cui assegnare la CPU può essere effettuata secondo diversi criteri.
 - **FIFO.** Il criterio più semplice è il FIFO (*First-In-First-Out*), che prevede di assegnare la CPU al processo pronto in attesa da più tempo.
 - **Priorità.** Ad ogni processo è assegnata una priorità, fissa o dinamica, che indica la sua importanza relativa nei confronti degli altri processi.
 - **Deadline (scadenza).** La scelta del prossimo processo può essere basata anche in termini di intervallo di tempo (**deadline**) in cui l'esecuzione del processo deve essere portata a termine. Nel descrittore del processo, in questo caso, è contenuto un valore che, sommato all'istante della richiesta di servizio da parte del processo, determina il tempo massimo entro il quale la richiesta deve essere soddisfatta (*sistemi in tempo reale*).

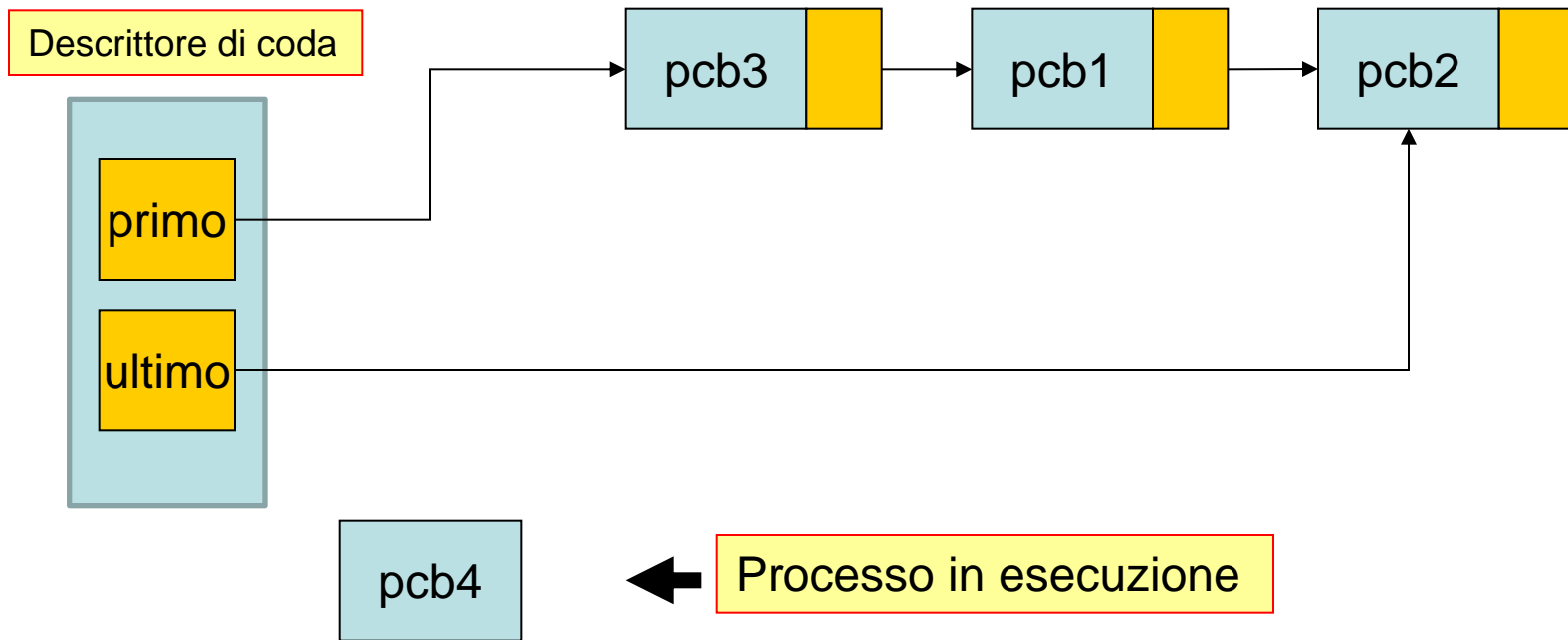
- **Quanto di tempo.** Nei sistemi time-sharing, nel descrittore è contenuto un valore che rappresenta l'intervallo di tempo (quanto) consecutivo in cui la CPU è assegnata allo stesso processo.
- **Informazioni sulla gestione della memoria.** Contiene le informazioni relative all'area di memoria principale nella quale sono caricati il codice, i dati e lo stack del processo. Queste informazioni dipendono dalla particolare tecnica di gestione della memoria usata dal sistema operativo.
- **Contesto del processo.** L'insieme dei valori dei registri del processore al momento della sospensione dell'esecuzione di un processo è salvato nel suo descrittore. Questo insieme di valori prende il nome di **contesto del processo** e dipende dall'architettura del processore. Tipici registri presenti in una CPU sono il *PSW (Program Status word)*, lo *SP (puntatore allo stack - stack pointer)*, registri indice, accumulatori, registri di uso generale. Il contesto è recuperato dal descrittore e riportato ai registri quando il processo torna in esecuzione.

