POLITECNICO DI MILANO





Processi

Laboratorio Software 2008-2009
C. Brandolese

Introduzione

I calcolatori svolgono operazioni simultaneamente

- Esempio
  - Compilazione di un programma
  - Invio di un file ad una stampante
  - Visualizzazione di una pagina web
  - Riproduzione di musica
  - Ricezione di una mail

Il concetto di processo permette ad un sistema di

- Svolgere più attività simultanee
- Seguirne l'evoluzione

Si introduce la nozione di "stato di un processo"

- I processi passano da uno stato all'altro
- Il sistema operativo svolge operazioni sui processi
  - Creazione, distruzione, sospensione, ripresa, …

Definizione

Un processo è una istanza di un programma in esecuzione

- Concetto dinamico

Un processo dispone di un suo spazio di indirizzamento

- Text
  - Codice che il processo deve eseguire (programma)
- Data
  - Variabili statiche
  - Memoria allocata dinamicamente
- Stack
  - Variabili automatiche
  - Frame di attivazione delle procedure

Categorie

Esistono diverse categorie di processi

- Definite in base alla criticità

Processi batch

- Ricevono tutti i dati d'ingresso all'inizio dell'esecuzione
- Producono tutti i risulatati alla fine
- Non vi è interazione

Processi interattivi

- Eseguono una attività in 'burst' successivi
- Richiedono l'interazione con un utente a terminale

Processi real-time

- Interagiscono con dispositivi o sistemi esterni
- Sono soggetti a vincoli temporali stringenti
- Il mancato rispetto dei vincoli costituisce un problema grave

Gestione

Un sistema operativo fornisce i seguenti servizi per la gestione dei processi

- Creazione
- Distruzione
- Sospensione
- Ripresa
- Cambiamento della priorità
- Blocco
- Risveglio
- Dispatch
- Comunicazione

Stati

Un processo passa attraverso una serie di stati

- Running
  - Il processo è in esecuzione sul processore
  - Non attende alcuna risorsa
  - Un solo processo per ogni processore può essere in questo stato
- Ready
  - Il processo è pronto per l'esecuzione su un processore
  - Attende la disponibilità del processore
- Blocked
  - Il processo è in attesa di un evento

Il sistema operativo mantiene

- Una lista dei processi ready
- Una lista dei processi blocked

Transizioni

Dispatching

- Assegna il processore al primo processo nella lista ready

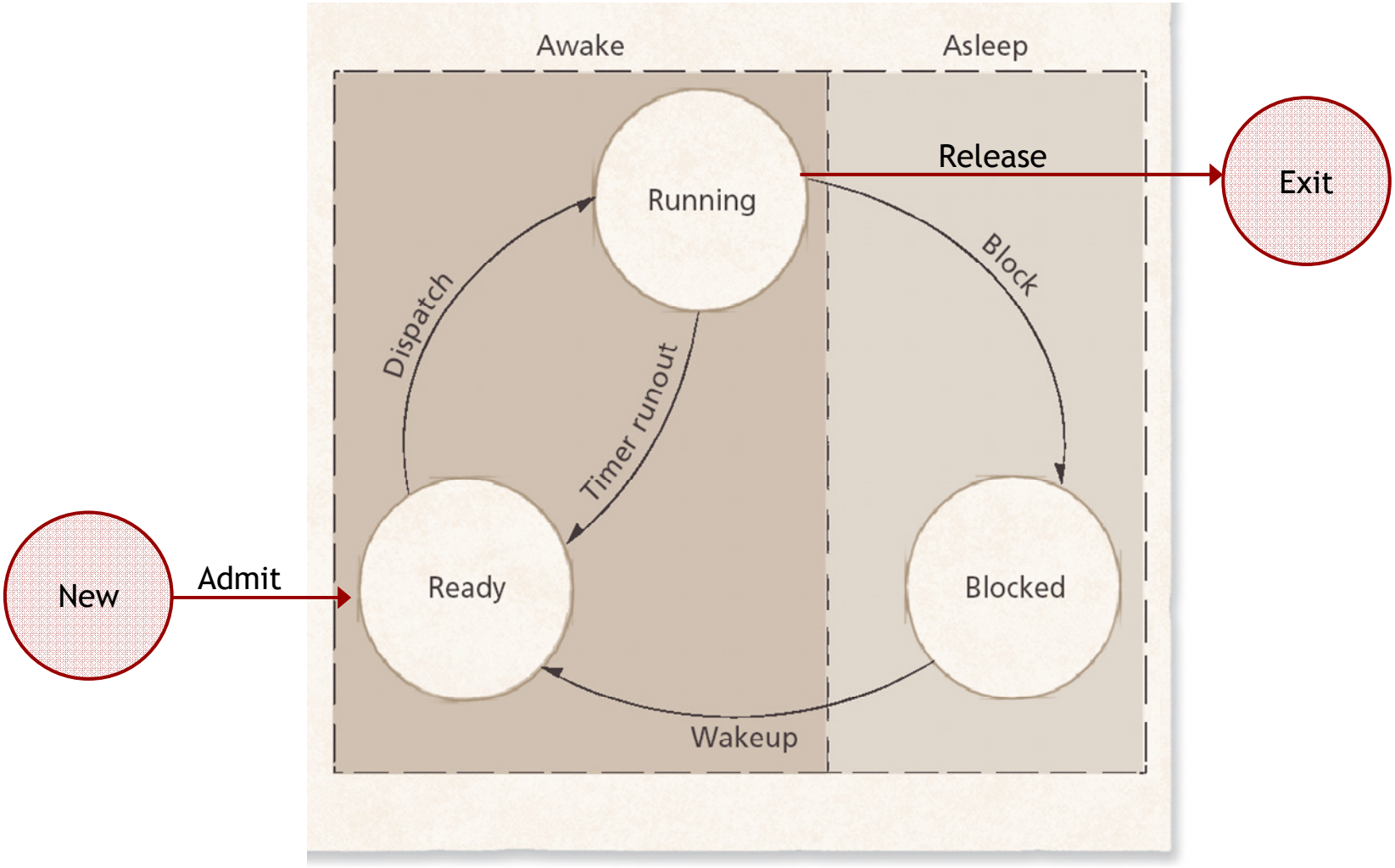
Quanto di tempo

- Il sistema operativo può utilizzare un timer per assegnare ad ogni processo un quanto di tempo in cui è in esecuzione
- Il multitasking cooperativo permette di completare ogni processo

