



Terza Lezione

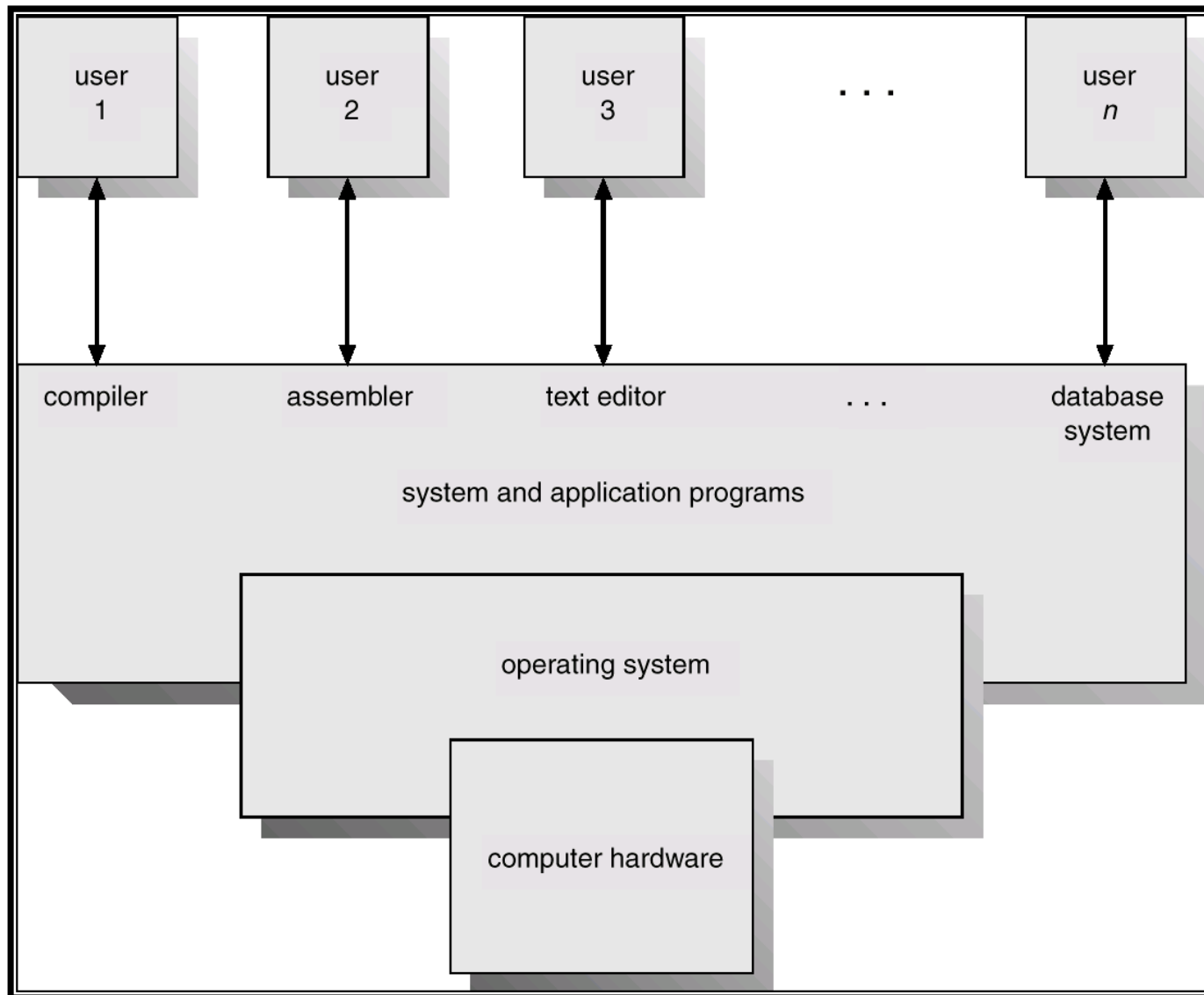
Il software



Il software

- Il software è **l'insieme dei programmi** che vengono eseguiti dal sistema.
- Un programma è un **insieme ordinato di istruzioni** sintatticamente corrette per l'elaboratore la cui esecuzione risolve un determinato problema.
- Tipi di software:
 - * **Sistemi Operativi** (Windows, Unix, Linux)
 - * **Il software Applicativo** (Word, Excel, Photoshop)

Componenti di un sistema





Il Sistema Operativo (SO)

