



CHAPTER 15:

Operating Systems: An Overview

The Architecture of Computer Hardware, Systems Software & Networking: An Information Technology Approach

5th Edition, Irv Englander

John Wiley and Sons ©2013

PowerPoint slides authored by Angela Clark, University of South Alabama

PowerPoint slides for the 4th edition were authored by Wilson Wong, Bentley University

PowerPoint slides modified by Gianluca Amato, Univ. di Chieti-Pescara



Senza sistema operativo....

- I programmi devono essere caricati in memoria manualmente
- Ogni programma deve prevedere le proprie procedure di input/output
- Nessuna capacità di manipolazione di file
- Nessuna capacità di controllare le periferiche
- Il sistema esegue un solo programma alla volta.
 - Il sistema non fa nulla quando è in attesa di un input dall'utente



Definizione di sistema operativo

“Una raccolta di programmi che integrano le risorse hardware del computer e rendono queste risorse disponibili all'utente e ai programmi applicativi, in maniera che l'utente possa accedere al computer in maniera produttiva, rapida ed efficiente.”



Servizi di base di un S.O.

- Accetta comandi e richieste dall'utente e dai programmi applicativi e risponde in maniera appropriata.
- Gestisce, carica ed esegue programmi.
- Gestisce le risorse hardware del computer, rendendone possibile la condivisione tra programmi differenti
- **Nota:** il sistema operativo può essere visto come una collezione di centinaia o migliaia di programmi, ognuno specializzato per uno specifico compito.

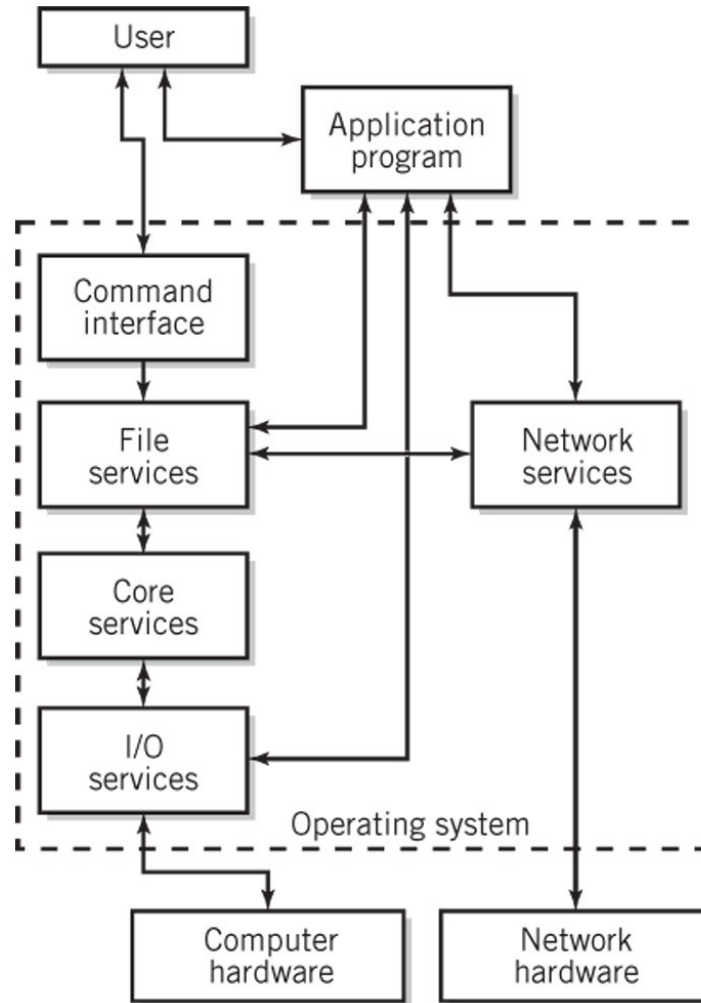


Multitasking

- Multitasking (multiprogramming)
 - Tecnica che consente l'elaborazione simultanea di più programmi, anche in presenza di una sola CPU.
- Multiprocessing
 - Esecuzione simultanea di più programmi in presenza di più CPU.
- Multiutenza
 - Evoluzione del multitasking nella quale più utenti possono accedere al computer in contemporanea.