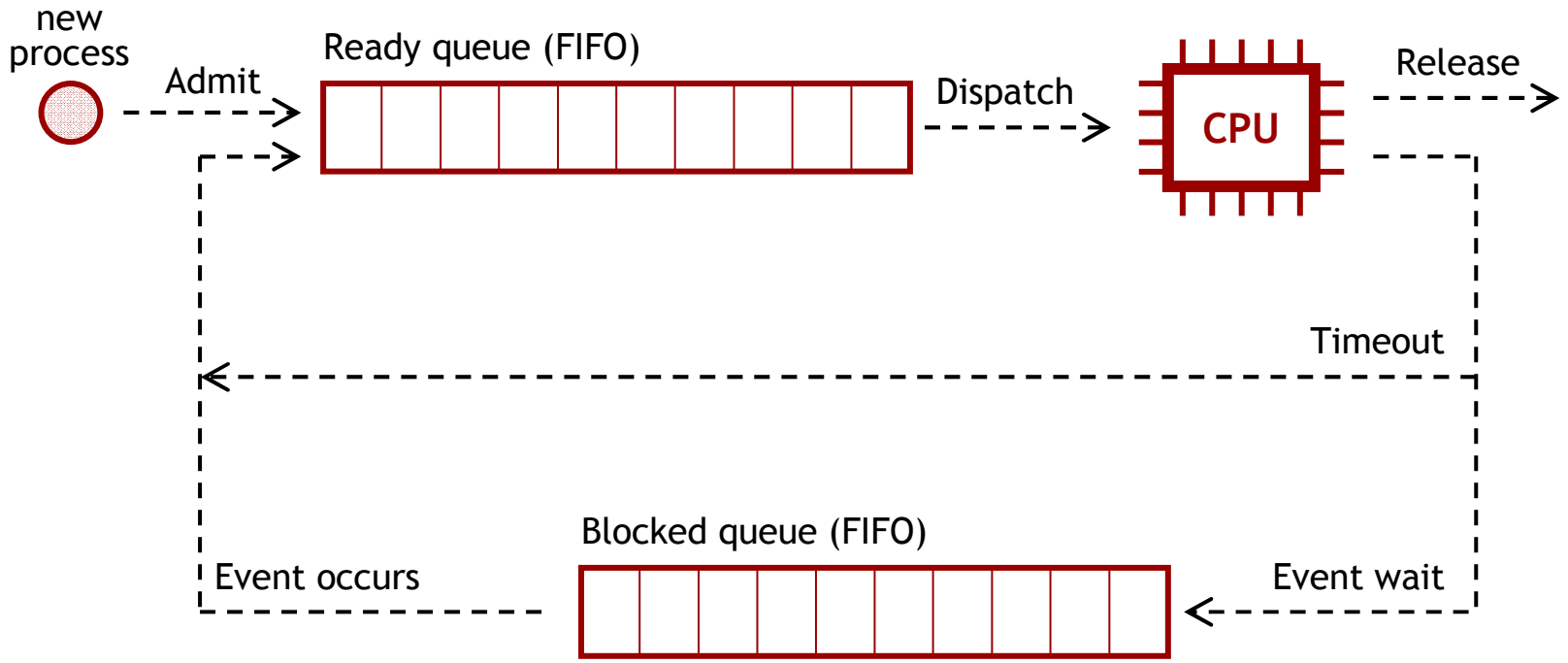
Transizioni di stato

- A questo punto si hanno 4 possibili transizioni
  - Se un processo è dispatched, passa da ready a running
  - Se il quanto di tempo termina, passa da running a ready
  - Se un processo si blocca, passa da running a blocked
  - Se si verifica uno specifico evento, passa blocked a ready