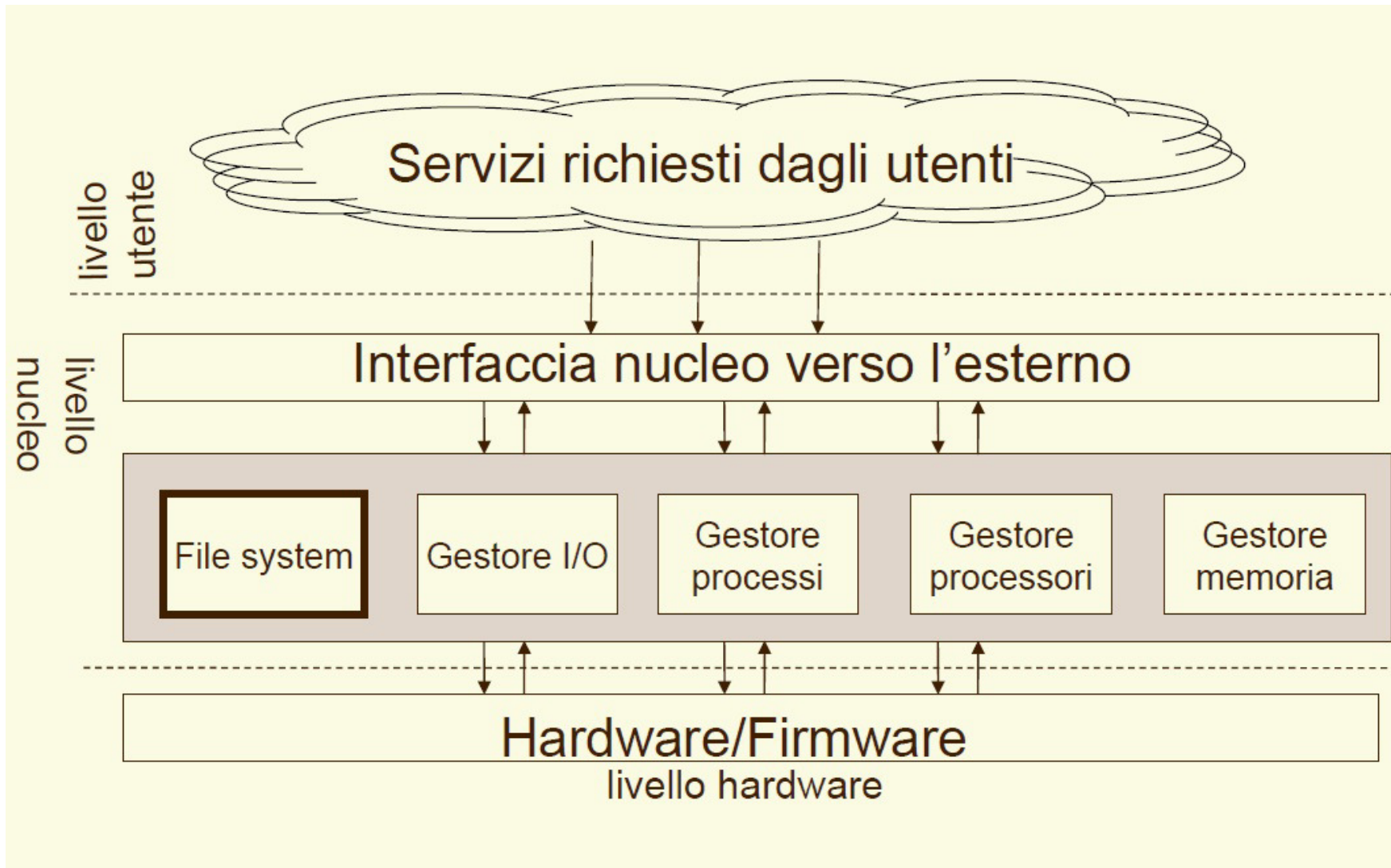
- È un **insieme di programmi** che gestisce il funzionamento di base del computer.
- Sono gestite dal sistema operativo **tutte le funzioni generali della macchina**, come l'aspetto grafico delle visualizzazioni su monitor, la scrittura e la lettura dai dischi, la messa in esecuzione e la chiusura dei vari programmi, la ricezione e trasmissione di dati attraverso tutti i dispositivi di I/O.
- Il sistema operativo **rimane sempre attivo** dal momento in cui viene caricato nella memoria centrale (all'accensione della macchina) fino allo spegnimento.



Componenti di un SO

- Gestore dei processi (programmi in esecuzione)
- Gestore della memoria principale
- Gestore della memoria secondaria
- Gestore delle Periferiche (dispositivi di I/O)
- Gestore dei file e del file system
- Gestore del sistemi di protezione
- Gestore del networking
- Sistema di interpretazione dei comandi (interfaccia utente)

Funzioni di un SO





Gestore dei processi

Un processo è un **programma in esecuzione**. Un processo necessita di certe risorse, tra cui tempo di CPU, memoria file, dispositivi di I/O, per portare a termine il suo compito.

Il SO è responsabile delle seguenti **attività** coinvolte nella gestione dei processi:

- **creazione e cancellazione** dei processi
- **sospensione e riesumazione** dei processi
- fornire **meccanismi** per
 - sincronizzazione dei processi
 - comunicazione tra processi
 - evitare, prevenire e risolvere i *deadlock*.



