

Introduzione ai sistemi operativi

Vittorio Zaccaria

November 29, 2018

Concetti di base

Gestione delle risorse

Processi

Files e file system

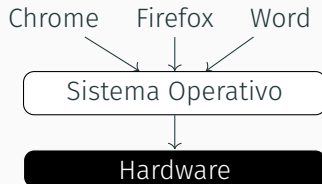
Concetti di base

Cosa è un sistema operativo

- E' un **programma** che gira permanentemente sul calcolatore
- Rende **operativo** un calcolatore che altrimenti sarebbe un ammasso di ferraglia immobile.
- Esempi: Microsoft Windows, macOS, Linux ed altri ancora meno conosciuti.

Perché serve?

E' un **intermediario** fra i programmi e i componenti fisici (**hardware**) del calcolatore.



Perché serve?

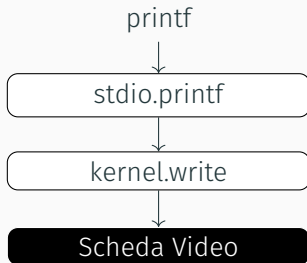
- Quando un programma deve accedere ad una risorsa (e.g., video o tastiera) manda un messaggio (**system call**) al sistema operativo.
- Effettivamente delega l'operazione complessa al sistema operativo.
- Le operazioni per manipolare a basso livello l'hardware sono state scritte una volta sola da chi ha sviluppato il sistema operativo o i suoi **driver**.

Perché serve?

- Quando scrivete del codice per stampare a video:

1 `printf("Hello world\n");`

state mandando un messaggio attraverso diversi livelli di sistema operativo:



Funzioni principali

Le funzioni principali sono:

- controllo e gestione delle componenti hardware (**periferiche**)
- esecuzione dei programmi. Un programma in esecuzione è chiamato **processo**.
- gestione dell'archiviazione (**files**).

Per fornire tali funzioni, il sistema operativo implementa una adeguata **gestione delle risorse**.

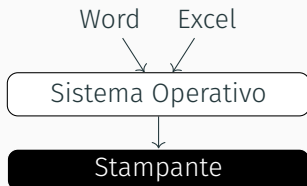
Gestione delle risorse

Cosa intendiamo come risorse?

- Una **risorsa** è qualsiasi entità utilizzabile all'interno di un calcolatore
- E.g.: stampante, CPU, memoria e così via

Cosa intendiamo con gestione?

- In un qualsiasi momento, più di un processo potrebbe voler richiedere l'uso di una risorsa.
- Cosa succederebbe se volessimo stampare contemporaneamente da due programmi?



- In questo caso il sistema operativo svolge la funzione di arbitro facendo in modo che le stampe non si mischino.

Fra i concetti principali che si incontrano nell'ambito dei sistemi operativi vi sono:

- **concorrenza**: è la caratteristica di un insieme di funzioni/algoritmi/applicazioni che potenzialmente possono essere eseguite simultaneamente. Non implica **parallelismo** 'effettivo'.
- **sincronizzazione**: è l'insieme di tecniche utilizzate per ordinare temporalmente le operazioni; necessario quando più operazioni concorrenti necessitano di accedere alla stessa risorsa

- La risorsa più importante di un calcolatore è la CPU.
- Il sistema operativo permette l'uso condiviso della CPU fra più programmi (multi-programmazione), assegnando ripetutamente un quantum di tempo di alcuni millisecondi a ciascuno dei programmi in esecuzione (timesharing).
- Il timesharing è l'esempio più avanzato di gestione dei processi.

Processi

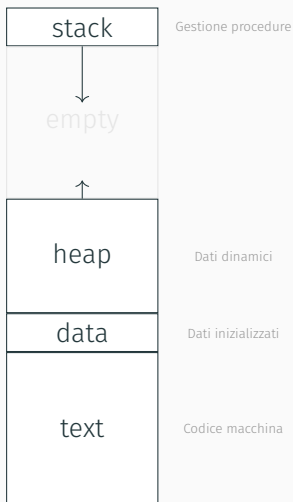
Cosa è un processo

Un **processo** è un insieme di informazioni utilizzato dal sistema operativo per eseguire un programma e comprende:

- il programma da eseguire (ovvero, come abbiamo visto, dal codice macchina dello stesso).
- i dati che tale programma sta elaborando e tutte le risorse che sta utilizzando.
- lo stato di avanzamento dell'elaborazione (ovvero il **program counter**).

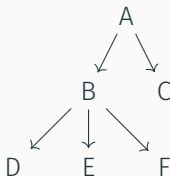
Struttura in memoria di un processo (text, data, stack)

Possiamo distinguere alcune parti fondamentali di questa struttura dati in memoria:



“Genealogia” dei processi

- I processi possono creare altri processi, creando una struttura ad albero

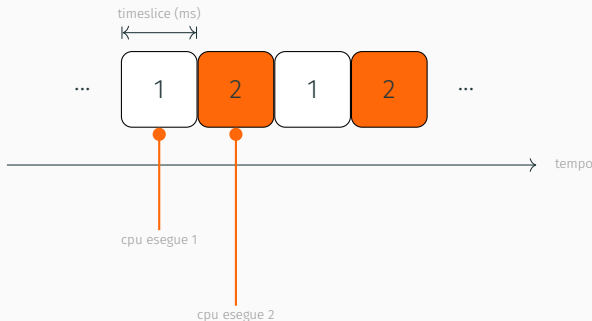


- Il primo processo eseguito dal computer in fase di **boot** è il processo `init`.

```
1 > pstree
2 init─Plex Media─Plex DLNA Serve─14*[{Plex DLNA Serve}]
3                  └Plex Script Hos─12*[{Plex Script Hos}]
4                  └─17*[{Plex Media Serv}]
5                  └─_thttpd_
6                  └─agent
7                  ...
```