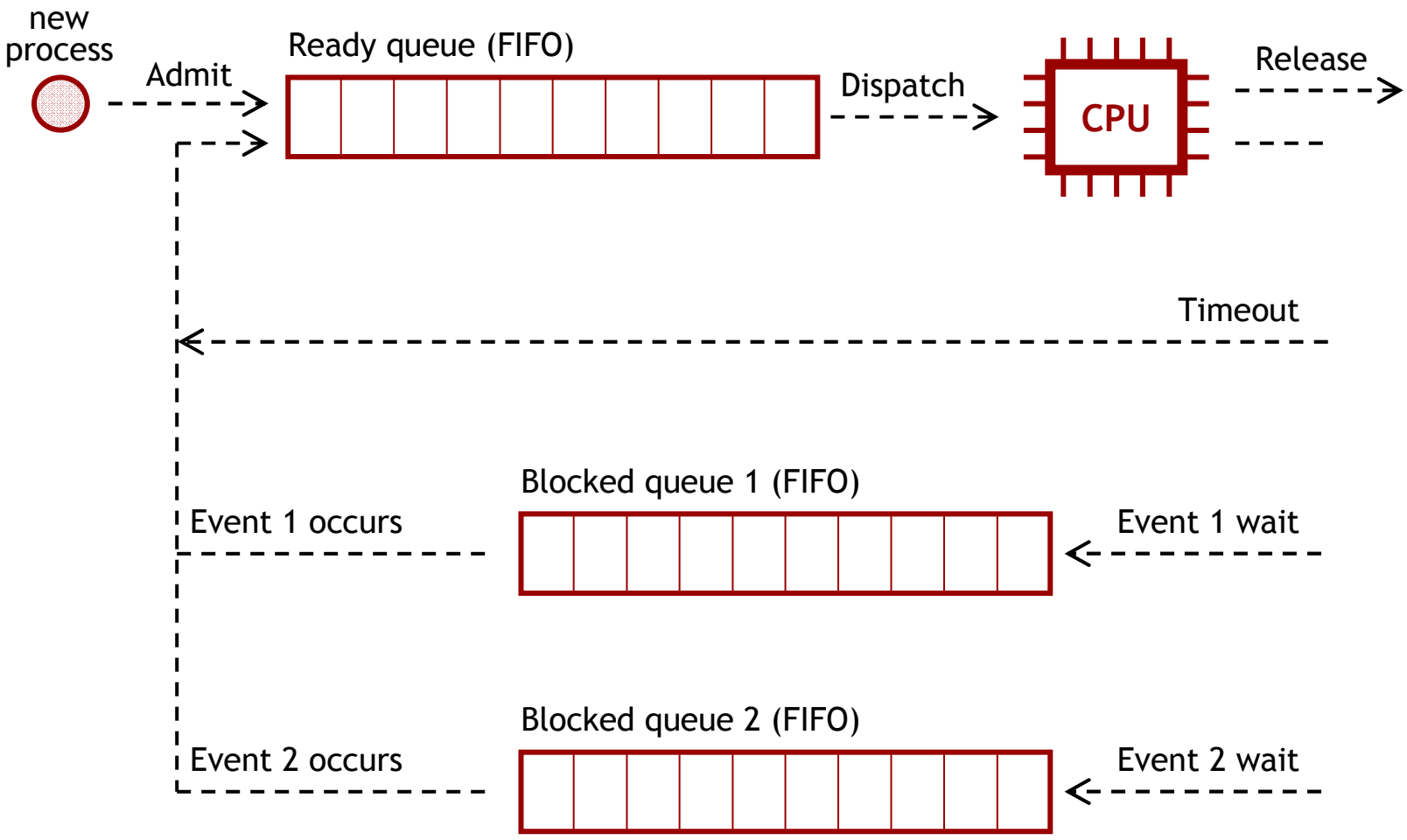
Transizioni



Implementazione a singola coda



Implementazione a code multiple



Descrittori

Un descrittore di processo o Process Control Block (PCB)

- Mantiene le informazioni necessarie al sistema operativo per la sua gestione

Informazioni nel PCB

- Identificatore univoco (PID)
  - Per esempio un indice nella tabella dei processi
- Identificatore del processo padre
  - Cioè del processo che ha creato il proceso in esame
- Identificatore dell'utente
  - Quale utente ha creato il processo
- Registri del processo visibili all'utente (GPRs)
- Regsitri di controllo e di stato
  - Program Status Word, registro EFLAGS nei Pentium
- Process stack pointers

Descrittori

Informazioni nel PCB [continua]

- Informazioni di stato
  - State del processo, priorità di scheduling, evento atteso, …
- Informazioni per interprocess communication
  - Flags, segnali, messaggi, …
- Privilegi
  - Instructions eseguibili, accesso alla memoria, …
- Gestione della memoria
  - Informazioni a proposito della memoria virstuale utilizzata
- Risorse
  - Proprietà
  - Utilizzo

Descrittori

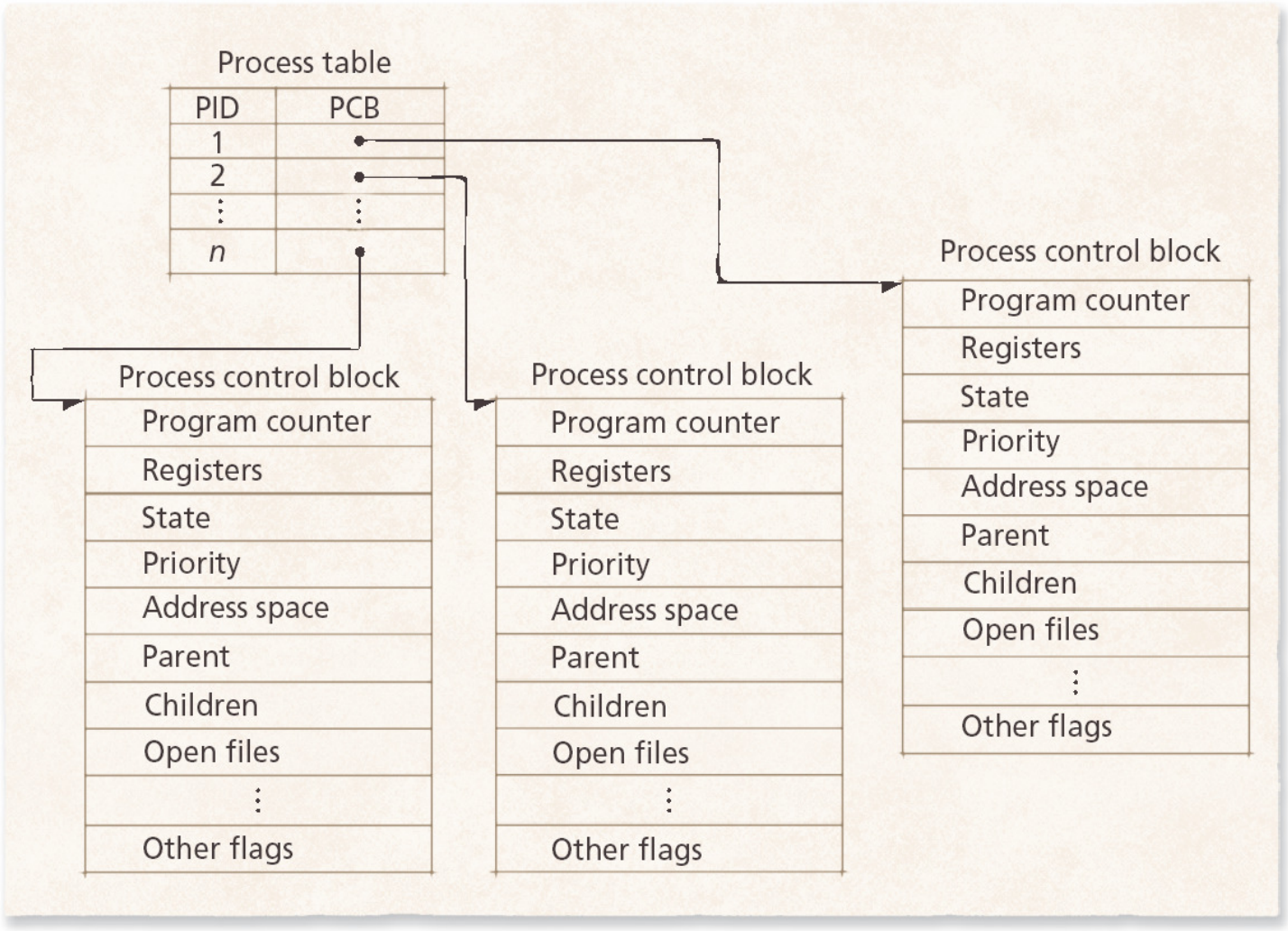
Tabella dei processi o Process Table

- Il sistema mantiene i puntatori ai vari PCB in una tabella
  - Unica per tutto il sistema
  - Specifica per ogni utente

Facilità di accesso ai PCB

- Creazione di un processo
  - Nuovo PCB e nuovo puntatore
  - Allocazione delle risorse necessarie
- Distruzione di un processo
  - Rimozione del PCB
  - Eliminazione del puntatore dalla tabella
  - Rilascio delle risorse

