

LINUX

- Una versione completa e affidabile di UNIX
- Disponibile per PC x86 Intel/AMD e numerose altre piattaforme
- Strumento (quasi) indispensabile per le esercitazioni
- Include gli strumenti di sviluppo necessari
 - compilatore C (gcc)
 - editor (vi , emacs, xemacs)
 - debugger (gdb, ddd)
 - manuali on-line (comandi e primitive)

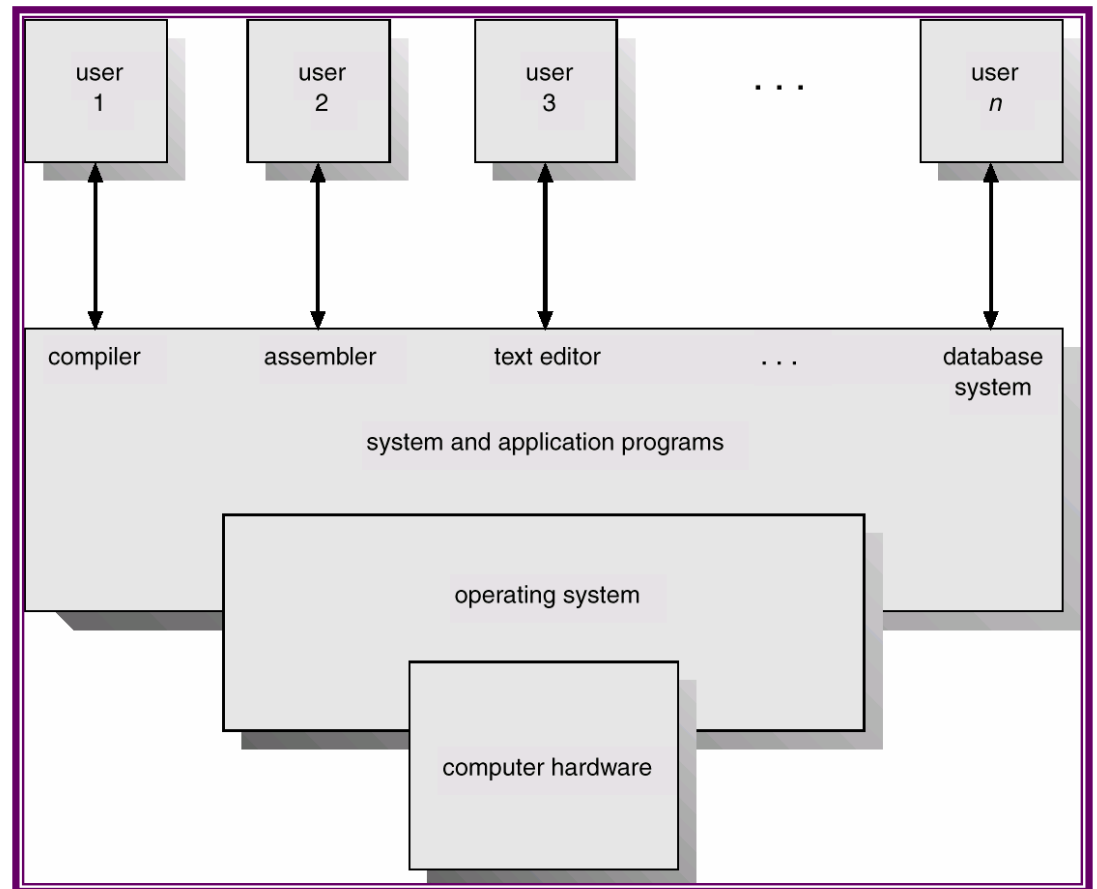
LINUX

- Numerose distribuzioni
(RedHat, Mandrake, Suse, Debian, ...)
 - può coesistere con Windows (dual-boot)
 - partizioni separate o comuni
 - possibile ridimensionamento di una partizione Windows esistente (fips)
- Numerose guide alla installazione
- Cercate con google "guida installazione Linux" – ad es.
<http://linux.html.it/guida>

