

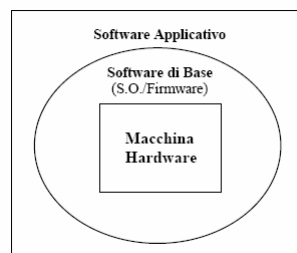
Sistemi Operativi

Definizioni iniziali

- Programma: una sequenza di istruzioni elementari
- Processo: un programma in esecuzione
- *Software*: insieme dei programmi

Tipi di software

- Base: lavora sull'*hardware* e fornisce le funzioni essenziali
- Applicativo: si basa sul *software* di base e soddisfa esigenze dell'utente



Sistema Operativo

- Ha due funzioni principali
 - gestisce le risorse fisiche (*hardware*)
 - controlla i processi e mette a disposizione un'interfaccia per l'utente

Avvio del calcolatore

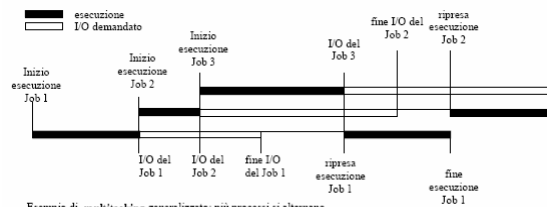
- La memoria ROM contiene il *software* necessario all'avvio (*firmware*)
 - carica il Sistema Operativo in memoria
 - controlla l'integrità delle componenti interne
 - verifica le periferiche collegate (ora *plug & play*)

Tipi di Sistemi Operativi

- *monotasking*: non più usati (DOS)
- *multitasking*: permettono l'esecuzione di più processi in parallelo (Windows)
- *real-time*: oltre al parallelismo, garanzie sull'esecuzione

Sistemi multitasking

- Sono i più usati: Windows, Linux, Mac OS X
- Sfruttano la tecnica di *time-sharing*



Esempio di multitasking generalizzato: più processi si alternano nell'uso della CPU

Interfacce

- A caratteri: i comandi sono stringhe di testo
 - DOS
 - UNIX / Linux
- Grafiche: sfruttano metafore grafiche e contesti
 - Windows
 - Mac OS X

Interfacce a caratteri

- Svantaggi
 - difficili
- Vantaggi
 - veloci
 - flessibili
 - richiedono poche risorse

Interfacce grafiche

- Svantaggi
 - molte risorse
 - poco flessibili
- Vantaggi
 - facili
 - maggior trasparenza rispetto all' *hardware*

Software applicativo

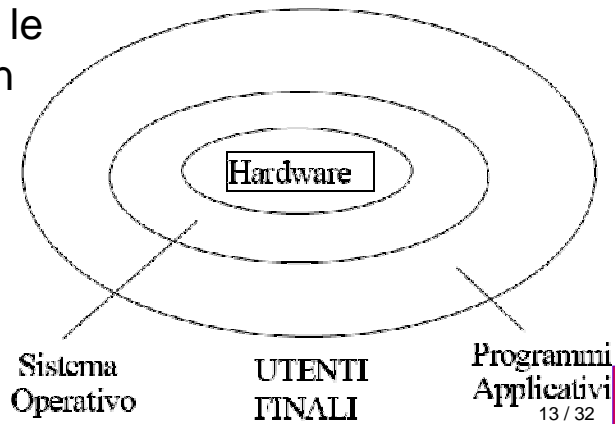
- Si cerca di costruirlo con interfacce amichevoli
- Tipologie
 - ambienti di sviluppo (programmazione)
 - produttività personale (e.g. elaborazione di testi)
 - programmi verticali “specializzati” (e.g. biblioteche)

Esempio di interfaccia di un programma

- Ciascun programma può fornire interfacce aggiuntive rispetto a quelle fornite dal S.O.

Macchine virtuali

- Si nascondono le funzionalità non necessarie
- Si uniforma l'accesso alle risorse



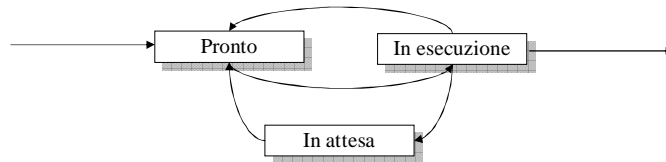
Macchine virtuali

- Anche il Sistema Operativo è organizzato in maniera simile

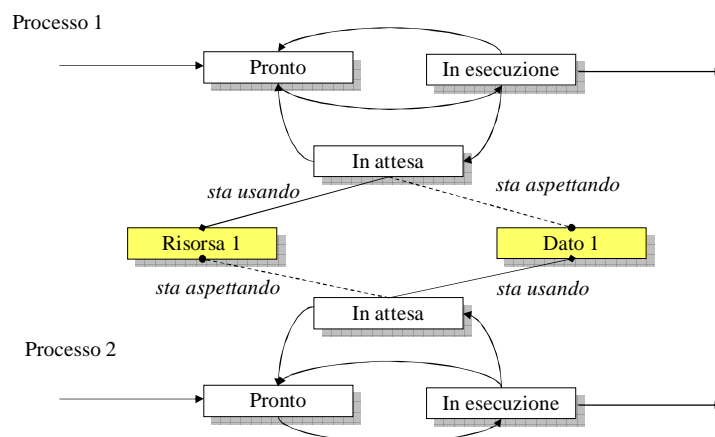


Gestore dei processi (kernel)

- Coordina l'esecuzione di più processi
 - in esecuzione (tanti quanti le CPU)
 - pronto
 - in attesa (di dati o risorse)



