

2. Cenni di sistemi operativi

Andrea Marongiu

(andrea.marongiu@unimore.it)

Paolo Valente

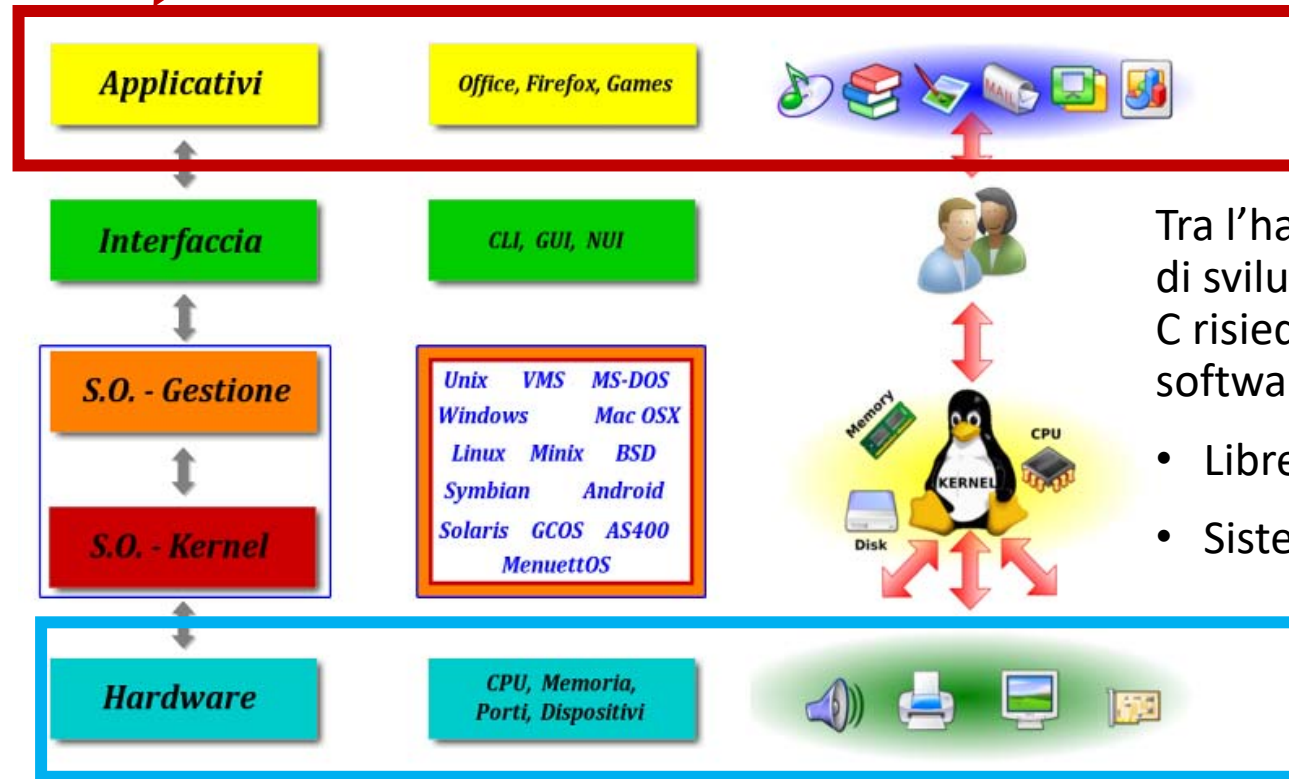
Contiene slides dal corso «Sistemi Operativi» dei prof.
Gherardi/Scandurra dell'Università degli studi di Bergamo

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA



Lo stack software di un calcolatore moderno

A questo livello lavoriamo noi di Programmazione I



Tra l'hardware e l'ambiente di sviluppo degli applicativi C risiede un sistema software stratificato

- Librerie
- Sistema operativo

A questo livello l'astrazione è quella dell' ISA

Programma (recall)

- Per far eseguire un programma ad un processore, basta
 - memorizzare da qualche parte nella memoria la sequenza di configurazioni di bit relativa alle istruzioni da eseguire
 - dire al processore a che indirizzo si trova la prima di tali istruzioni
- Il processore eseguirà, una dopo l'altra, le istruzioni che trova a partire da tale indirizzo
 - Finché non si esegue un'istruzione di salto

Avvio del sistema operativo

- Il primo programma eseguito all'accensione del calcolatore è il sistema operativo
- Il sistema operativo dice al processore dove si trova la prima istruzione di ogni programma di cui l'utente chiede l'esecuzione
- Ma chi dice al processore dove si trova la prima istruzione del sistema operativo?