Gestore dei processi

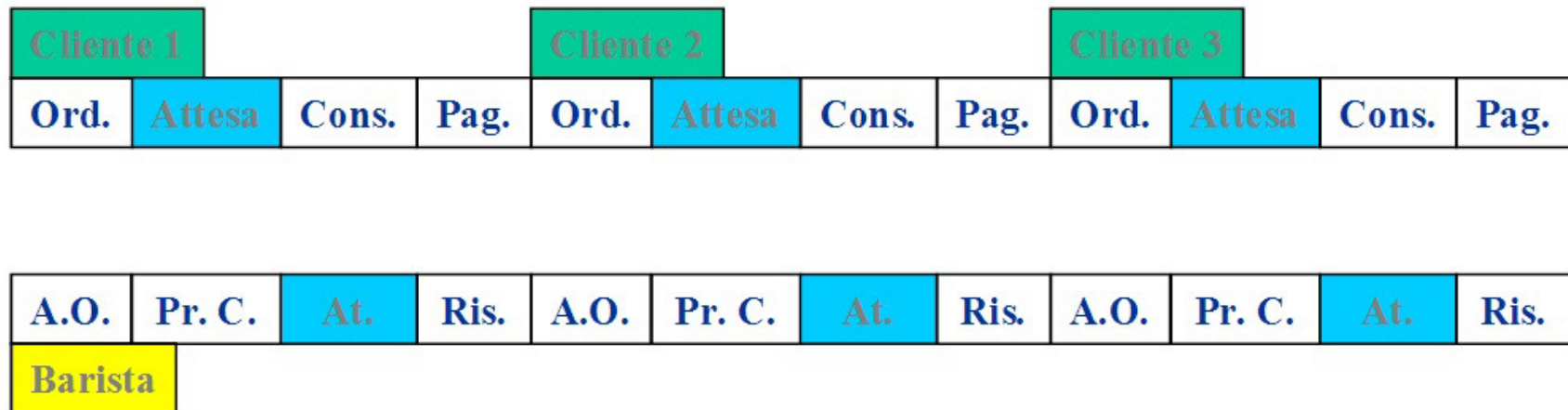
Uno dei compiti principali del sistema operativo riguarda la gestione dei processi

- Per il momento assumiamo di trattare sistemi monoprogrammati (anche detti **mono-tasking**)
- In sistemi di questo tipo è possibile eseguire un solo programma alla volta: i programmi devono essere eseguiti in modo sequenziale e si può mandare in esecuzione un programma solo quando quello precedente ha terminato l'esecuzione (es. MS/DOS)

Gestore dei processi

Supponiamo che il nostro sistema sia un bar in cui un barista serve diversi clienti

- Il barista è il corrispondente del processore, i clienti sono l'equivalente dei processi da eseguire
- Esecuzione mono-tasking:



A.O.=attesa ordine, Pr.C.=prepara cocktail, Ris.=risuoti, At.=attesa,
Pag.=pagamento, Ord.=ordina



Gestore dei processi

Quello che viene evidenziato dall'esempio precedente è vero in generale:

qualunque processo P alterna fasi di esecuzione a fasi in cui è bloccato in attesa di qualche evento esterno

Un processo può essere in attesa che sia terminata un'operazione di input di dati oppure in attesa di poter usare una risorsa in quel momento occupata



Gestore dei processi

La velocità di elaborazione di un calcolatore (quindi il tempo in cui un processo è in esecuzione) è dell'ordine del milionesimo di secondo, quindi molto minore dei tempi di lavoro delle periferiche di input/output, a volte dell'ordine di secondi, o addirittura i tempi di reazione umani

In un'esecuzione sequenziale, mentre il processo attivo è bloccato in attesa di eventi esterni, il processore rimane inattivo, in uno stato chiamato **idle**, e risulta pertanto sotto-utilizzato



Gestore dei processi

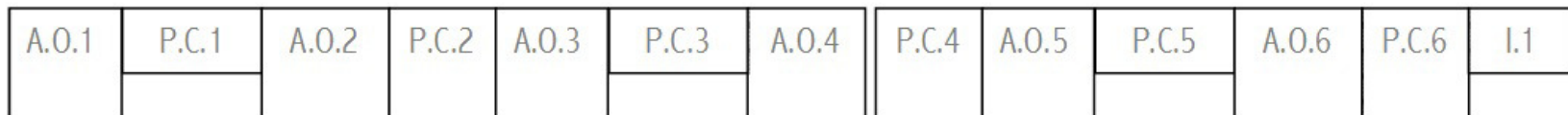
Una soluzione ragionevole per migliorare sia la “**produttività**” del barista che la qualità del servizio (i clienti arrivano e si mettono in coda aspettando di essere serviti) potrebbe essere quella di servire i più clienti ***contemporaneamente***

Ogni volta che un cliente inizia la consumazione del suo caffè, il barista, invece di restare inattivo può iniziare il servizio del prossimo cliente

Contemporaneamente non è esatto, infatti, poiché vi è un solo barista (*un solo processore* nel caso di un computer) il servizio ***non è realmente eseguito in parallelo*** ma avviene alternando il servizio tra i vari clienti.