- **Utilizzo delle risorse.** Queste informazioni comprendono la lista dei dispositivi di I/O allocati al processo, i file aperti, il tempo di uso della CPU ecc.
 - **Identificatore del processo successivo.** Come si è detto, a seconda del loro stato (pronto o bloccato), i processi sono inseriti, in apposite code. Ogni descrittore contiene pertanto un puntatore al processo successivo (e spesso anche un puntatore al precedente) nella stessa coda.
 - **Informazioni sulla sicurezza**
 - **Informazioni sulle variabili di ambiente**
 - **Informazione utente**
- Per la loro fondamentale importanza i descrittori sono memorizzati in un'area di memoria accessibile solo dal kernel.

Code di processi

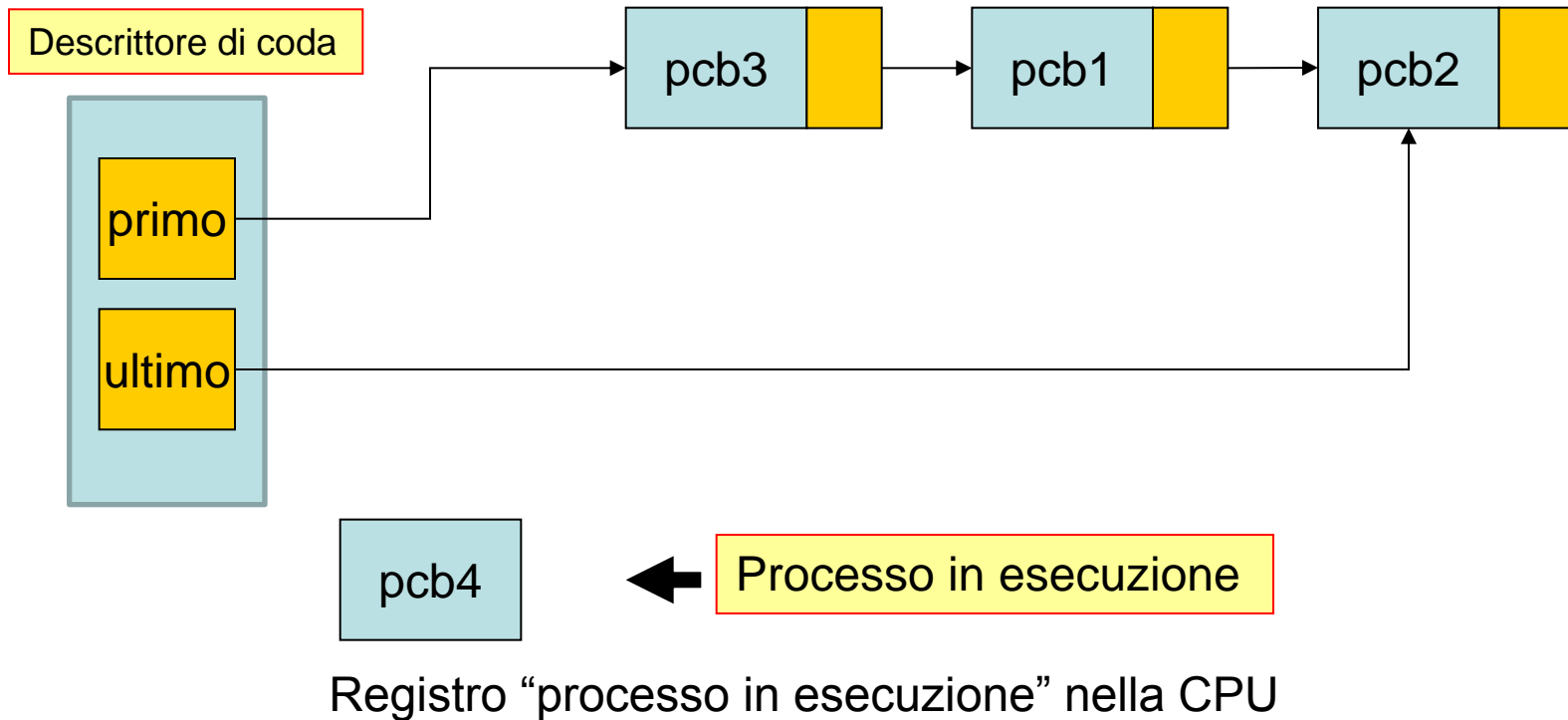
- Il diagramma di transizione degli stati, fornisce una visione astratta del funzionamento dei processi quando questi compiono transizioni di stato. A livello d'implementazione, ciascuno stato è realizzato con una o più code.
- Assumendo che il computer abbia una sola CPU, in ogni istante un solo processo può trovarsi nello stato di esecuzione. Gli altri processi possono essere o nello stato di pronto o nello stato di bloccato in attesa che si verifichi un determinato evento che li riporti nello stato di pronto.
- Generalmente, le CPU possiedono un registro detto **registro del processo in esecuzione** nel quale il sistema operativo memorizza il puntatore al descrittore del processo che è in esecuzione.
- I processi presenti nella memoria principale che sono pronti per essere eseguiti sono organizzati in una o più code dette **code dei processi pronti**.