Avvio del sistema operativo

- Il calcolatore (PC, smartphone, ...) è configurato per caricare in memoria, all'avvio, il contenuto di un blocco predefinito
- **Master Boot Record (MBR)**
 - Ospitato in uno dei dispositivi di memorizzazione di massa disponibili
 - Configurabile (possono esistere più sistemi operativi installati sul disco, e si può decidere una specifica sequenza di avvio)

Avvio del sistema operativo

- Il codice presente nel MBR viene tipicamente chiamato **bootloader**
- Il suo compito è quello di caricare il sistema operativo installato nella macchina
- Consiste in una sequenza di byte, che si assume quindi essere un programma in linguaggio macchina

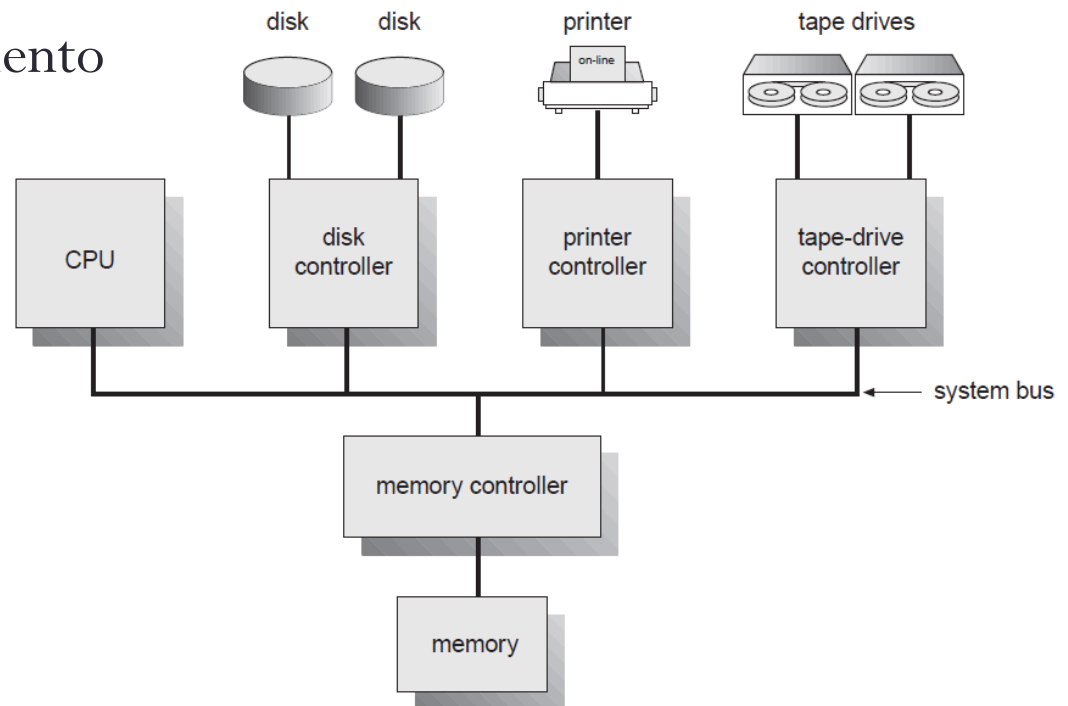
OBIETTIVI E FUNZIONI DI UN SO

Obiettivi di un sistema operativo (1)

- Astrazione

Alzare il livello di astrazione dei componenti del sistema di elaborazione (macchina di von Neumann)

- Astrazione del comportamento
- Semplificazione dell'uso
- Più facile da programmare dell'hardware sottostante
- Semplificazione nella implementazione dei programmi applicativi



Obiettivi di un sistema operativo (2)

- Virtualizzazione

Creazione di un'immagine del sistema di elaborazione dedicata a ciascun programma in esecuzione

- Indipendenza dalla presenza di altri programmi
- Gestione ottimizzata delle risorse
- Allocazione e condivisione ordinata delle risorse rispetto a *spazio / tempo*

Funzione principale di un SO

- Un SO fornisce l'ambiente per l'esecuzione dei programmi

Più in dettaglio

Funzioni (1)

- Gestione del processore
- **Gestione dei processi: un processo è un programma in esecuzione**
- Creazione e terminazione dei processi
- Sospensione e riattivazione dei processi
- Schedulazione dei processi
- Sincronizzazione tra processi
- Gestione di situazioni di stallo (deadlock)
- Comunicazioni tra processi (memoria condivisa o scambio di messaggi)

Funzioni (2)

- Gestione della memoria centrale
- **Processi in memoria centrale per esecuzione**
- Multiprogrammazione
- Allocazione e deallocazione della memoria ai processi
- Caricamento e scaricamento di processi e di loro porzioni in memoria centrale
- Protezione della memoria centrale

Funzioni (3)

- Gestione delle periferiche
- Omogeneità di interazione (verso hardware eterogeneo)
- Configurazione e inizializzazione
- Interfaccia generale e omogenea
- Gestione ottimizzata dei dispositivi di I/O, memorizzazione di massa e rete informatica
- Protezione delle periferiche
- Bufferizzazione
- Caching

