Componenti comuni dei SO

- Process management: gestione dei processi
- Protection system
- Networking: gestione internet
- Command-interpreter system: interfaccia verso l'utente

Gestione dei processi

Un processo è un programma in esecuzione, che richiede risorse tra cui tempo CPU, memoria, file, dispositivi I/O. Si occupa della creazione/sospensione dei processi.

System Call

Interfaccia tra il SO e l'ambiente esterno, disponibile in assembly e altre lingue. Le system call sono chiamate molto generiche che richiedono molti parametri.
I system program, o "utility" sono implementate con le system call.

System Structure

I primi SO erano scritti senza preoccuparsi troppo dei problemi di ingegneria del software, erano istruzioni scritte in un solo blocco (non divise in moduli).
Al giorno d'oggi il SO UNIX è costituito da

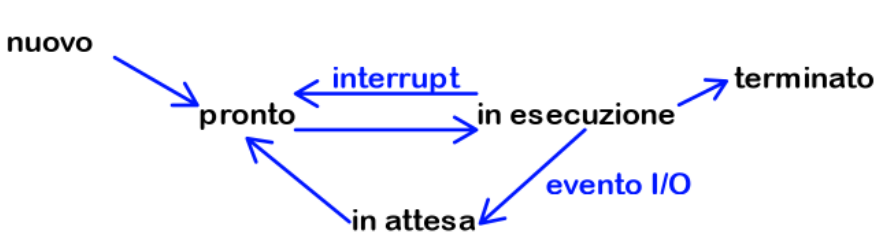
- programmi di sistema
- kernel, che fornisce file system, gestione della memoria e altre funzioni del SO
  I SO sono divisi con una struttura a livelli, questo cambiamento è dovuto alla necessita di una struttura più dinamica dovuta in parte all'avento di internet.

Microkernel

Il kernel conitne solo le funzioni più essenziali: i processi di base e la gestione della memoria.
I servizi principali del SO sono poi aggiunti in cima come moduli che interagiscono sfruttando i microkernel. Questi avvantaggiano l'estensione, portabilità e rendono più facili da modificare i SO.

Processi

Processo: un programma in esecuzione, deve avanzare in maniera sequenziale. Un processo contiene program counter, stack, sezione dei dati. Durante l'esecuzione, un processo cambia stato.



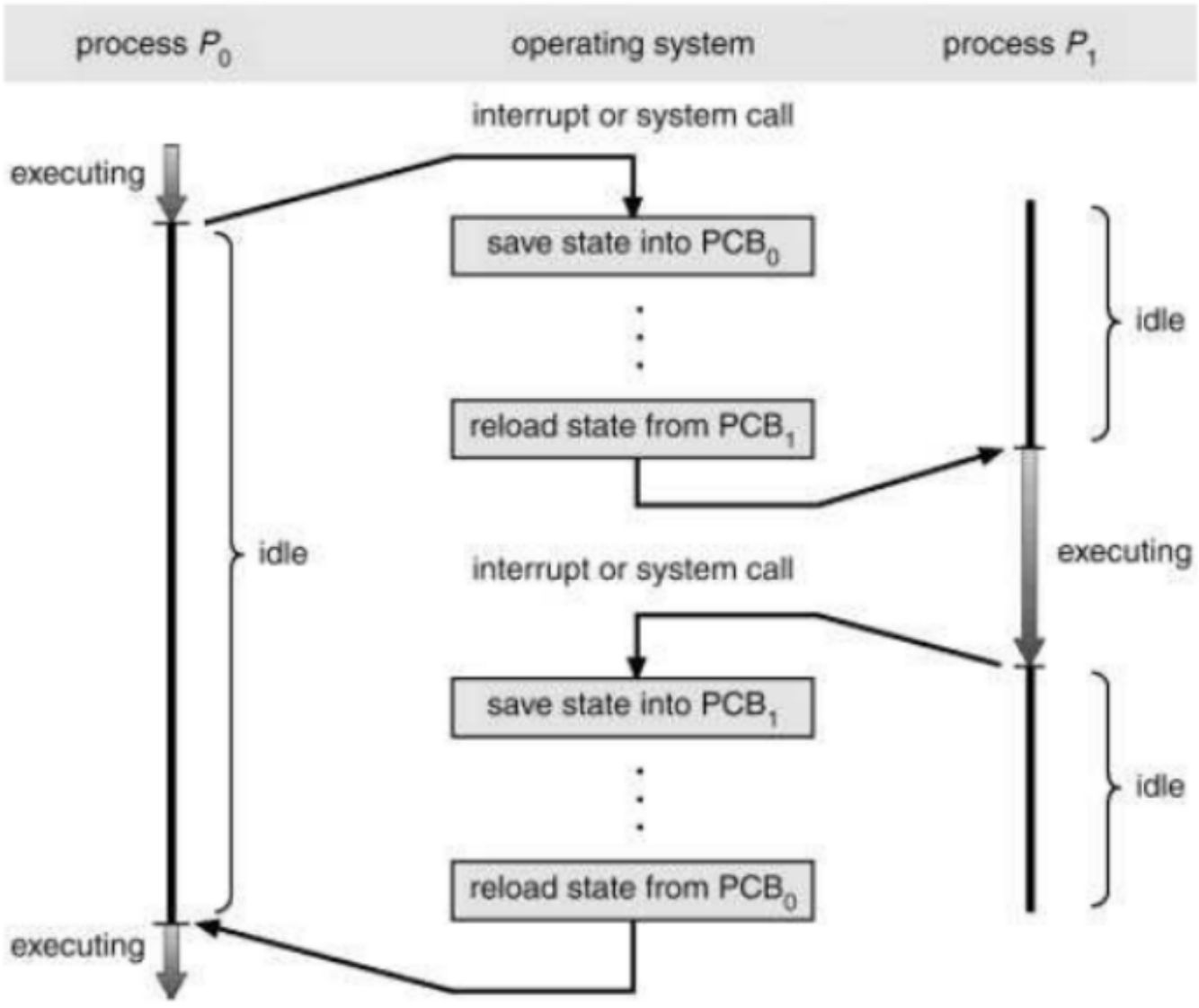
La creazione dei nuovi processi viene gestita dal SO. Quando terminano tutti i dati in memoria non servono più e vengono cancellati.
Un processo in attesa può solo diventare ready. Verrà poi deciso se potrà essere eseguito o no.

Process Control Block (PCB)

Gestisce tutte le informazioni relative ai processi, è in grado di gestire i context switch (passaggio da un processore ad un altro).
Bisogna assicurarsi che dopo che c'è stato un context switch, il processo può continuare come prima.
Bisogna tenere traccia di questo momento dell'esecuzione per ripartire da esso salvando i contenuti del processo sul PCB.

Esecuzione Context Switch

- Durante l'esecuzione di P0 deve essere effettuato un context switch
- Lo stato di P0 viene salvato nel PCB di P0
- Vengono lette dal PCB di P1 le istruzioni per far riprendere il P1
- C'è un'altra interrupt, si salva lo stato di P1 nel suo PCB e si carica lo stato di P0 dal suo PCB.



Creazione e terminazione di processi

Molti processi (di sistema) sono creati quando la macchina viene avviata oppure un utente può richiedere la creazione di un nuovo processo. Se due processi lavorano sulle stesse risorse, queste vengono duplicate solo se necessario.
La terminazione dei processi può essere di vari tipi:

- normale
- dovuta ad un errore
- dovuta a gravi errori (Fatal Error)
- Il processo viene sovrascritto da un altro processo(Kill)