Lo stallò

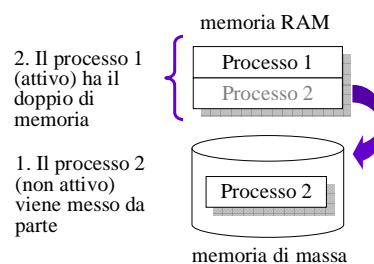


Esempio di gestore di processi

- In Windows esiste un programma chiamato Task Manager
 - CTRL + ALT + CANC

Gestione della memoria centrale

- Assicura l'accesso corretto alla memoria RAM da parte dei programmi
- Interagisce con la memoria secondaria per aumentare la capacità: memoria virtuale



Esempio

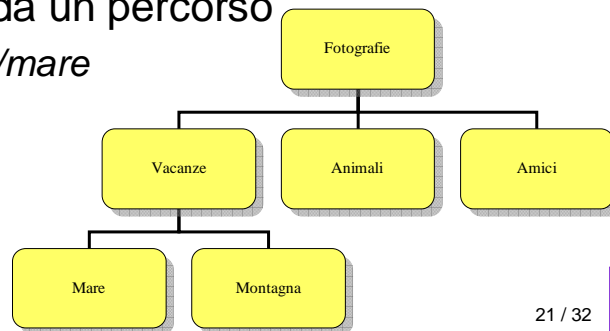
- Windows permette di gestire la memoria virtuale, anche se la “nasconde” all’utente usuale

Gestore della memoria secondaria (filesystem)

- I dati nella memoria secondaria sono organizzati in *file*
 - unità di aggregazione (importanza del contesto)
 - si misurano in *byte*
 - hanno un nome univoco (con limitazioni)
 - ulteriori proprietà: dimensione, tipo, date, permessi di lettura/scrittura *et c.*

Organizzazione dei file

- I *file* sono organizzati in cartelle
- Sono individuati da un percorso
fotografie/vacanze/mare



Tipi di file (estensione)

- Per individuare il tipo di *file* ci sono vari metodi
 - aggiungere una proprietà
 - riconoscerne la struttura
 - aggiungere un "simbolo" convenzionale (estensione) al nome

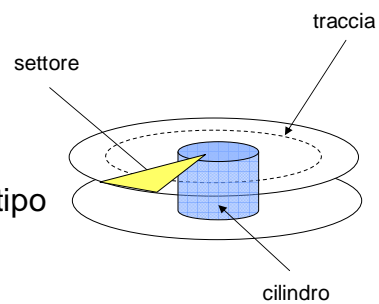
Estensione	Tipo
bmp, jpg, gif	immagine
exe, com	eseguibile
wav, mp3	suono
mpg, mov	video
...	...

Esempio

- In Windows “Risorse del computer” rappresenta una interfaccia per il *filesystem*

Tipi di filesystem e formattazione

- Il disco fisso viene formattato
 - fisicamente in tracce e settori
 - logicamente in partizioni di vario tipo
- Ci sono vari tipi di *filesystem*
 - Windows: FAT, NTFS
 - Linux: Ext2, Ext3, ReiserFS *et c.*
 - Mac OS X: HFS+
 - CD: ISO9660



Integrità e frammentazione

- I dischi fissi sono maggiormente soggetti ad usura
 - controlli di integrità sui *file*
- Le operazioni sui *file* tendono a disperderli in pezzettini sempre più piccoli
 - deframmentazione (Windows)

Disposizione iniziale (non frammentato)



Dopo diverso tempo (frammentato)



è necessario più tempo per leggerlo

25 / 32

Esempio

- In Windows ci sono due programmi appositi
 - ScanDisk si occupa del controllo dell'integrità
 - Defrag si occupa della "deframmentazione"

Gestione delle periferiche

- Nasconderne ed uniformarne il funzionamento
- Gestire gli errori (e.g. carta mancante) e gli accessi
- Esistono programmi specifici per ogni periferica, i *driver*

Coda di stampa

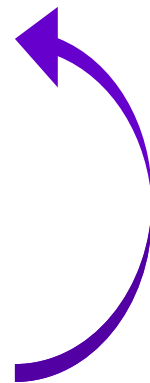
- La stampa è una delle periferiche più lente
- Lo *spooler*
 - costruisce una coda di richieste (di stampa)
 - invia i documenti alla periferica
 - permette ai programmi di continuare ed essere “avvertiti” quando il documento è pronto

Interprete dei comandi

- Ogni Sistema Operativo ha un interprete che riconosce le richieste degli utenti (comandi)
 - avvio di programmi
 - funzioni del Sistema Operativo
- Le interfacce a caratteri prevedono un linguaggio specifico

Terminale dei comandi

1. Pronto per ricevere un comando
`C: \>`
2. L'utente digita il comando
`C: \> ren *.jpg *.jpeg`
3. Il S.O. controlla la sintassi ed esegue



Sintassi di un comando

- Un comando è formato da
 - un nome (* indica qualsiasi serie di caratteri)
 - una serie di opzioni
 - una serie di argomenti

dir

nome

/p /od

opzioni

**.jpg*

argomento

Comandi più comuni

	Windows	Linux / Mac OS X
elenco dei <i>file</i>	dir	ls -la
creazione cartella	md	mkdir
eliminazione cartella	rd	rmdir
copia	copy	cp
spostamento	move	mv
eliminazione di <i>file</i>	del	rm
ridenominazione	ren	mv