Funzioni (4)

- Gestione del file system
- File e albero del file system
- Nasconde all'utente l'organizzazione della memoria attraverso il concetto di **file**
- File e directory
 - Creazione e cancellazione
 - Lettura e scrittura
 - Copiatura
 - Ricerca
 - Salvataggio e ripristino
 - Protezione e sicurezza (diritti di accesso)

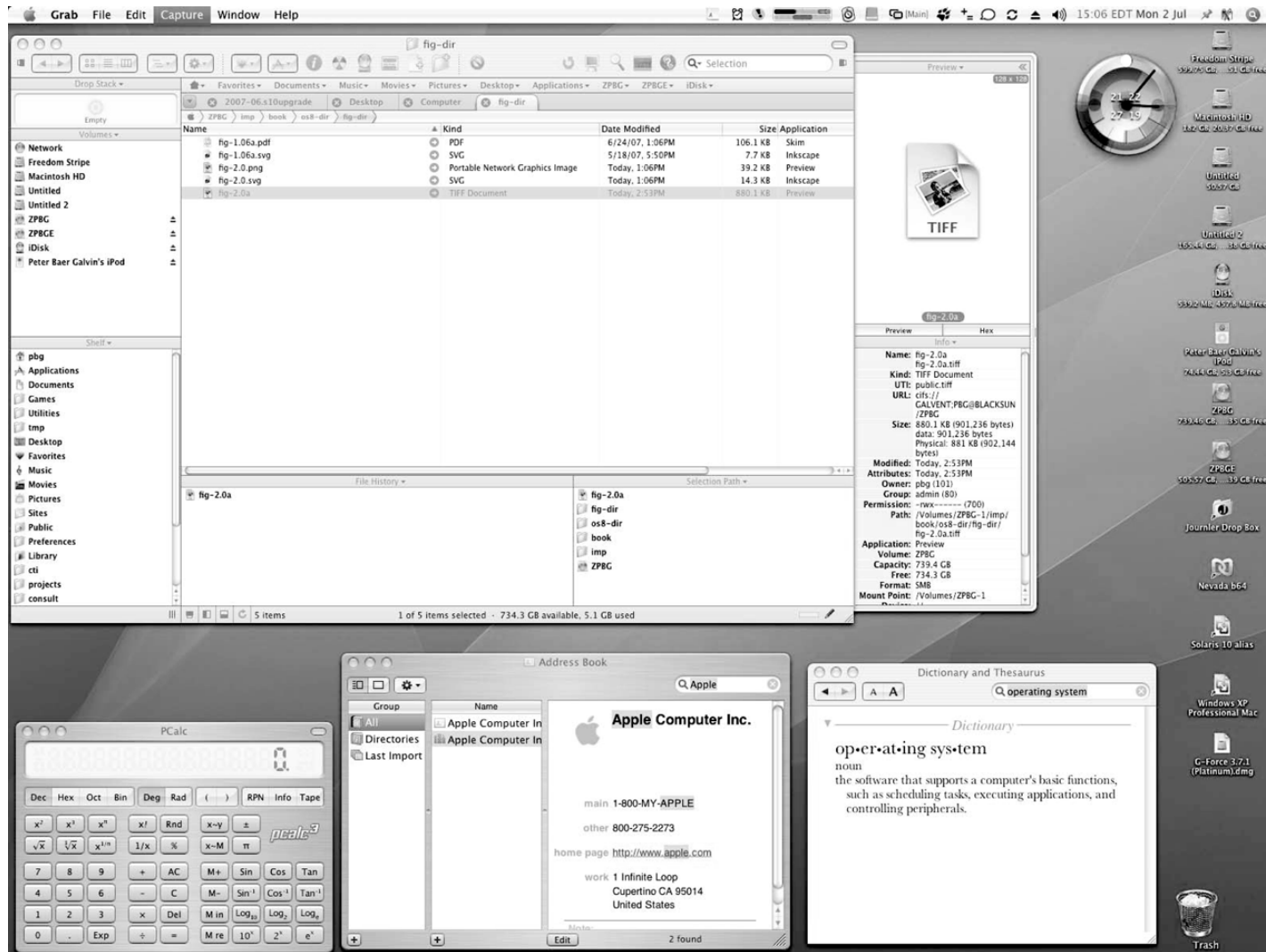
Funzioni (5)

- Gestione dell'interfaccia utente
- **Interazione con utenti e processi attraverso istruzioni di controllo**
- Interfacce a riga di comando: interprete dei comandi o shell (programma che legge ed interpreta istruzioni di controllo)
- Interfacce grafiche
- Interfacce a lotti
 - comandi e direttive codificati in file ed eseguiti a lotti
- L'interfaccia è separata dall'esecuzione dei comandi (flessibilità)!!!