Diagramma componenti di un S.O.





Componenti del S.O.

- Programma Bootstrap
- Componenti *residenti* in memoria
 - Sempre presenti in memoria
 - Comunemente denominato **kernel**
 - Contengono servizi essenziali richiesti da altre parti del S.O. e dei programmi utente
 - Responsabili per la gestione della memoria primaria, dei programmi in esecuzione, e della memoria secondaria.
- Componenti *non residenti*
 - Programmi usati poco frequentemente (formattazione disco)
 - Comandi



Tipi di sistemi

1. Singolo utente, singolo task (obsoleti)
 - MS-DOS
2. Singolo/multiutente, multitasking
 - Sistema predominante
 - Macintosh OSX, Windows, Linux, Oracle Solaris
3. Sistemi mainframe
 - Sviluppatisi per gestire risorse computazionali su larga scala
 - Estese capacità di I/O per gestire un numero enorme di transazioni
4. Server di rete
 - Focalizzati nel supportare client che ad esso si connettono
 - Sicurezza potenziata, alta affidabilità, backup



Tipi di sistemi

1. Sistemi operativi mobili

- Disegnati per dispositivi come PDA, telefoni e tablet
- Simili a sistemi singolo utente multitasking, ma con vincoli più stringenti su memoria principale e di massa, CPU e consumo di elettricità
- Necessità di rimpiazzare la tastiera come mezzo di input

2. Sistemi in tempo reale

- Alcuni processi devono essere in grado di reagire ad eventi esterni il più velocemente possibile
- Esempi
 - Sistemi di controllo degli aerei, dei propulsori a razzo o dei freni per le automobili



Tipi di sistemi

1. Sistemi di controllo embedded

- Sistemi specializzati sviluppati per controllare un singolo pezzo oggetto, come una automobile, un forno a microonde, una TV
- Il software è generalmente fornito su ROM
- Sistemi real-time dedicati ad applicazioni specifiche
- Esempio: General Motors Delphi system

2. Sistemi distribuiti

- Potenza di elaborazione distribuita tra computer in una cluster o rete
- Esempio: Distributed Computing Environment (DCE)



S.O. e utente

- Interattivi
 - Anche noti come **conversational systems**
- A lotti (batch)
 - L'utente sottomette programmi o **jobs** per l'elaborazione
 - Poca o nessuna interazione con l'utente



Hardware e S.O.

- Una piattaforma hardware può supportare vari sistemi operativi.
- Un sistema operativo può funzionare su una varietà di piattaforme hardware
- Un sistema operativo che lavora su piattaforme differenti
 - Assicura portabilità di file e programmi
 - Aumenta l'efficienza degli utenti grazie ad una interfaccia comune
 - È implementato con linguaggi di programmazione a medio/alto livello (C/C++) piuttosto che in assembly



Servizi e funzioni

- Interfaccia utente
- Sistema di gestione file
- Sistema di controllo dell'I/O
- Gestione dei programmi in esecuzione e della comunicazione tra programmi
- Gestione della memoria primaria
- Scheduling
- Gestione della memoria secondaria
- Gestione della rete
- Gestione della sicurezza e protezione dei sistemi
- Amministrazione di sistema



Interfaccia utente

- Tipi di interfacce utente:
 - CLI - Command Line Interface
 - GUI - Graphical User Interface
- Shell: programma che implementa la CLI
 - Uso interattivo o come linguaggio di programm.
 - UNIX/Linux: C, Bourne, bash and Korn shells
- Command Languages
 - IBM Mainframes – JCL
 - MS Windows – .BAT files, Windows Powershell
 - UNIX/Linux – shell scripts



Gestione dei file

- File – unità logica di memorizzazione
- Un sistema di gestione file (file system) prevede:
 - Struttura a directory per ogni dispositivo di memoria sec.
 - Strumenti per copiare, spostare, memorizzare, recuperare e manipolare file
 - Informazioni su ogni file nel sistema file (ex. data di creazione) e strumenti per accedervi
 - Meccanismi di sicurezza per proteggere i file
- Caratteristiche aggiuntive
 - Backup e recovery
 - Compressione dei file
 - Journaling
 - Accesso trasparente ai file su rete