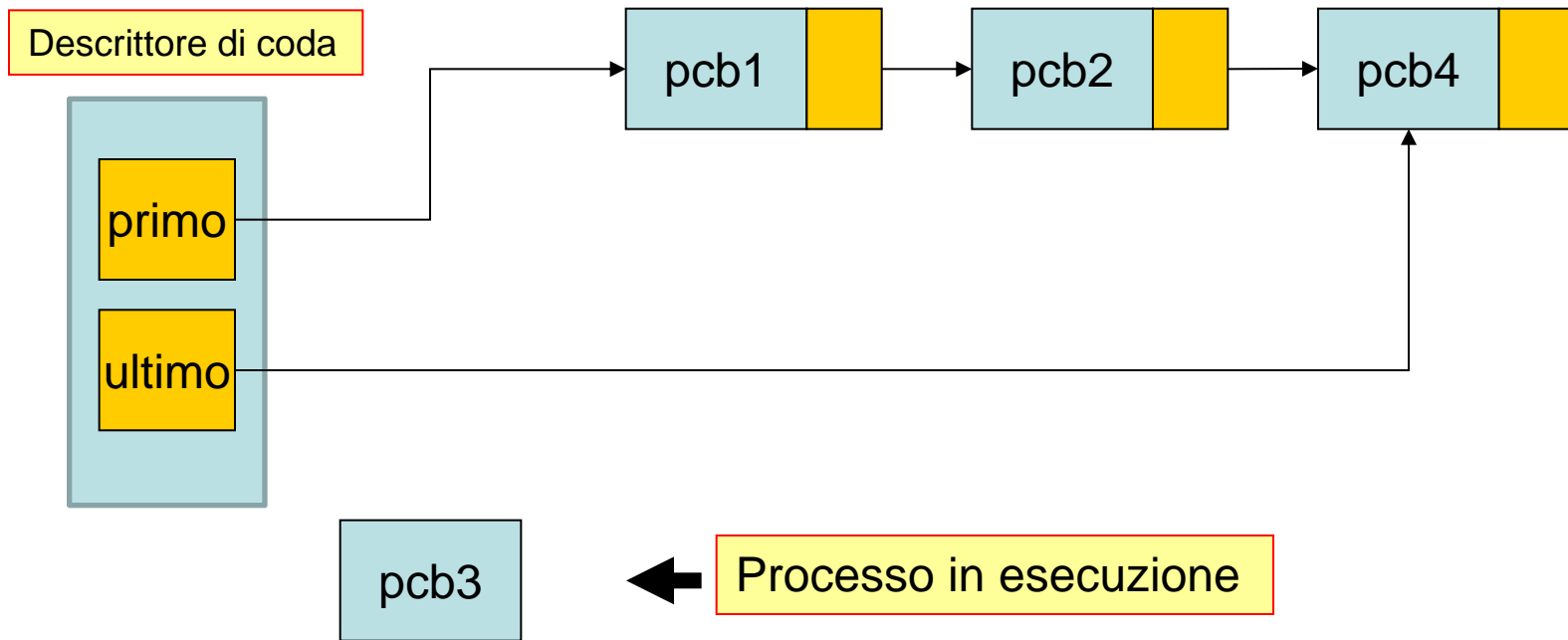
- Ad ogni coda è associato un **descrittore di coda** che contiene due campi che specificano rispettivamente l'indice del primo e dell'ultimo descrittore di processo in coda.
- Ogni descrittore di processo ha un campo per indicare il processo successivo contenuto nella coda dei processi pronti.