Funzioni (5)

```
Terminal
File Edit View Terminal Tabs Help
fd0      0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0    0
sd0      0.0    0.2    0.0    0.2    0.0    0.0    0.4    0    0
sd1      0.0    0.0    0.0    0.0    0.0    0.0    0.0    0    0
          extended device statistics
device   r/s    w/s    kr/s   kw/s  wait  actv  svc_t  %w   %b
fd0      0.0    0.0    0.0    0.0   0.0   0.0   0.0    0    0
sd0      0.6    0.0   38.4   0.0   0.0   0.0   8.2    0    0
sd1      0.0    0.0    0.0    0.0   0.0   0.0   0.0    0    0
(root@pbg-nv64-vn)-(11/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# swap -sh
total: 1.1G allocated + 190M reserved = 1.3G used, 1.6G available
(root@pbg-nv64-vn)-(12/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# uptime
12:53am up 9 min(s), 3 users, load average: 33.29, 67.68, 36.81
(root@pbg-nv64-vn)-(13/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# w
4:07pm up 17 day(s), 15:24, 3 users, load average: 0.09, 0.11, 8.66
User      tty          login@ idle   JCPU  PCPU  what
root      console     15Jun0718days    1      /usr/bin/ssh-agent -- /usr/bi
n/d
root      pts/3       15Jun07          18     4    w
root      pts/4       15Jun0718days          w
(root@pbg-nv64-vn)-(14/pts)-(16:07 02-Jul-2007)-(global)
-(/var/tmp/system-contents/scripts)#
```


Funzioni (5)



Funzioni ⁽⁶⁾

- Rilevamento d'errori
- Capacità di rilevare errori nella CPU o nei dispositivi
- Esempi:
 - Divisione per 0
 - Fine della carta
 - Guasto di una connessione di rete
- Per ogni errore è necessario eseguire una appropriata azione di recovery

SERVIZI OFFERTI DA UN SO

Servizi del sistema operativo (1)

Il SO fornisce **servizi** ai programmi e agli utenti dei programmi

- **Elaborazione**
 - il sistema deve potere caricare un programma in memoria e eseguirlo
- **Operazioni di I/O** – in generale gli utenti non possono controllare direttamente i dispositivi di I/O; è il SO che deve fornire i mezzi per compiere le operazioni di I/O
- **Manipolazione del file system** – i programmi devono poter leggere, scrivere, creare e cancellare i file

Servizi del sistema operativo (2)

- **Comunicazione** fra processi in esecuzione sullo stesso computer o in esecuzione su computer differenti collegati fra loro in rete
 - Le comunicazioni possono avvenire attraverso
 - una *memoria condivisa* o
 - attraverso *scambio di messaggi*
- **Rilevamento degli errori** – il sistema assicura un calcolo corretto per individuare errori
 - nella CPU
 - nell'hardware della memoria,
 - nei dispositivi di I/O e nei programmi dell'utente

Servizi del sistema operativo (3)

Esistono **servizi aggiuntivi** per una gestione/monitoraggio efficiente del sistema

- **Allocazione delle risorse** – *quando ci sono più utenti o più processi contemporaneamente in funzione, le risorse devono essere assegnate a ciascuno di loro*
- **Contabilità (accounting)**– *mantiene traccia di quali utenti usano le risorse e di quale tipo e genere di risorse si tratta per stilare statistiche d'uso o per la contabilità*
- **Protezione e sicurezza** – *implica la garanzia che tutti gli accessi alle risorse del sistema siano controllati*

Implementazione del SO

- **Tradizionalmente scritti in linguaggio assembly, oggi possono essere scritti in linguaggi ad alto livello**
 - un codice scritto in un linguaggio ad alto livello:
 - può essere scritto più velocemente
 - è più compatto
 - è più facile da capire e da mettere a punto
- Un SO è molto più flessibile e manutenibile se è scritto in un linguaggio ad alto livello
- I SO basati sui linguaggi si affidano alla **sicurezza del linguaggio**
 - Java è considerato un linguaggio sicuro
 - Linguaggi ritenuti sicuri sono utili per implementare SO su HW limitati (palmari, smart card, ecc..) che non hanno protezione fornita da HW

Generazione di un SO

- Tipicamente lo stesso SO può essere eseguito su più macchine
- Il sistema deve essere quindi configurato (generazione di sistema)

- Parametri
 - CPU
 - Memoria (indirizzo legale finale, i.e. dimensione)
 - Formattazione del disco di avvio (partizioni, dimensioni, contenuto)
 - Dispositivi disponibili
 - Opzioni (e.g. algoritmo di schedulazione, max. num. processi_)
- Vale anche per le macchine virtuali