In questo modo i tempi di inattività del barista sono eliminati

Gestore dei processi



Barista



Gestore dei processi

L'idea illustrata mediante l'esempio precedente è l'idea di base del *multi-tasking*: più programmi vengono eseguiti contemporaneamente sullo stesso processore.

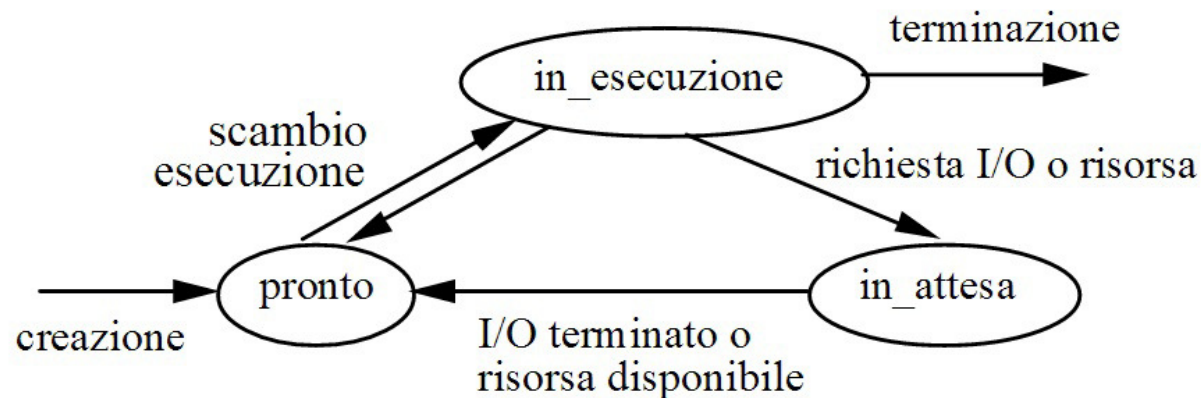
Il numero di processi attivi viene detto *grado di multiprogrammazione* del sistema

- Dal punto di vista del **processore**, in ogni istante vi è un solo processo in esecuzione
- Se l'alternanza tra i processi è frequente (ad esempio ogni 10 millisecondi), **l'utente** ha l'impressione che l'esecuzione dei programmi sia veramente simultanea

Gestore dei processi

Un processo quindi può trovarsi in tre diversi stati:

- *in esecuzione*: quando sta utilizzando il processore;
- *in attesa*: (bloccato), quando è in attesa del verificarsi di un evento esterno (terminazione di un'operazione di input/output o altro);
- *pronto*: quando è potenzialmente in condizione di poter utilizzare il processore che è occupato da un altro processo.





Gestore dei processi

Il *multi-tasking* funziona efficacemente se il S.O. è in grado di offrire alcune assicurazioni:

- Il *cambio di contesto* deve essere invisibile al processo. Nell'esempio del bar i clienti non si accorgono che il barista serve un gruppo di clienti contemporaneamente
- Quando *il barista riprende a servire un cliente*, il servizio viene ripreso esattamente nel punto in cui era stato interrotto
- Il barista ha un *cronometro* e ogni *n* secondi *sospende* il servizio che sta effettuando e passa a servire un altro cliente



Gestore dei processi

- La politica di gestione in cui il barista divide il suo tempo tra i vari clienti assegnando a ciascuno un certo intervallo di tempo è implementata nei S.O. multi-tasking e prende il nome di ***Round-Robin***
- La politica di gestione in cui il barista serve in modo sequenziale i vari clienti viene chiamata ***FIFO***



Gestore di memoria principale

La memoria principale è un grande **insieme di parole**, ognuna identificata univocamente con un **indirizzo**, e può considerarsi come un deposito di dati velocemente accessibili da **CPU** e **dispositivi di I/O**. Tale memoria è volatile e perde i dati in caso di ***system failure***.

Il SO è responsabile delle seguenti attività:

- **tenere traccia** delle aree di memoria correntemente utilizzate e dei rispettivi utenti
- **decidere** quale processo caricare in memoria quando dello spazio si rende disponibile
- **allocare o liberare** (deallocare) spazio in memoria, a seconda delle richieste.



Gestore di memoria secondaria

Poiché la memoria principale è volatile e troppo piccola per contenere tutti dati ed i programmi permanentemente, il calcolatore deve prevedere anche una memoria secondaria, di supporto a quella principale. La maggior parte dei calcolatori utilizza **dischi** come supporti di memoria secondaria, per memorizzare sia dati che programmi.