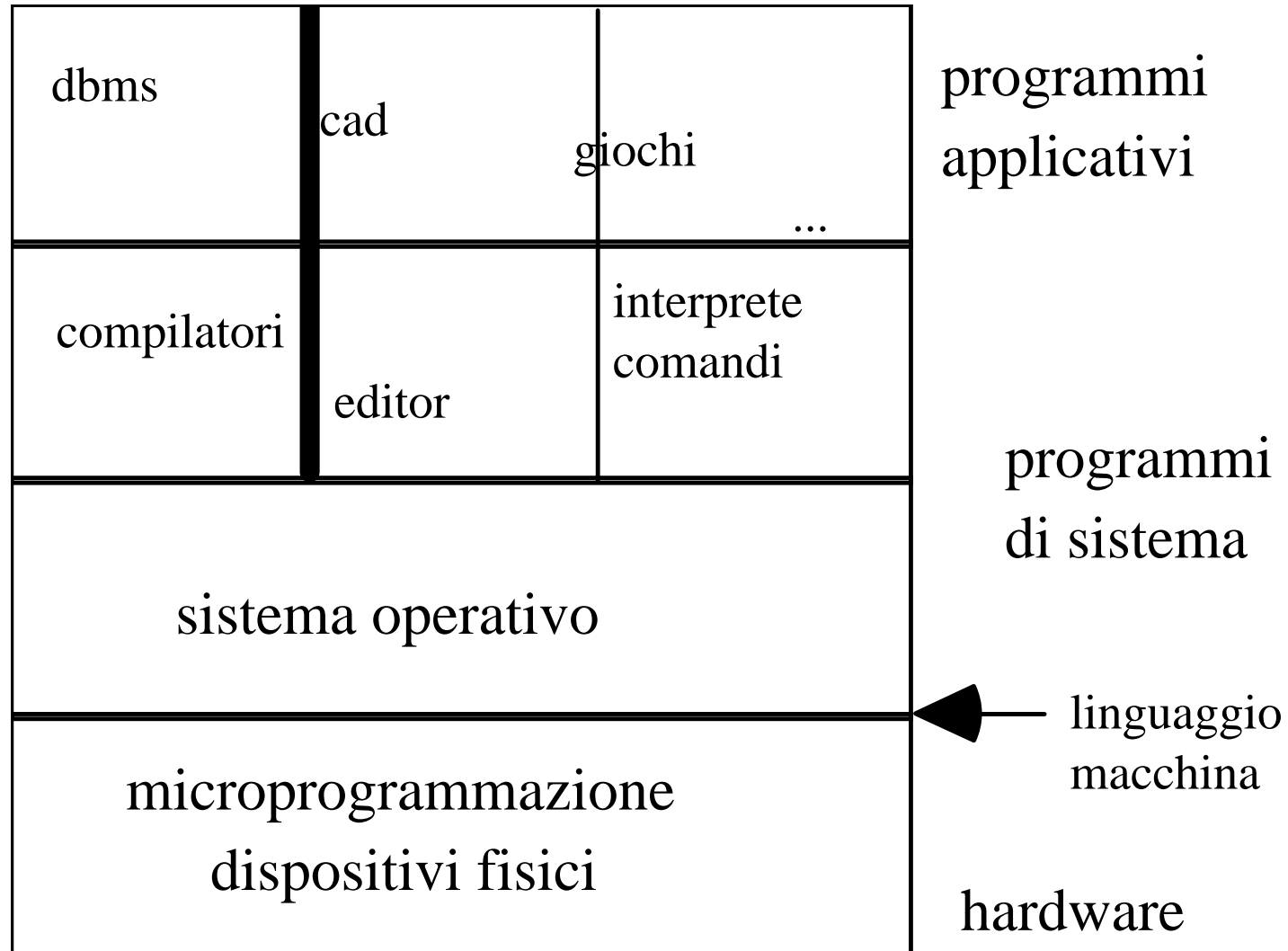
Introduzione

Un *sistema di elaborazione* può essere visto come l'insieme di:

- hardware
- sistema operativo
- programmi applicativi
- utenti



Sistema di elaborazione



Che cos'è un sistema operativo?

- Un sistema di calcolo può essere visto come un insieme di risorse Hw e Sw utilizzate per lo sviluppo e la esecuzione dei programmi utente
- Tali risorse devono essere:
 - utilizzate secondo *un determinato ordine*
 - rese disponibili *a più utenti*
 - *protette* contro accessi non autorizzati
 - organizzate in modo da garantire la *sopravvivenza del sistema* in caso di guasti
 - gestite in modo che risulti *semplificato ed efficiente il loro uso*, etc.

- Con il termine **sistema operativo** si intende quell'insieme di programmi che provvedono alla gestione delle risorse Hw e Sw di un sistema di calcolo
- Una definizione alternativa (Tanenbaum):
un sistema operativo e' un programma che *controlla le risorse* di un calcolatore e fornisce ai suoi utenti un'interfaccia o *macchina virtuale piu' agevole* da utilizzare della macchina "nuda".

Sistema Operativo (1)

Puo` essere visto come:

- allocatore di risorse Hw e Sw:
 - tempo di CPU, spazio di memoria, dispositivi di I/O, compilatori, etc.

Le risorse devono essere assegnate a programmi specifici secondo determinate politiche

Sistema Operativo (2)

Puo` essere visto come:

- programma di controllo:

- controlla l'esecuzione dei programmi per prevenire errori ed usi impropri del calcolatore (in particolare per il controllo dei dispositivi di I/O)

Obiettivi del S.O

- rendere *piu` semplice* l'uso di un sistema di calcolo
- rendere *piu` efficiente* l'uso delle risorse del sistema di calcolo

Sistema Operativo

Il **sistema operativo** e` costituito dall'insieme dei programmi (software o firmware) che *rendono praticamente utilizzabile* l'elaboratore agli utenti cercando contemporaneamente di *ottimizzarne le prestazioni*.