Controllo dell'I/O

- *Device driver* (o anche solo *driver*)
 - Controllano un dispositivo, implementando interruzioni e controllo del DMA
 - Uniformano l'accesso a dispositivi differenti
 - Unico punto di accesso al dispositivo
- Aggiunta di nuovi driver *plug & play*
- Device driver durante il boot
 - I PC usano il BIOS (basic input/output system) o UEFI



Gestione dei processi

- *Processo*: programma in esecuzione
 - Memoria, risorse di I/O, registri, etc.
- Servizi di comunicazione tra processi
 - Esempio: una pipe in Unix e Windows
- Thread
 - Parte eseguibile individualmente di un processo
 - Condivide memoria e altre risorse con gli altri thread di un processo



Gestione memoria primaria

- Tiene traccia della memoria:
 - Identifica i programmi caricati in memoria
 - Ammontare di spazio che ogni programma usa
 - Memoria disponibile
 - Impedisce ai programmi l'accesso a memoria di altri programmi
- Mantiene code di programmi in attesa di essere caricati
- Alloca la memoria ai programmi caricati
- Dealloca la memoria al termine dei programmi
- Generalmente implementa la *memoria virtuale*



Scheduling

- Scheduling ad alto livello
 - Processi messi in coda sulla base di priorità ed eventualmente messi pronti in esecuzione
- Scheduling a basso livello (Dispatching)
 - Effettiva selezione dei processi da eseguire
 - Preemptivo / non-preemptivo
- Context switching
 - Trasferimento del controllo al processo scelto
 - Ripristina registri e altre informazioni di stato
- Tipi di processi: CPU vs. I/O bound vs. real-time



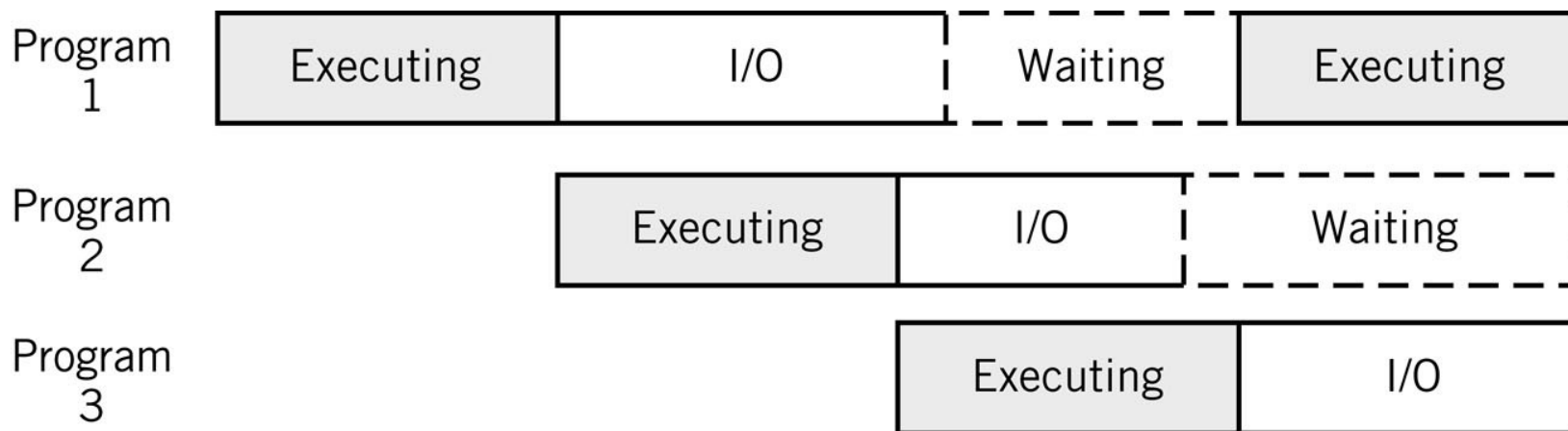
Multitasking

- Mentre un programma è in attesa della terminazione di una richiesta di I/O gli altri possono essere eseguiti.
 - L'I/O è una percentuale consistente nel tempo di esecuzione di un programma
- Time-slicing
 - La CPU può passare da un processo all'altro sulla base di un orologio