Il SO è responsabile delle seguenti attività per gestire i dischi:

- **gestione dello spazio libero**
- **allocazione dello spazio**
- **schedulazione dei dischi.**



Gestore del sistema di I/O

Il SO gestisce le operazioni sulle periferiche di I/O tramite:

- un sistema di **memoria** a buffer (buffer: area di memoria usata per conservare temporaneamente i dati da trasferire ad un dispositivo)
- **l'interfaccia** per il gestore del dispositivo di I/O (interface driver)
- i **driver** stessi per ciascun dispositivo di I/O (controller).



Gestore dei files

Un file è inteso come una **collezione di informazioni correlate**, cioè un insieme di byte che codificano una certa entità logica (testo, immagine, suono, programma, etc), organizzati secondo un certo formato, memorizzati su supporti di memoria secondaria.

Comunemente rappresentano programmi e dati.

Il SO è responsabile di:

- creazione e cancellazione di **files**
- creazione e cancellazione di **directory**
- supporto di **primitive** (semplici funzioni o programmi) per la gestione di files e directory
- allocazione di files in **memoria secondaria**
- salvataggio di dati su **supporti non volatili**.



File System

Il File System è quella parte del S.O. che si occupa di **gestire e strutturare** le informazioni memorizzate su supporti permanenti.

Deve **fornire una visione astratta** dei file su disco e l'utente deve avere la possibilità di:

- **identificare ogni file con un nome** (filename) astraendo completamente dalla sua memorizzazione fisica (blocchi su disco rigido e localizzazione dei blocchi)
- avere un **insieme di operazioni** per lavorare sui file: creare o rimuovere un file, copiarlo, cambiargli nome, inserire informazioni in un file

continua ...



File System

- **effettuare l'accesso alle informazioni** mediante operazioni ad alto livello, che non tengono conto del tipo di memorizzazione (accedere ad un file memorizzato sul disco rigido oppure su un CD-ROM allo stesso modo)
- avere la possibilità di **strutturare un insieme di file**, organizzandoli in sottoinsiemi secondo le loro caratteristiche, per avere una visione ordinata e strutturata delle informazioni sul disco
- in un sistema multi-utente, inoltre l'utente deve avere dei **meccanismi per proteggere i propri file**, ossia per impedire ad altri di leggerli, scriverli o cancellarli



File System

Un insieme di operazioni minimale, presente in tutti i sistemi, è il seguente:

- **creazione di un file**
- **cancellazione** (rimozione) di un file
- **copia** di un file
- **visualizzazione** del contenuto di un file ad esempio sul video del terminale o in una finestra sul video
- **stampa** di un file
- **lettura e modifica** del contenuto di un file
- **ridenominazione** di un file, per cambiare il nome ad un file
- **visualizzazione delle caratteristiche** di un insieme di file (dimensione, data di creazione e dell'ultima modifica, vincoli di protezione, ecc.)



Organizzazione gerarchica

Il numero di file che devono essere memorizzati su un disco può essere estremamente elevato

- si ha quindi la necessità di **mantenere i file** in una **forma ordinata**
- **un unico spazio** (contenitore) di file è **scomodo** perché le operazioni di ricerca di un file e di creazione di un nuovo file diventano onerose (non è possibile avere due file con lo stesso nome)
- l'idea è quella di **raggruppare i file** in sottoinsiemi (seguendo magari dei criteri di tipo logico)