Descrittori



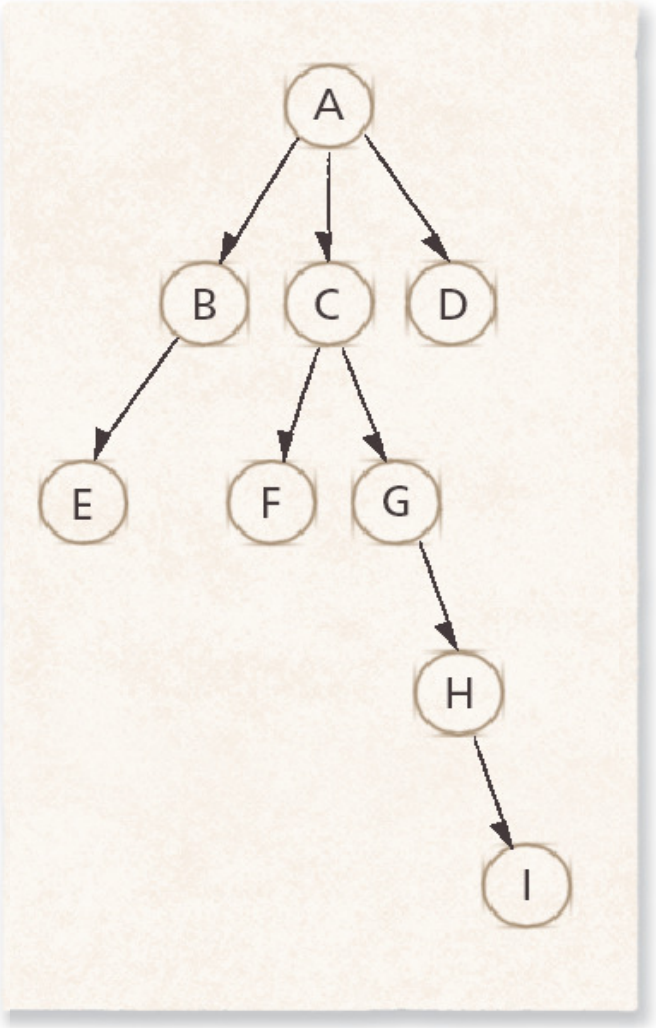
Operazioni – Creazione

Perché può avvenire

- Richiesta di un nuovo job batch
- Login di un utente
- Richiesta di un servizio
  - Spooler, server web, ...
- Richiesta di un altro processo
  - Il creatore è detto padre o parent
  - Il creato figlio o child

Quando un processo termina

- Distrugge tutti i processi figli
- Permette ai figli di completare l'esecuzione indipendentemente dal processo gentitore



Operazioni – Creazione

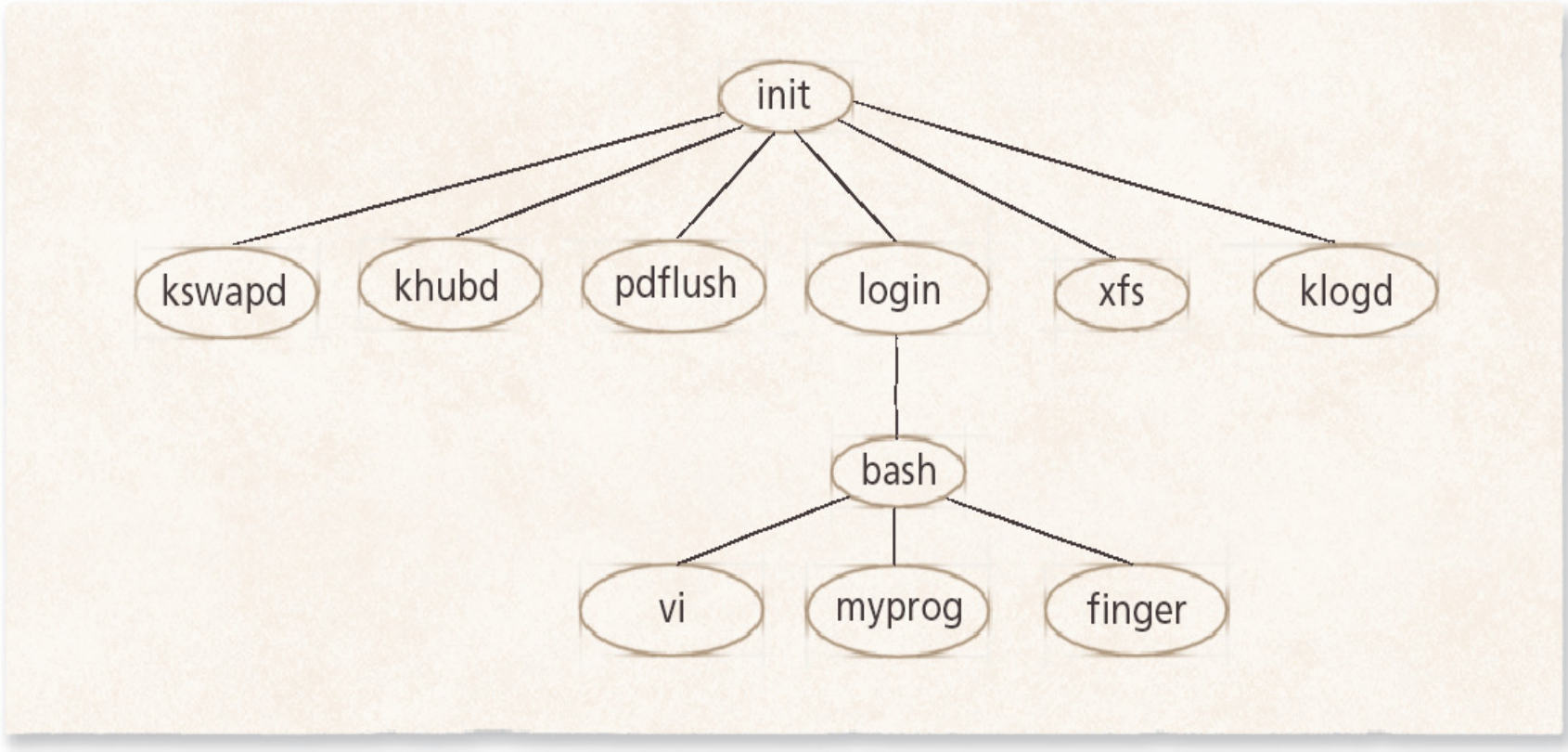
La creazione di un processo comporta le seguenti fasi