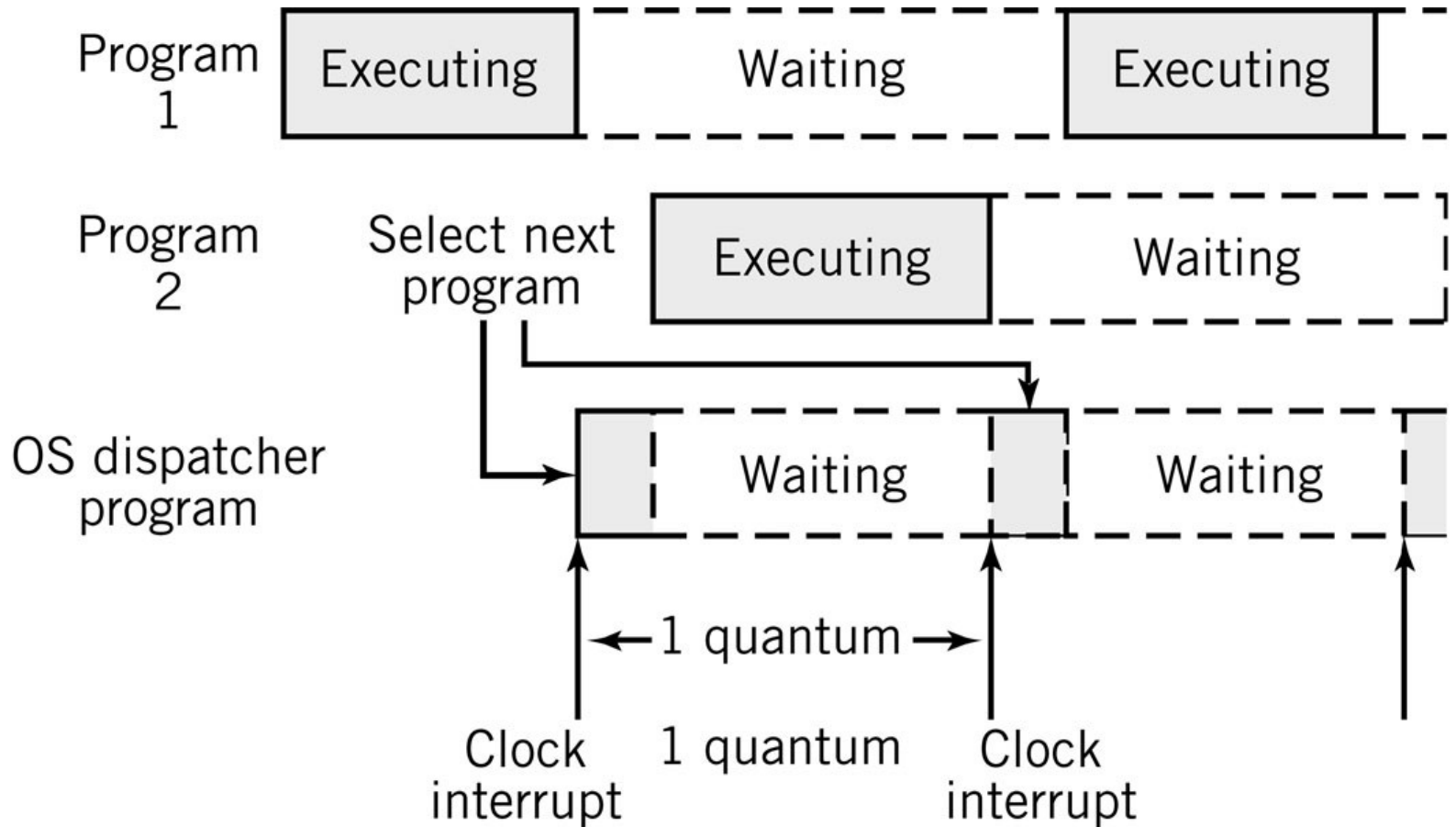
Multitasking e I/O

- L'I/O rappresenta una porzione consistente del tempo di esecuzione del programma