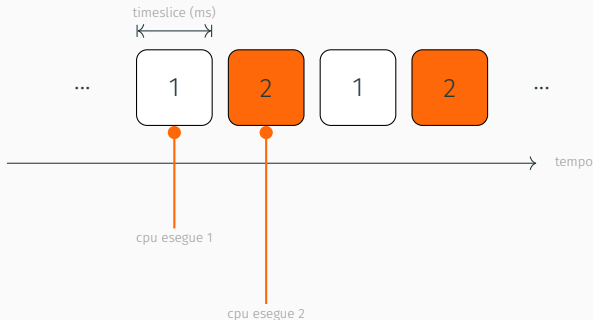
Condivisione del processore

Il processore viene condiviso nel tempo ai vari processi del sistema; supponiamo di avere due processi, che occupano il processore ad intervalli alterni:



Condivisione del processore

- Il cambio di assegnamento fra (1) e (2) si chiama **cambio di contesto**.
- Può durare qualche frazione di secondo e quindi, più è frequente il cambio di contesto, più il sistema può sembrare lento.



- Supponiamo che il processo (1), se fosse l'unico nel sistema, effettua p operazioni al secondo.
- Se c'è anche (2) attivo, il processo (1)
 - avrà solo per 50% il tempo la CPU e
 - alla fine di ogni timeslice ci sarà un cambio di contesto

Generalizzando:

- Supponiamo che il timeslice sia di t sec. e che il cambio di contesto duri u sec., quante operazioni effettive farà il processo 1 in un secondo, al variare del numero dei processi n nel sistema ($n > 1$)?
- Il numero di cambi di contesto al secondo è

$$c = \frac{1}{u + t}$$

- Il tempo perso per i cambi di contesto in un secondo è

$$cu = \frac{u}{u + t}$$

Allora:

- Il processo 1 avrà a disposizione, in un secondo:

$$\frac{1}{n} \left(1 - \frac{u}{u+t} \right)$$

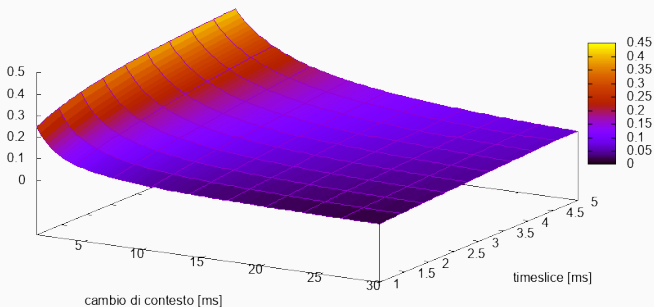
dove n è il numero dei processi nel sistema.

- La velocità effettiva di (1) è

$$\frac{p}{n} \left(1 - \frac{u}{u+t} \right)$$

Condivisione del processore / performance massime

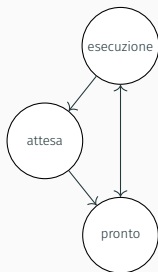
Operazioni al secondo effettive, processo (1) con $n=2$



Stati di un processo

Quando un processo non è in esecuzione, può essere:

- in stato di **pronto** se gli è scaduto il quanto di tempo assegnato, oppure
- in **attesa** se ha richiesto qualche operazione alle periferiche.



Lo **scheduler** decide quali processi in stato di pronto far passare in esecuzione. Se il sistema è multi-cpu, ci possono essere più processi in esecuzione contemporanea..

Files e file system

Cosa è un file system

- Meccanismo del sistema operativo con il quale i file sono posizionati e organizzati su un dispositivo di archiviazione (e.g., memoria di massa, disco rigido o chiavetta USB).
- Fornisce una visione omogenea delle informazioni (sia in termini di contenuto che di posizione) che **astrae** dalla loro effettiva implementazione su periferica.

Organizzazione gerarchica

I **nomi assoluti** dei file sono tipicamente gerarchici, ovvero sono composti da un **prefisso complesso**, da un **nome** e da un'**estensione**.

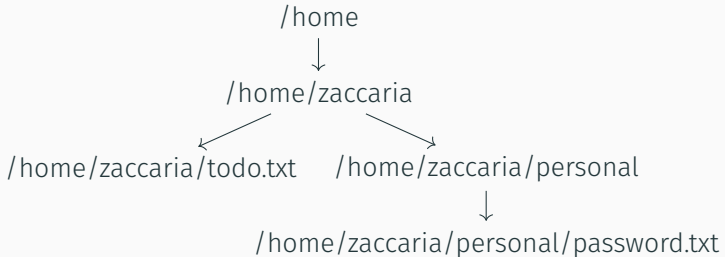


Un prefisso complesso viene comunemente detto **cartella**, quindi file con lo stesso prefisso **appartengono alla stessa cartella**.

Siccome un prefisso complesso può essere scomposto in prefissi più semplici, si dice anche che una **cartella può essere contenuta in un'altra**. Ad esempio `/home/zaccaria/` è contenuto in `/home`.

Organizzazione gerarchica

E' quindi possibile utilizzare le relazioni di contenimento fra cartelle e files per rappresentare la struttura del file system.



- E' possibile specificare per ciascun file qual degli utenti può accedervi e quali operazioni può effettuare.
- In UNIX/Linux possiamo specificare le autorizzazioni in lettura, scrittura ed esecuzione per il proprietario del file, il gruppo a cui appartiene e il resto degli utenti.