Registro "processo in esecuzione" nella CPU

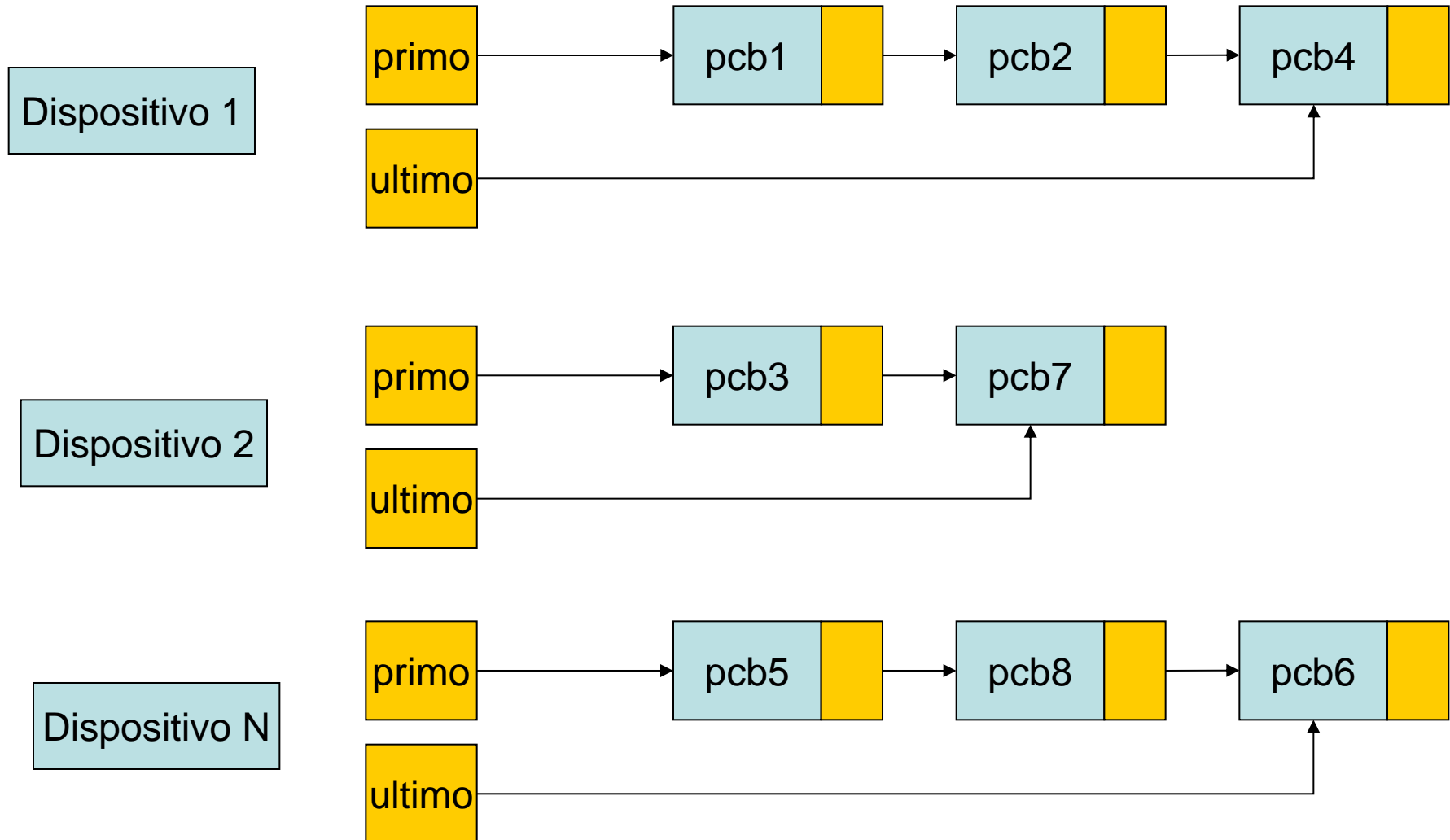
- L'operazione di **inserimento** di un processo in una coda dei processi pronti è dovuta alla transizione nello stato di pronto di un processo dallo stato nuovo, bloccato o in esecuzione.
- L'operazione di prelievo corrisponde al passaggio dallo stato di pronto allo stato di esecuzione.