Time-slicing (preemption)





Memoria secondaria e sicurezza

- Gestione memoria secondaria
 - Ottimizza il completamento delle richieste di I/O
 - Combinazione di hardware e software
- Sicurezza e protezione dei servizi
 - Proteggere il S.O. dall'utente
 - Proteggerre un utente da altri utenti
 - Impedire accesso ad utenti non autenticati
 - Impedire azioni non autorizzate ad utenti autenticati



Servizi di rete e comunicazione

- TCP/IP
- Applicazioni di rete
 - Email, login remote, Web services, streaming multimediale, telefonia VOIP
- Funzioni avanzate
 - Supporto per cluster o altri sistemi distribuiti
 - Accesso comodo ai dati e dispositivi presenti in altri computer



Amministrazione di sistema

- Configurazione del sistema
- Aggiunta e cancellazione utenti
- Controllo e modifica privilegi utenti
- Sicurezza del sistema
- Gestione del file system
- Amministrazione di rete
- Backup
- Installazione e aggiorn. dei software
- Installazione e aggiornamento del S.O.
- Ottimizzazione del sistema
- Monitoring delle prestazioni
- Recupero dati perduti



Metodi di amministrazione

- Superutente
 - utente amministratore in grado di ignorare tutti i normali controlli di sicurezza
 - root in Unix/Linux, Administrator in Windows
- Unix/Linux
 - pannello di controllo grafico
 - vari programmi per la CLI
 - file di testo con configurazioni di sistema
- Windows
 - pannello di controllo
 - task manager (menù ottenuto con Ctrl-Alt-Del)
 - registro di sistema



Tipico report sullo stato di un sistema

The screenshot displays the GNOME System Monitor interface. The top window shows the 'Resident Sizes of Processes' section with a bar chart and a list of processes and their memory usage. The bottom window shows the 'CPU' section with a table of process details.

Resident Sizes of Processes

X	: 18592k
artsd	: 4096k
bash (2)	: 3156k
gimp	: 13860k
gtop	: 4776k
httpd (2)	: 11632k
in.identd (4)	: 2976k
kde	: 1240k
kdeinit (12)	: 103060k
kdesud	: 3824k
kdm (2)	: 2128k
klogd	: 1124k
knotify	: 8116k
kmsserver	: 5852k
ksnapshot	: 10364k
medusa-idled	: 2176k
mingetty (6)	: 3024k

CPU

PID	User	Pri	Size	Resident	Stat	CPU	MEM	Time	Cmd
856	root	17	26720	18592	S	2.6	7.2	1.18m	/usr/X11R6/bin/X
1914	iengland	15	4776	4776	R	1.6	1.8	16.35s	gtop
1239	iengland	9	9328	9324	S	0.0	3.6	8.85s	kdeinit: kwin -sessi
1	root	9	208	208	S	0.0	0.0	4.56s	init [5]
1250	iengland	9	13108	13104	S	0.0	5.1	4.03s	kdeinit: kicker
1229	iengland	9	4096	4096	S	0.0	1.6	3.01s	/opt/kde2/bin/artsd
2011	iengland	9	10364	10364	S	0.0	4.0	2.75s	ksnapshot
1244	iengland	9	10464	10464	S	0.0	4.0	1.65s	kdeinit: kdesktop
1102	iengland	9	2176	2176	S	0.0	0.8	1.58s	/opt/gnome/bin/medus
2020	iengland	9	13860	13860	S	0.0	5.4	1.36s	gimp
1906	iengland	9	9288	9284	S	0.0	3.6	0.83s	kdeinit: konsole -ic
1071	root	9	5804	5804	S	0.0	2.2	0.50s	/usr/sbin/httpd
1257	iengland	9	9408	9404	S	0.0	3.6	0.49s	kdeinit: konsole -se



