

# CAPITOLO 16: Il Sistema Operativo visto dall'utente

---

## **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 4<sup>th</sup> edition were authored by Wilson Wong, Bentley University

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



# Interfaccia utente (UI)

---

- Funzione primaria
  - Aiutare l'utente a usare il computer in maniera **produttiva**
  - Rendere accessibili all'utente le caratteristiche del computer
- Funzione secondaria
  - Fornire un aspetto comune (look and feel) alle applicazioni
    - Una interfaccia consistente diminuisce il tempo di apprendimento e aumenta la produttività



# Accesso ai servizi del S.O.

---

Tra approcci differenti:

## 1. Interfaccia a comandi

- Accetta comandi direttamente dall'utente
- Può essere grafica (GUI) o a linea di comando (CLI)

## 2. Linguaggio di comandi (command language)

- Accetta ed esegue gruppi di comandi come fosse un programma
- Anche chiamato linguaggio di script

## 3. Application Programming Interface (API)

- Accetta ed esegue richieste direttamente dai programmi utente.



# Tipici servizi di un S.O.

---

- Caricare ed eseguire programmi
- Manipolare file
- Servizi I/O (stampa, copia dischi, etc..)
- Sicurezza ed integrità dei dati
- Comunicazione e condivisione dati tra utenti
- Informazioni sullo stato del sistema
- Generazione di una interfaccia grafica
  - Apertura finestre, controllo mouse, etc..
  - Rivolto ai programmi, non agli esseri umani



# Esecuzione dei programmi

---

- Operandi
  - Nomi di file passati al programma
  - Parametri passati al programma che ne modificano il comportamento
- Interfaccia a linea di comando
  - Digitare il nome del programma e i parametri
- Interfaccia grafica
  - Double-click su una icona grafica del programma
  - Double-click su una icona grafica dei dati. Il programma associato con quel file è eseguito, e il nome del file gli viene passato come parametro



# Manipolazione dei file

- Categorie di comandi sui file
  - Lettura
  - Scrittura
  - Organizzazione
  - Gestione delle directory

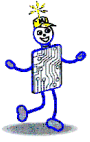
Windows	UNIX/Linux	
<b>dir</b>	<b>ls</b>	List a directory of files or get information about files
<b>copy</b>	<b>cp</b>	Copy a file from one place to another
<b>move</b>	<b>mv</b>	Move a file from one place to another
<b>del</b> or <b>erase</b>	<b>rm</b>	Delete (remove) a file
<b>type</b>	<b>cat</b>	Type a file out to the screen (or redirected to a printer)
<b>mkdir</b>	<b>mkdir</b>	Attach a new subdirectory to the tree at this tree junction
<b>rmdir</b>	<b>rmdir</b>	Delete a subdirectory



# Programmi di utilità

---

- Usati al posto di programmi applicativi per manipolare file di dati
- Possono essere combinati con un linguaggio di script
- Esempi
  - Ordinare dati in un file
  - Recuperare in maniera selettiva dati da un file
  - Modificare dati in un file



# Comandi di I/O

---

## Comandi per operazioni dirette di I/O

- Formattazione e controllo dischi
- Copia di dischi
- Invio e rimozione di documenti dalla coda di stampa
- Montaggio e smontaggio di dispositivi
  - Attaccare o staccare l'albero delle directory di un dispositivo all'albero directory principale
  - Rimozione sicura di dispositivi in Windows





# Servizi di comunicazione

---

- **Condivisione dei programmi**
  - Mettere i programmi condivisi in una zona dove sono raggiungibili da tutti gli utenti
- **Condivisione di dati**
  - Utenti multipli che lavorano su un documento
  - Database
- **Servizi di scambio di messaggi**
  - E-mail, FTP, terminal facilities (telnet, ssh), http, instant messaging, audio and video conferencing
- **Servizi per la comunicazione tra programmi**
  - Redirezione di file e pipe



# Sicurezza e integrità dei dati

---

- Protezione da lettura, scrittura o esecuzione non autorizzata
- Protezione dalla cancellazione
- Unix: tre livelli di sicurezza nella forma di privilegi di lettura, scrittura, esecuzione per il proprietario del file, il proprio gruppo e tutti gli altri.
- Windows: access control lists (ACLs)



# Informazioni sul sistema

---

- Informazioni comuni sul sistema
  - Ammontare di spazio disponibile su disco
  - Ammontare di memoria principale libera
  - Numero e identità degli utenti del sistema
  - % di tempo che la CPU o i canali di I/O sono occupati
- Funzionalità di *logging* che mantengono traccia in un file di tutti gli eventi verificatisi nel sistema



# Amministrazione del sistema

---

- Strumenti per analizzare e controllare il sistema
  - Attivare o disattivare features
  - Impostare connessioni di rete
  - Controllare l'esecuzione dei programmi
  - Aggiungere o rimuovere utenti, cambiare i permessi, limitare l'uso di spazio disco



# Progetto di interfacce utente

---

- Due tipi principali di interfacce
  - CLI - Command Line Interface
    - Prompt comandi Windows
    - Prompt comandi UNIX/Linux
    - Storicamente il primo tipo di interfaccia
  - GUI - Graphical User Interface
    - Hanno soppiantato la CLI per la maggior parte dell'uso quotidiano
    - Apple Macintosh, Windows, la maggior parte delle versioni di Linux



# Command Line Interface

---

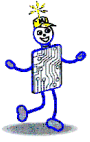
- `command <operand1> <operand2> ...`  
`<switch1> <switch2> ...`
- Operandi
  - Parole chiavi (switches) e/o posizionali
- Esempio Windows per salvare il listato di una directory in un file:
  - `dir pathparta\pathpartb > putfilea`
- Esempio Linux equivalente:
  - `ls -lF pathparta/pathpartb > putfilea`



# Batch System Commands

---

- Interfaccia simile agli interpreti su linea di comando:
  - `command <operand1> <operand2> ...`
- Specifica dove si trova il programma da eseguire e i dati da usare
- Usa un Job Control Language (JCL)
- I programmi sono eseguiti senza interazione con esseri umani
- Esempio: IBM zOS/Job Control Language
- Qual è la differenza con gli script della shell?