- Assegnazione di un nuovo id unico
- Aggiunta di una nuova voce nella tabella dei processi
- Allocazione dello spazio per
  - Descrittore di processo
  - Immagine del processo
- Inizializzazione del descrittore di processo
- Inizializzazione dei puntatori allo stack
- Inizializzazione liste
  - Aggiunta del processo alla lista utilizzata per lo scheduling
- Altre attività
  - Inizializzazione delle attività di accounting

Operazioni – Creazione

Esempio:

- La gerarchia dei processi in Linux



Operazioni – Distruzione

Perché può avvenire

- Terminazione naturale
- Richiesta di un altro processo
- Superamento di un limite temporale
- Errore o malfunzionamento
  - Errori di memoria (non disponibile, out of bound, …)
  - Errore di protezione: Scrittura su un file read-only, …
  - Errore artimetico: Divisione per 0, …
  - Errore di I/O
  - Istruzione non valida: esecuzione di ‘dati’, istruzione privilegiata

La distruzione di un processo richiede le seguenti fasi

- Richiesta d’intervento del sistema operativo
- Output dei dati verso il processo padre
- Deallocazione delle risorse associate

Operazioni – Sospensione e Ripresa

La sospensione

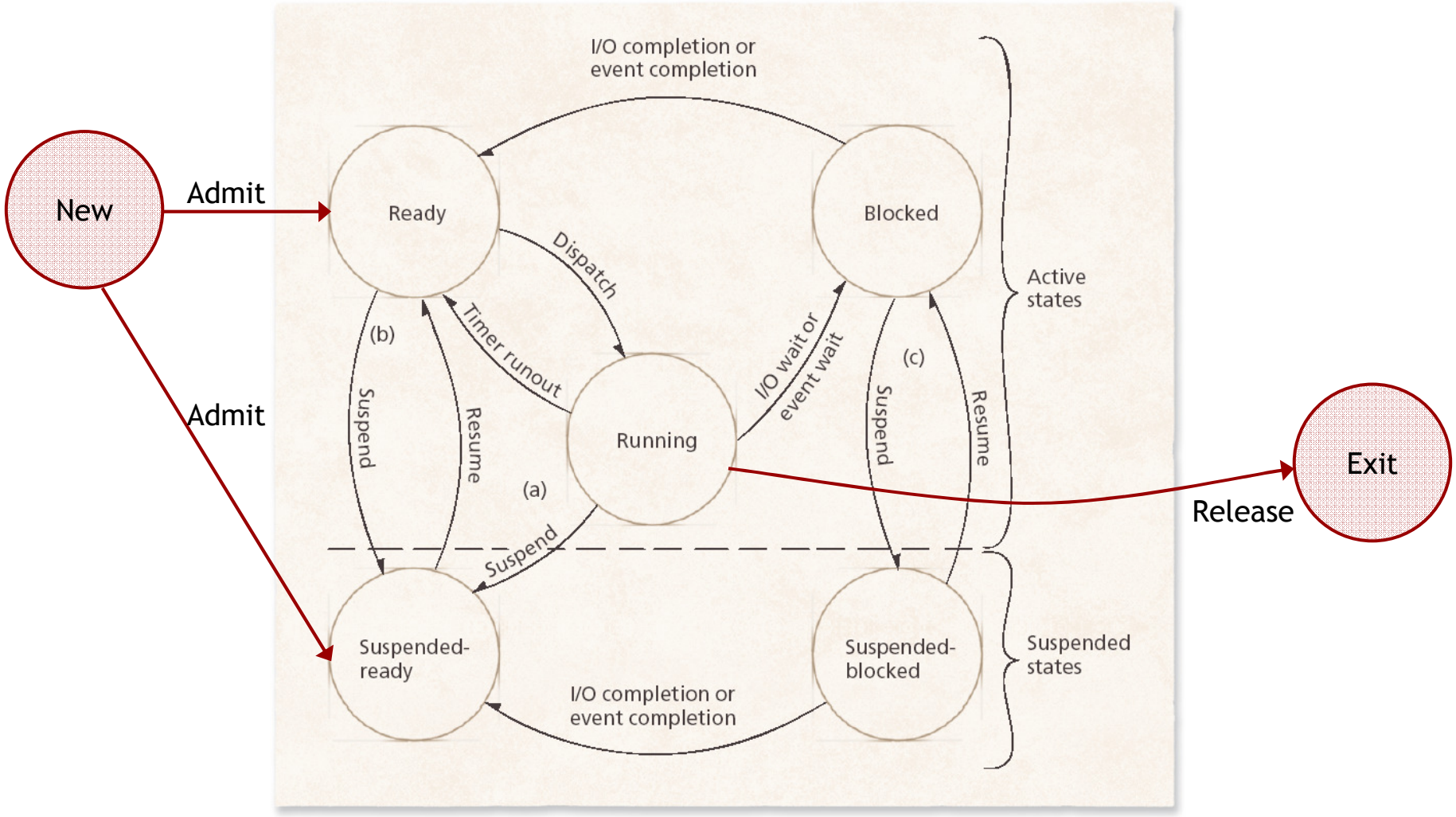
- Elimina un processo dalla competizione per il processore
- Non distrugge il processo
- Molto uitle per
  - Debugging, individuare minacce alla sicurezza, ...
- Può essere richiesta
  - Dal processo stesso che viene sospeso
  - Da un altro processo
- Si definiscono due stati di sospensione
  - Suspended-Ready
  - Suspended-Blocked

La ripresa di un processo

- Deve essere richiesta da un altro processo

Operazioni – Sospensione e Ripresa

Transizioni di stato con sospensione e ripresa



Context Switch

Eseguito dal sistema operativo per

- Interrompere l'esecuzione di un processo running
- Iniziare l'esecuzione di un processo ready