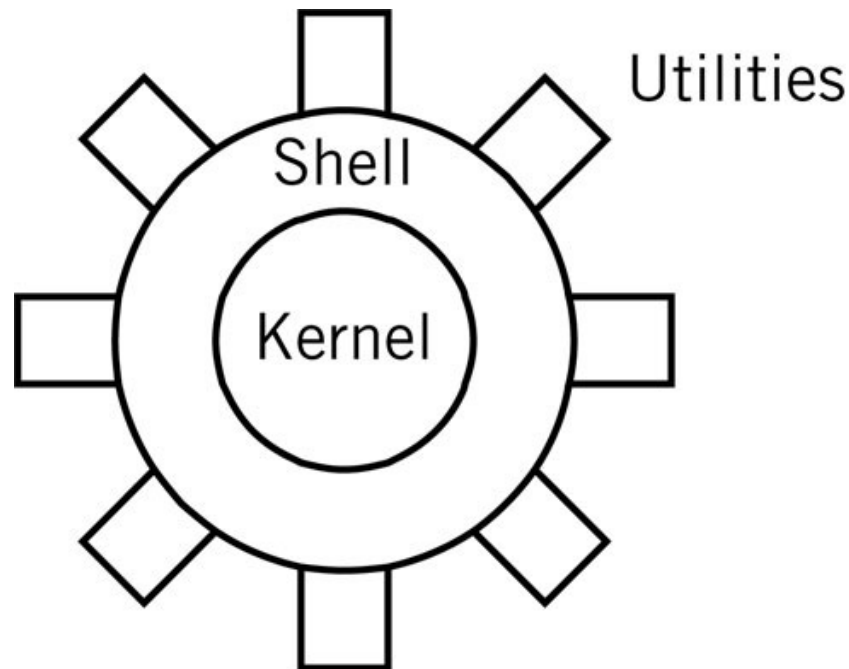
Organizzazione del S.O.

- Tre principali organizzazioni per un sistema operativo
 1. Monolitico
 - Unix/Linux
 2. Gerarchico
 - Windows 2000 e successivi
 3. Basato su microkernel
 - Mac OS X



Kernel monolitico

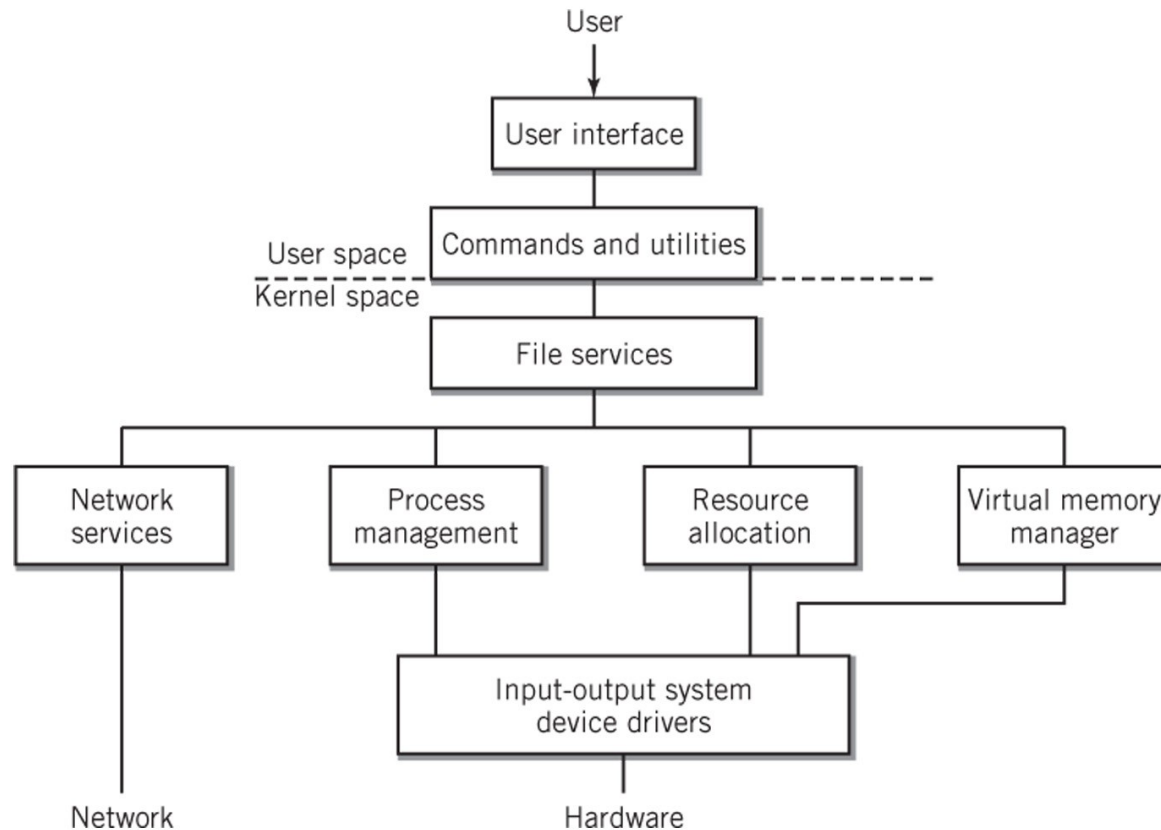
- Vantaggi: prestazioni
- Svantaggi: difficile avere un sistema stabile
- Examples: UNIX/Linux