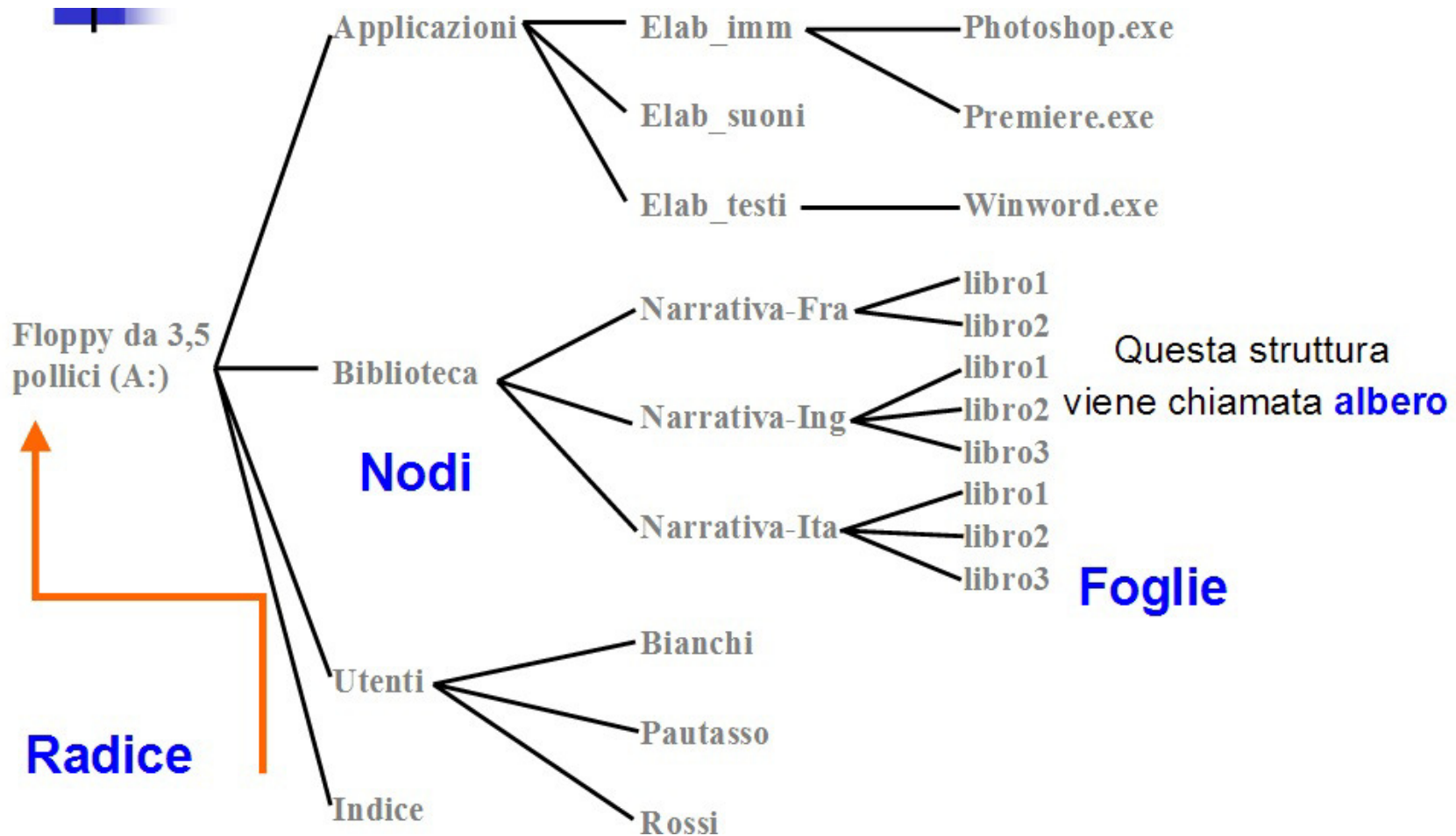


Organizzazione gerarchica

Tutti i sistemi operativi forniscono operazioni per creare delle **directory** (cartelle)

- una directory (cartella) è costituita da un **insieme di file**
- i nomi dei file sono **locali** alle directory (si possono avere due file con lo stesso nome purché siano in due directory diverse)
- in questo modo **l'indice conterrà due tipi di oggetti:**
nomi di file e nomi di directory
- senza la strutturazione in directory, tutti i file potrebbero essere **identificati** semplicemente **mediante il loro nome**
- nel caso di un'organizzazione gerarchica a più livelli si deve specificare **l'intera sequenza di directory** che lo contengono, a partire dalla radice dell'albero

Organizzazione gerarchica





Organizzazione gerarchica

Ad esempio il file **libro1** di narrativa italiana è univocamente identificato dalla sequenza:

A:\Biblioteca\Narrativa-Ita\libro1

La directory **Pautasso** di **Utenti** è identificata dalla sequenza:

A:\Utenti\Pautasso

il carattere “\” (**slash**) viene usato come separatore

- una sequenza di questo tipo può essere vista come il cammino che si deve compiere per raggiungere un determinato file a partire dalla **radice dell'albero**, ed è chiamata **pathname assoluto**

- il **pathname relativo** è invece il percorso che si deve compiere per raggiungere il file a partire dalla directory nella quale siamo (directory corrente)



Organizzazione gerarchica

Per organizzare gerarchicamente i file, il sistema operativo deve fornire all'utente anche un insieme di operazioni sulle directory, per esempio:

- creare una nuova directory
- rimuovere una directory
- ridenominare una directory
- elencare il contenuto (l'insieme di file e sotto-directory) di una directory
- copiare o spostare i file da una directory ad un'altra



Gestore della protezione

Per protezione si intende un meccanismo per la gestione degli accessi al sistema ed alle risorse degli utenti da parte di programmi, processi o altri utenti.

Il SO deve:

- **distinguere** tra uso autorizzato e non autorizzato
- fornire un modo per **specificare i controlli** da imporre
- **forzare utenti e processi** a sottostare ai controlli richiesti.



Networking (sistemi distribuiti)

Un sistema distribuito è una collezione di processori che non condividono memoria e clock. Tali processori sono connessi tramite una rete di comunicazione.

Il SO deve gestire un sistema distribuito per garantire agli utenti l'accesso alle risorse del sistema. Tale gestione deve proporsi come fine:

- l'aumento delle **prestazioni computazionali**
- l'incremento della quantità di **dati accessibili**
- l'aumento dell'**affidabilità** del sistema.