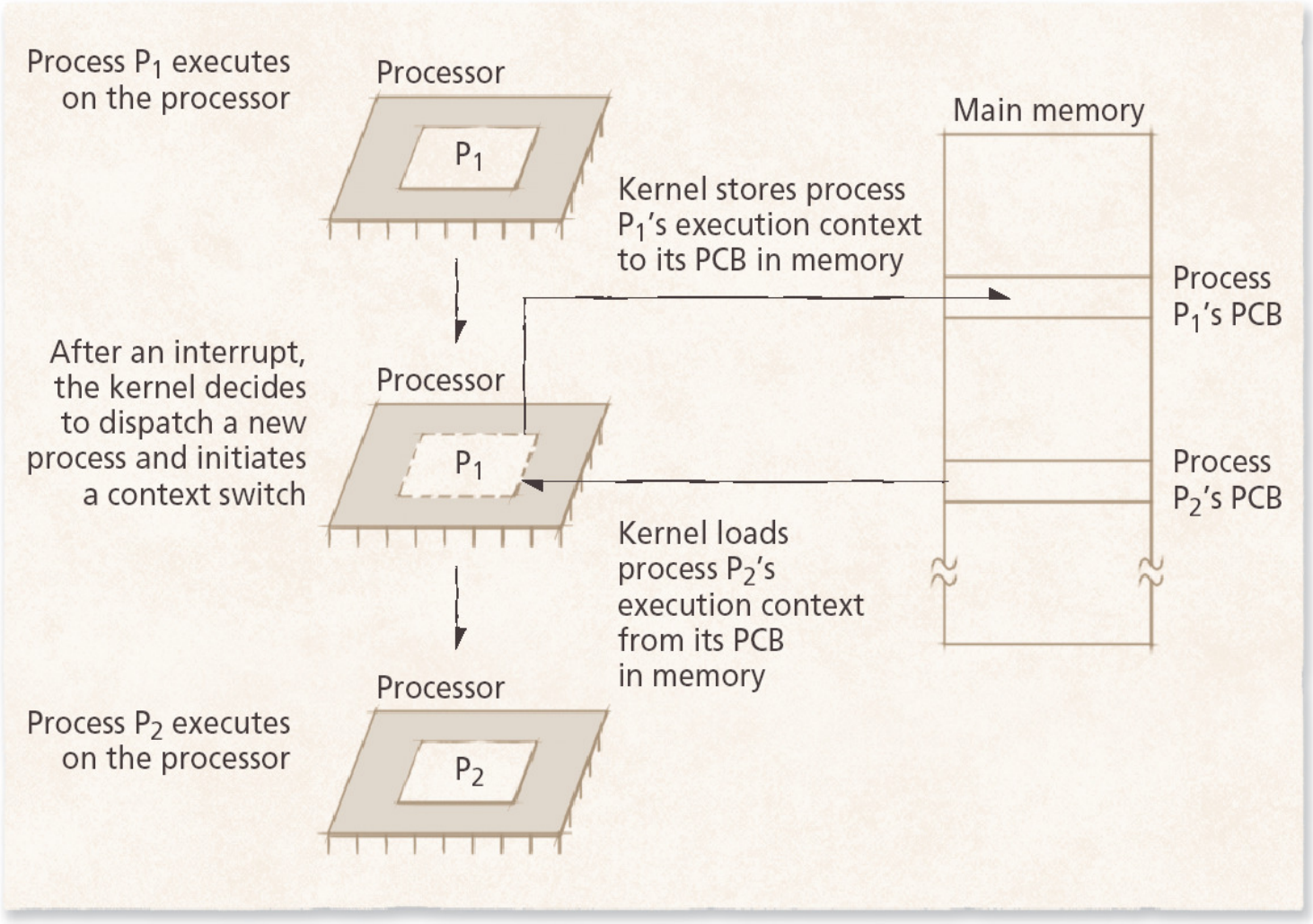
Il context switch comporta

- Salvataggio del contesto del processo running nel corrispondente PCB
- Caricamento del contesto del processo ready dal corrispondente PCB

Il context switch

- Deve essere trasparente al processore
  - Il processore non può fare nulla di utile nel frattempo
  - Il sistema operativo deve quindi minimizzare il tempo di context-switch
- In alcune architetture viene svolto da hardware dedicato