S.O. Gerarchico

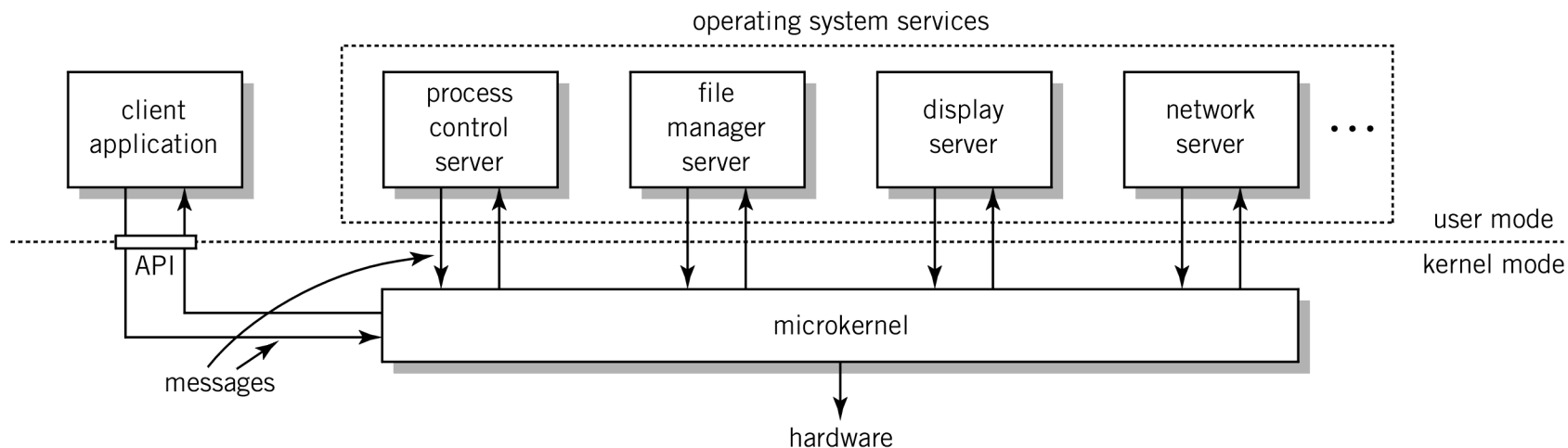
- Ogni strato è indipendente dagli strati sottostanti
- Le richieste sono passate da uno strato a quello sotto





Microkernel

- Kernel ha solo funzionalità essenziali
 - Cosa è funzionalità essenziale?
- Sistema client-server: il client chiede servizi al kernel, che incarica il server appropriato





Copyright 2013 John Wiley & Sons

All rights reserved. Reproduction or translation of this work beyond that permitted in section 117 of the 1976 United States Copyright Act without express permission of the copyright owner is unlawful. Request for further information should be addressed to the Permissions Department, John Wiley & Sons, Inc. The purchaser may make back-up copies for his/her own use only and not for distribution or resale. The Publisher assumes no responsibility for errors, omissions, or damages caused by the use of these programs or from the use of the information contained herein.