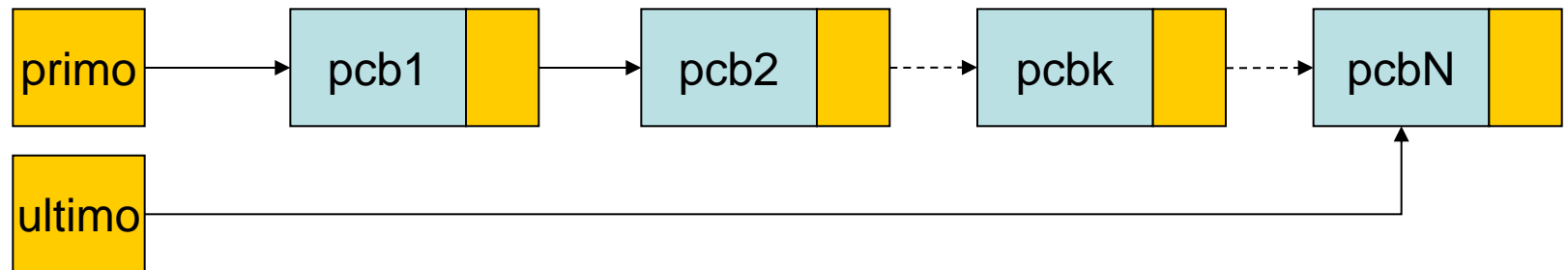
Registro “processo in esecuzione” nella CPU

- Nella coda dei processi pronti è sempre presente un particolare processo: il **processo di inattività (idle)**, che va in esecuzione quando tutti i processi sono bloccati.
- Il processo *idle* resta sempre nella coda di pronto e ha sempre la priorità più bassa. Rimane in esecuzione fino a quando qualche altro processo entra nella coda di pronto.
- E' compito di un componente del sistema operativo, lo **scheduler a breve termine (short term scheduler)**, selezionare, dalla coda di pronto, il prossimo processo che andrà in esecuzione, in base a qualche strategia (politica).
- Esistono inoltre **code per i processi bloccati**, una per ogni evento o condizione per cui i processi possono attendere. Ad esempio, esistono code per ogni dispositivo di I/O nelle quali sono inseriti i descrittori dei processi in attesa della conclusione di una richiesta di I/O su quel determinato dispositivo.
- L'operazione di creazione di un processo implica la creazione di un nuovo descrittore e l'operazione di terminazione l'eliminazione del descrittore.



Code per i processi bloccati

- In alcuni sistemi può essere presente una coda nella quale sono presenti un numero fisso di **descrittori disponibili** per la creazione di nuovi processi. In tal caso il sistema avrà un numero massimo di processi predefinito. L'operazione di creazione estrae da questa coda il descrittore del processo da creare che sarà successivamente inserito nella coda dei processi pronti. Viceversa l'operazione di eliminazione di un processo riporta nella coda dei descrittori disponibili il descrittore del processo eliminato.



Coda di descrittori disponibili

Cambio di contesto

- Il cambio del processo che è in esecuzione con un altro scelto dalla coda dei processi pronti si realizza, mediante un insieme di operazioni, appartenenti al kernel, che prendono il nome di ***cambio di contesto***. Tali operazioni sono:
 1. Salvataggio del contesto del processo in esecuzione nel suo descrittore (***salvataggio dello stato***). il contesto dipende dall'architettura del processore.
 2. Inserimento del descrittore nella coda dei processi pronti o dei processi bloccati. L'operazione di inserimento del descrittore nella coda dei processi pronti si ha se il cambio di contesto è dovuto ad una revoca mentre l'inserimento in una coda di bloccato si ha se il cambio di contesto è dovuto alla sospensione del processo.
 3. Selezione di un altro processo dalla coda dei processi pronti e caricamento del descrittore di tale processo nel registro processo in esecuzione.

Questa operazione è eseguita dallo scheduler a breve termine, secondo una determinata strategia la quale può essere basata su un algoritmo di tipo FIFO nel caso esista una sola coda per tutti i processi pronti o utilizzando l'informazione relativa alla priorità dei processi nel caso di più code di processi (una per livello di priorità).

4. Caricamento del contesto del nuovo processo nei registri del processore (***ripristino dello stato***).
 5. Aggiornamento di tutte le strutture dati che rappresentano le risorse utilizzate dai processi: memoria, dispositivi di I/O, file aperti ecc.
- Le operazioni di *salvataggio dello stato* e *ripristino dello stato*, comportano il trasferimento di dati dai registri del processore alla memoria centrale e viceversa.
 - La durata del cambio di contesto dipende sia dal sistema operativo che dall'architettura della CPU. In genere è di alcuni microsecondi.