Context Switch



Context Switch

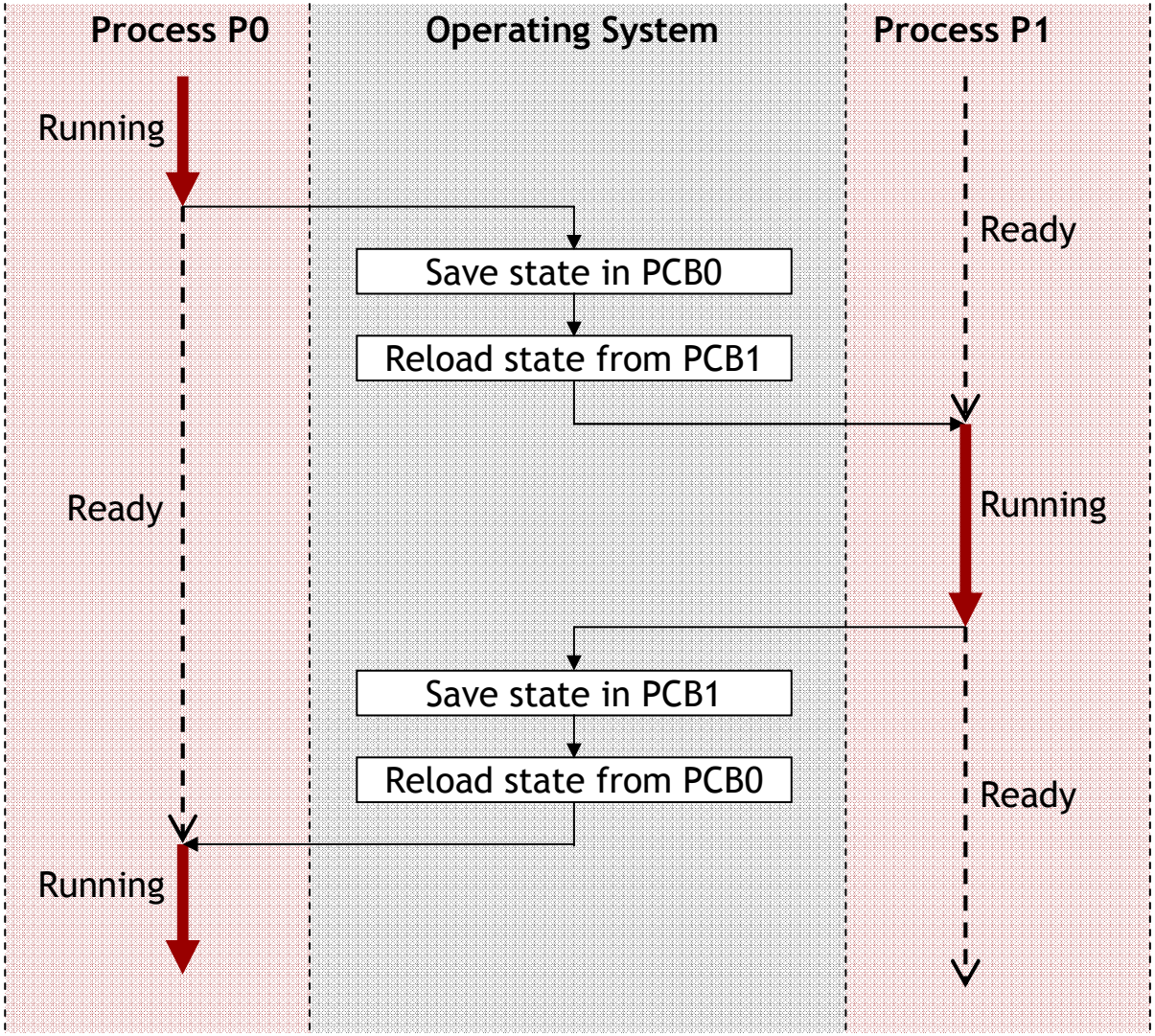
Quando avviene

- Interrupt – evento esterno
  - Clock: il processo ha esaurito il quanto di tempo
  - I/O: un altro richiede la stessa risorsa di I/O
  - Memory fault: la richiesta di un indirizzo si riferisce ad una pagina della memoria virtuale che deve essere caricata
- Trap – Evento interno
  - Tipicamente un errore
  - Se l'errore è fatale causa la distruzione del processo
- Chiamata di sistema
  - Il processo richiede un servizio di sistema operativo

Il passaggio ad una routine di servizio

- Richiede il salvataggio del contesto del chiamante
- Non modifica lo stato del processo interrotto