Interprete dei comandi

Molte istruzioni sono date al SO tramite *control statements*, (letteralmente: dichiarazioni di istruzioni) che servono per creare e gestire processi, gestire gli I/O, gestire la memoria principale e secondaria, gestire il file system ...

Il programma del SO che legge ed interpreta tali istruzioni viene chiamato interprete della linea di comando (sistemi DOS e Windows) o shell (UNIX) o infine interfaccia grafica (MacOS).

La sua funzione è di **ricevere** un comando ed **eseguirlo**.



Servizi offerti dal SO

- **Esecuzione di programmi** – capacità di caricare un programma nella memoria ed eseguirlo.
- **Operazioni I/O** – strumenti per l'uso delle periferiche (lettura disco, scrittura video, ...).
- **Gestione del File-system manipulation** – capacità di leggere, scrivere, creare e cancellare i files.
- **Comunicazione** – scambio di informazioni fra i processi eseguiti sullo stesso computer o su computer diversi.
- **Determinazione degli errori** – assicurare una corretta computazione determinando gli errori nella CPU, nella memoria, nelle periferiche o nei programmi.



Servizi Addizionali

- **Allocazione delle risorse** – allocazione delle risorse del sistema fra i vari utenti e i vari programmi che usano contemporaneamente il sistema.
- **Contabilità** – mantenimento delle informazioni relative all'uso delle risorse eseguite dagli utenti sul sistema. Scopo di tale operazione è: uso per statistiche, o protezione, o pagamento dell'uso del sistema.
- **Protezione** – assicurazione che tutti gli accessi alle risorse del sistema siano controllate.



Sistemi Operativi

- Il sistema operativo risiede sull'hard disk come tutti gli altri programmi e viene caricato nella memoria RAM all'accensione della macchina.
- Esistono molti sistemi operativi:
 - **MS-DOS**, oggi soppiantato da Windows.
 - **Windows** della Microsoft
 - **MacOS** della Apple per i computer Macintosh,
 - **Linux** e **Unix**



SO con linea di comando

- MS-DOS, Linux e Unix sono sistemi operativi in cui non compare nessuna grafica e tutti i comandi devono essere digitati da tastiera.

```
C:\TEMP>dir

Il volume nell'unità C non ha etichetta
Numero di serie del volume: 226D-1E07
Directory di C:\TEMP

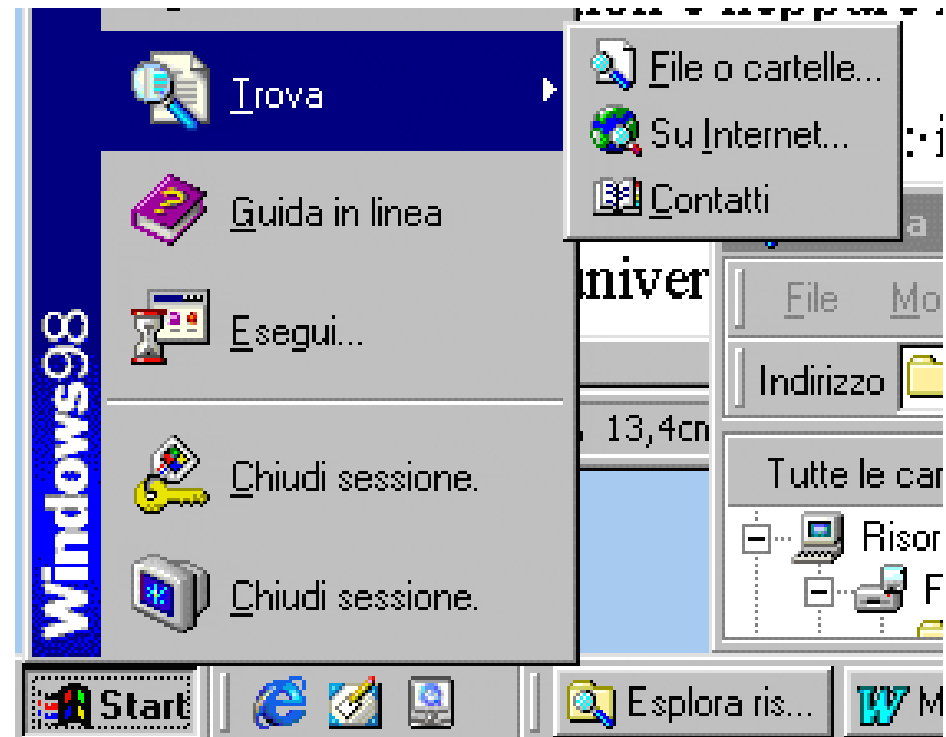
.           <DIR>           21/12/98   23.32
..          <DIR>           21/12/98   23.32
_.$SETUP.DLL  11.264  12/05/97   17.32
    1 file          11.264 byte
    2 dir           638.816.256 byte disponibili

C:\TEMP>
```

SO con Interfaccia Grafica

- Windows e MacOS usano una interfaccia grafica (Graphical User Interface - **GUI**).