- Visione top-down: il sistema operativo come una *macchina estesa* (fornisce astrazione, hiding di dettagli)
- Visione bottom-up: il sistema operativo come un *gestore di risorse* (fornisce protezione, risoluzione di conflitti o interferenze)

Risorse hardware

- processori (registri, unità aritmetiche, parallelismo interno)
- memorie
- canali di comunicazione
- dispositivi di I/O

Grande evoluzione sia sui singoli componenti che sulle tecniche di collegamento

Spostamento di intelligenza verso i dispositivi

Risorse hardware (2)

- Gerarchia di memoria
 - memoria centrale (principale, core)
 - memoria cache
 - memoria secondaria

Gestione delle risorse

Significa:

- tenere traccia delle risorse
- adottare strategie di assegnazione
- allocare le risorse
- recuperare le risorse inutilizzate
- rilevare eventuali usi impropri

Gestione delle risorse (2)

Funzioni specifiche:

- gestione della memoria principale
- gestione dei processori
- gestione dei dispositivi periferici
- gestione della memoria secondaria

Funzioni specifiche di gestione

Gestione dei dispositivi periferici

- mascherare al programmatore la complessità delle operazioni di I/O
- effettuare controlli sul corretto funzionamento delle operazioni
- risolvere conflitti nell'utilizzo di una stessa periferica da parte di più programmi
- consentire il massimo sfruttamento delle periferiche.

Funzioni specifiche di gestione (2)

Gestione dei processori

- decidere quale programma userà il processore (scheduling) in base a criteri di corretto funzionamento e di efficienza
- verificare che i programmi rilascino il processore entro il tempo stabilito.

Funzioni specifiche di gestione (3)

Gestione della memoria centrale

- caricare in memoria programmi e dati
- evitare interferenze fra programmi diversi
- assegnare la memoria in base a criteri di efficienza
- minimizzare i trasferimenti tra memoria centrale e memoria di massa.

Funzioni specifiche di gestione (4)

Gestione della memoria secondaria

- consentire l'accesso all'informazione in base alla sua organizzazione logica (**File System**) anzichè fisica (ad es. dischi, tracce, settori)
- controllare i diritti di accesso ai file da parte degli utenti
- consentire creazione, modifica e cancellazione dei file, ...

Proprietà fondamentali di un S.O.

- affidabilità

- efficienza

Aree di applicazione di un S.O.

- sistemi di tipo generale
- sistemi in tempo reale
 - applicazioni per il controllo di processo
 - applicazioni di tipo gestionale (interrogazione di basi di dati)

Funzioni di un S.O.

- definizione e gestione dell'interfaccia utente
- gestione dei lavori degli utenti
- gestione delle risorse del sistema
- ausilii per la messa a punto dei programmi
- ausilii per la gestione dei dati -- file system
- funzioni ausiliarie di sistema per
 - affidabilità
 - sicurezza
 - contabilità

Utenti del S.O.

- *utenti finali* del sistema
per essi il sistema operativo è *trasparente*
- *programmatori applicativi*
utilizzano i servizi del S.O. per la realizzazione e l'esecuzione dei loro programmi

Utenti del S.O. (2)

- *programmatori di sistema*

aggiornano e modificano i programmi del S.O. per adeguarli a nuove necessità del sistema o degli utenti applicativi

- *operatori*

controllano il funzionamento e rispondono alle richieste di intervento da parte del sistema

Utenti del S.O. (3)

- *amministratore del sistema*

stabilisce le politiche di gestione del sistema e ne cura l'osservanza



Riassumendo:

- utenti finali
- programmatori applicativi
- programmatori di sistema
- operatori
- amministratore del sistema

Tipi di S.O. (1)

Sistemi proprietari

- progettati dai costruttori al fine di *sfruttare in modo ottimale* le risorse di ciascun tipo di macchina
- l'interfaccia con l'utente varia tra le diverse famiglie di sistemi
- esempi:
 - IBM: OS/360 - 370, VM, MVS
 - DEC: RT-11, VMS

Tipi di S.O. (2)

Sistemi standard

- progettati da case di software o da grandi utenti per *creare applicazioni portabili* su sistemi diversi
- l'interfaccia con l'utente rimane costante nelle diverse versioni
- esempi:
 - UNIX, MS-DOS, Windows

Evoluzione nell'uso dei calcolatori (1)

- Scrivere programmi che realizzano algoritmi:
 - strutture dati \Rightarrow *transienti*
 - libreria di sottoprogrammi \Rightarrow *capitale*
- Evoluzione verso applicazioni in cui i dati rappresentano lo stato del sistema che evolve
 - strutture dati \Rightarrow capitale

Le strutture dati sopravvivono al programma

Evoluzione nell'uso dei calcolatori (2)

- Embedded applications (sistemi bancari, banche dati, controllo di processo)
- Evoluzione verso applicazioni di A.I.:
 - riconoscimento del linguaggio naturale
 - basi di conoscenza, sistemi esperti
 - robotica
 - visione

Il sistema operativo deve evolvere di conseguenza