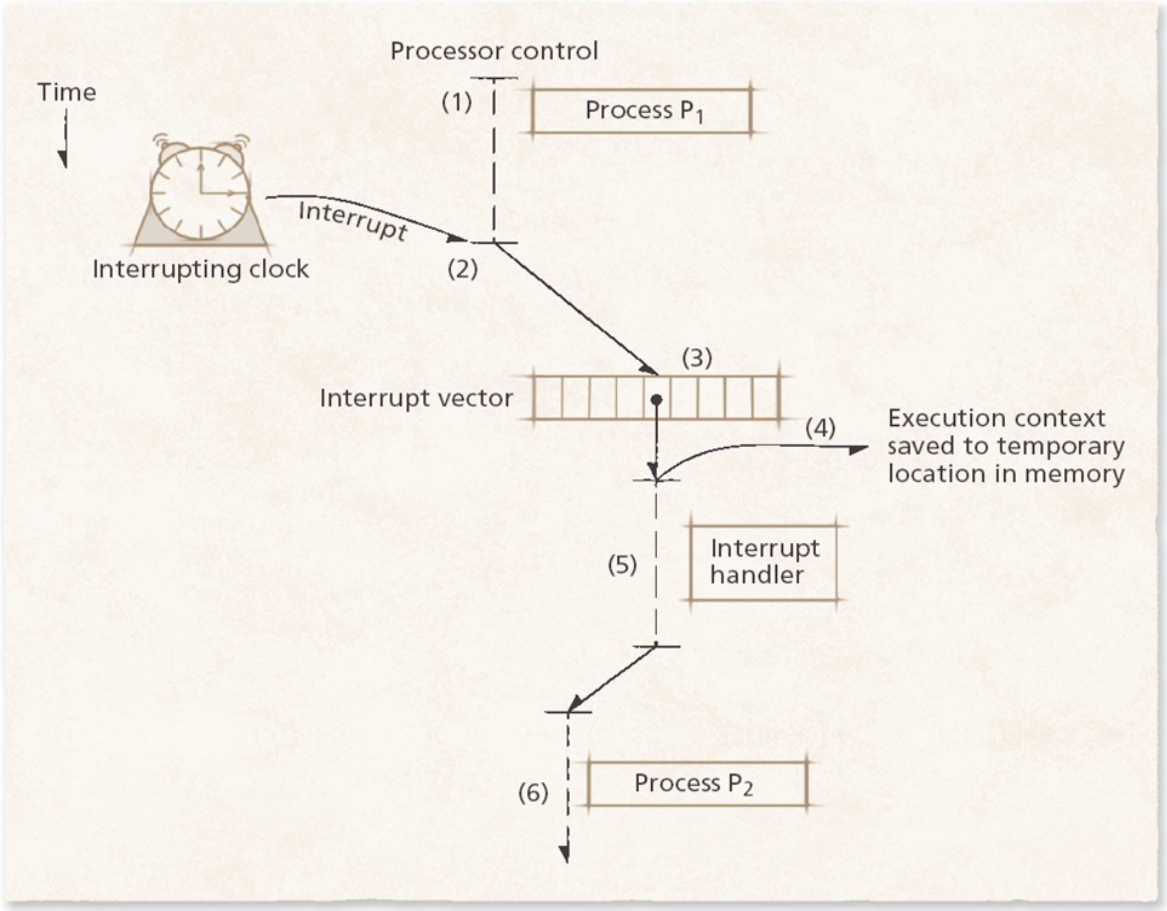
Context Switch



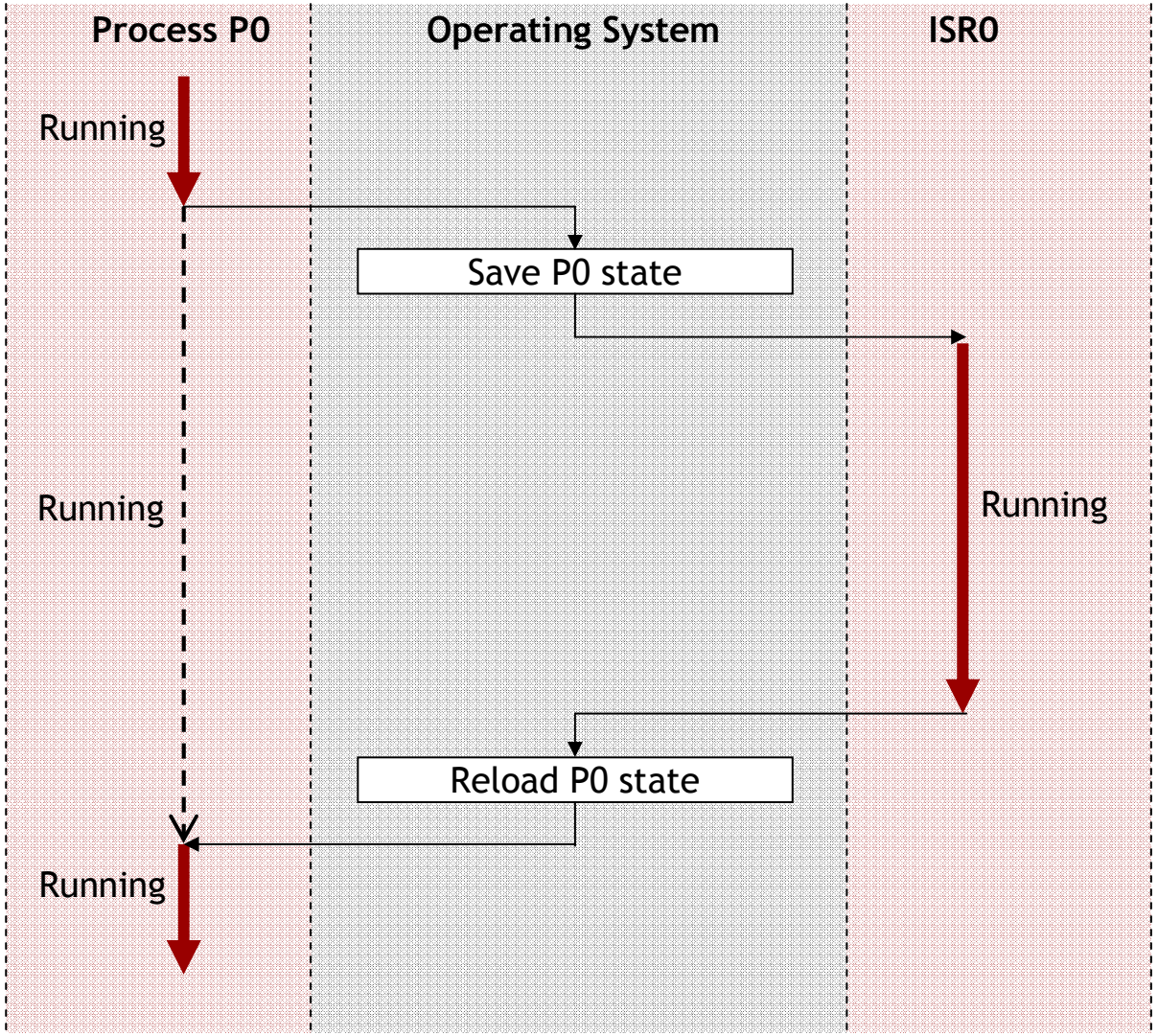
Context Switch e interrupt

Un interrupt

- Flusso di esecuzione simile a due processi che si alternano
- Context switch è realizzato mediante un meccanismo specifico



Context Switch



IPC – Inter-Process Communication

Molti sistemi operativi forniscono un meccanismo per la comunicazone tra processi

- I processi devono poter comunicare per esepio
  - In ambienti multiprogrammati
  - In ambienti e sistemi operativi di rete

La comunicazione e la sincronizzazione

- Sono essenziali per coordinare più processi che concorrono a realizzare una funzione comune

Sistemi operativi diversi dispongono di meccanismi diversi per la comunicazione e la sincronizzazione

IPC – Segnali

Un segnale o interrupt software

- Utilizzato per notificare un evento ad un processo
- Diversi tipi di segnali a seconda dei tipi di eventi
  - Allarmi, Errori, ...

Si tratta di un meccanismo di sincronizzazione

- I processi non possono scambiarsi dati mediante segnali

Un processo può decidere come trattare un segnale

- Accettare e gestire (catch)
  - È necessario specificare una routine di gestione
- Ignorare (ignore)
  - Viene utilizzata una routine di gestione di default del sistema operativo
- Mascherare (mask)
  - Una maschera di bit indica quali segnali devono essere 'consegnati' ad un processo per la gestione e quali no

IPC – Messaggi

La comunicazione tra processi può usare messaggi

- Anche altri meccanismi: memoria condivisa, ...

I messaggi possono essere

- Monodirezionali
  - Un processo è il mittente (sender) e l'altro è il ricevente (receiver)
- Bidirezionali
  - Ogni processo può agire come sender o come receiver
- Bloccanti
  - Il receiver deve notificare al sender (che è in attesa) l'avvenuta ricezione
- Non-bloccanti
  - Il sender può contunuare la sua attività subito dopo l'invio

Diverse implementazioni

- Pipe, FIFO, socket, ...

Esempio: I processi UNIX

Tutti i processi dispongono di

- Un insieme di indirizzi di zone di memoria
  - Implementano uno spazio di indirizzamento virtuale
  - Si utilizza la cosiddetta virtual address table

Il PCB di ogni processo

- È mantenuto dal kernel in una zona di memoria protetta
  - I processi utente non possono accedervi
- Un PCB contiene
  - I registri del processo
  - Il PID
  - Il program counter
  - Lo stack di sistema

Tutti i processi (PCBs) sono contenuti nella process table

Esempio: I processi UNIX

I processi

- Interagiscono con il sistema operativo per mezzo delle system call o chaimate di sistema
- Hanno una priorità indicata da un valore tra -20 e 19
  - Un valore numerico basso indica una alta priorità

Un processo può generare altri processi

- Un processo viene creato con la chiamata di sistema **`fork()`**
  - Crea una nuova copia del processo padre
  - Il processo figlio riceve una copia di tutte le risorse del padre

Comunicazione

- UNIX dispone di diversi meccanismi per la comunicazione
  - Pipe, FIFO, socket, ...

Esempio: I processi UNIX

Alcune chaiamte di sistema:

`fork()`	Crea un nuovo processo ed una copia delle risorse
`exec()`	Carica un programma nello spazio di un processo
`wait()`	Blocca il processo chiamante in attesa della fine di un suo processo filgio
`signal()`	Installa una routine di gestione per uno specifico tipo di segnale
`exit()`	Termina il processo corrente
`kill()`	Invia un seganle ad un determinato processo
`nice()`	modifica la priorità di scheduling di un processo