Tutte le operazioni si svolgono tramite icone e finestre, usando il mouse per lanciare programmi.





I drivers

- Sono dei file usati dal sistema operativo per effettuare la comunicazione fra il computer e le varie periferiche.
- Per poter funzionare correttamente, ogni dispositivo deve avere il suo particolare driver registrato dal sistema operativo. Esistono perciò i driver di stampante, modem, lettore CD, scheda video, scheda audio, scheda di rete, ...
- I driver vengono installati sul computer attraverso un software di installazione automatica che viene fornito assieme ai dispositivi stessi.
- Windows possiede una funzione detta **Plug and Play** che all'accensione del computer verifica la presenza di nuovi componenti hardware, ricercando automaticamente il driver adatto.



Il software Applicativo

- È l'insieme dei programmi che non sono compresi nel sistema operativo, e che vengono invece installati dall'utente per svolgere compiti specifici, come:
 - Elaborare un testo (Microsoft Word, Latex)
 - Elaborare dati (Microsoft Excel, Lotus 123)
 - Archiviare dati (Microsoft Access, Oracle)
 - Grafica (Adobe Photoshop, Corel Draw)
 - Posta elettronica (Outlook, Eudora)



Il software Applicativo

- **Utilità di Sistema** - programmi che servono per migliorare la gestione e la sicurezza della macchina (**antivirus**, **compressione file**, **programmi diagnostici**, **programmi di backup**, **di installazione**, **ottimizzazione**, ...)
- **Strumenti di sviluppo** - programmi che servono per la creazione di nuovi applicativi (compilatori di linguaggi di programmazione, programmi per la creazione di oggetti multimediali, elaborazione di audio, video e immagini, ...)



Software proprietario

- Occorre acquistarlo per poterlo usare. Il costo varia da poche decine di Euro fino ad alcune migliaia, in funzione della complessità del prodotto.
- È detto anche **software non libero**. Il suo utilizzo, la redistribuzione o modifica sono proibiti o richiedono un permesso.
- Tipicamente nel CD di installazione è sempre compresa anche la **licenza d'uso** personale che consente di utilizzare liberamente quel software per la propria attività (i termini esatti della licenza sono sempre inclusi nella documentazione allegata al prodotto).



Software shareware

- È dato in prova gratuita per un periodo di tempo limitato (es. 30 giorni) a partire dal momento dell'installazione. Una volta scaduti i termini, se l'utente è soddisfatto e vuole continuare ad usare il prodotto, deve acquistarlo.
- I programmi shareware alla scadenza possono:
 - avvisare l'utente del mancato pagamento ma continuano a funzionare.
 - smettere di funzionare o funzionare in modalità ridotta.
- I programmi shareware riacquistano la completa funzionalità solo con l'inserimento di una **chiave software** che il produttore fornisce dopo il pagamento.



Software Pirata

- La copiatura del software è consentita unicamente come copia di riserva per l'utente che lo ha acquistato.
- Non è consentito invece né copiare i CD per cederli ad altri (a nessun titolo), né usare gli stessi CD originali per effettuare più installazioni su diverse macchine (a meno che la licenza non lo preveda espressamente); si parla in questi casi di **software pirata**.



Software freeware

- Il software freeware è gratuito e può essere copiato e usato liberamente (eccetto rivenderlo ad altri).
- Si tratta di solito di programmi relativamente semplici, messi a disposizione del pubblico per motivi pubblicitari (di solito sono versioni semplificate di prodotti a pagamento, oppure versioni di lancio per nuovi software).
- A volte l'uso gratuito viene vincolato ad impieghi esclusivamente non commerciali.



Free Software

- La **Free Software Foundation** (FSF) si occupa di eliminare le restrizioni sulla copia, sulla redistribuzione, sulla comprensione e sulla modifica dei programmi per computer. Opera promovendo lo sviluppo e l'uso del **software libero** in tutte le aree dell'informatica
- Il **software libero** è software distribuito in modo che chiunque ne abbia il permesso di **uso, copia e distribuzione, in forma modificata o meno, gratis o a pagamento**. In particolare, ciò significa che il codice sorgente deve essere disponibile.



Free Software

- L'espressione "**software libero**" si riferisce alla libertà dell'utente di:
 - Libertà 0:** eseguire il programma, per qualsiasi scopo
 - Libertà 1:** di studiare come funziona il programma e adattarlo alle proprie necessità. L'accesso al codice sorgente ne è un prerequisito.
 - Libertà 2:** ridistribuire copie in modo da aiutare il prossimo.
 - Libertà 3:** di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito.
- Un programma è software libero se l'utente ha tutte queste libertà.
- Una descrizione completa è disponibile su:
<http://www.gnu.org/philosophy/free-sw.it.html>