# Graphical User Interfaces

---

- Paradigma WIMP: Windows Icons Menu and Pointing device.
- Esempi: Apple Mac e Windows PCs; smartphone Apple, Android eBlackberry; tablet
- WYSIWYG
- Sistemi a finestre
  - Possono variare nelle apparenza but hanno in comune molti elementi grafici e operativi
  - Contengono la “scrivania”, icone, finestre, barra dei title, barra dei task, orologio, barra dei menù, gadget e widget.

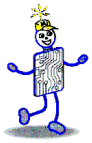




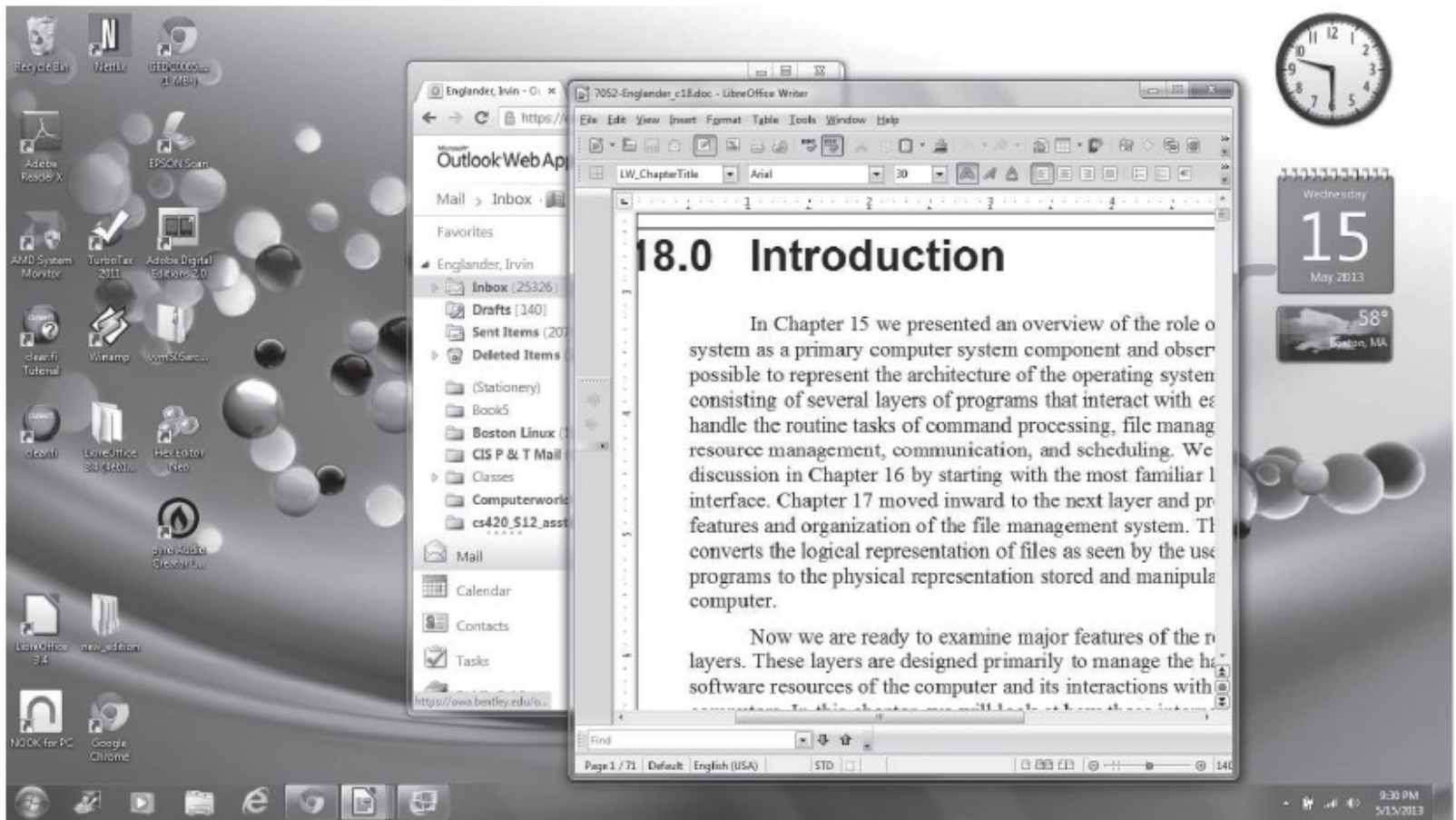
# Progetto di interfacce utente

---

- Web browser come interfaccia utente
  - Not propriamente parte del S.O.
  - Fornisce una interfaccia semplice, consistente, molto adatta a utenti poco esperti.
  - C'è un grosso proliferare di applicazioni con una interfaccia basate su web

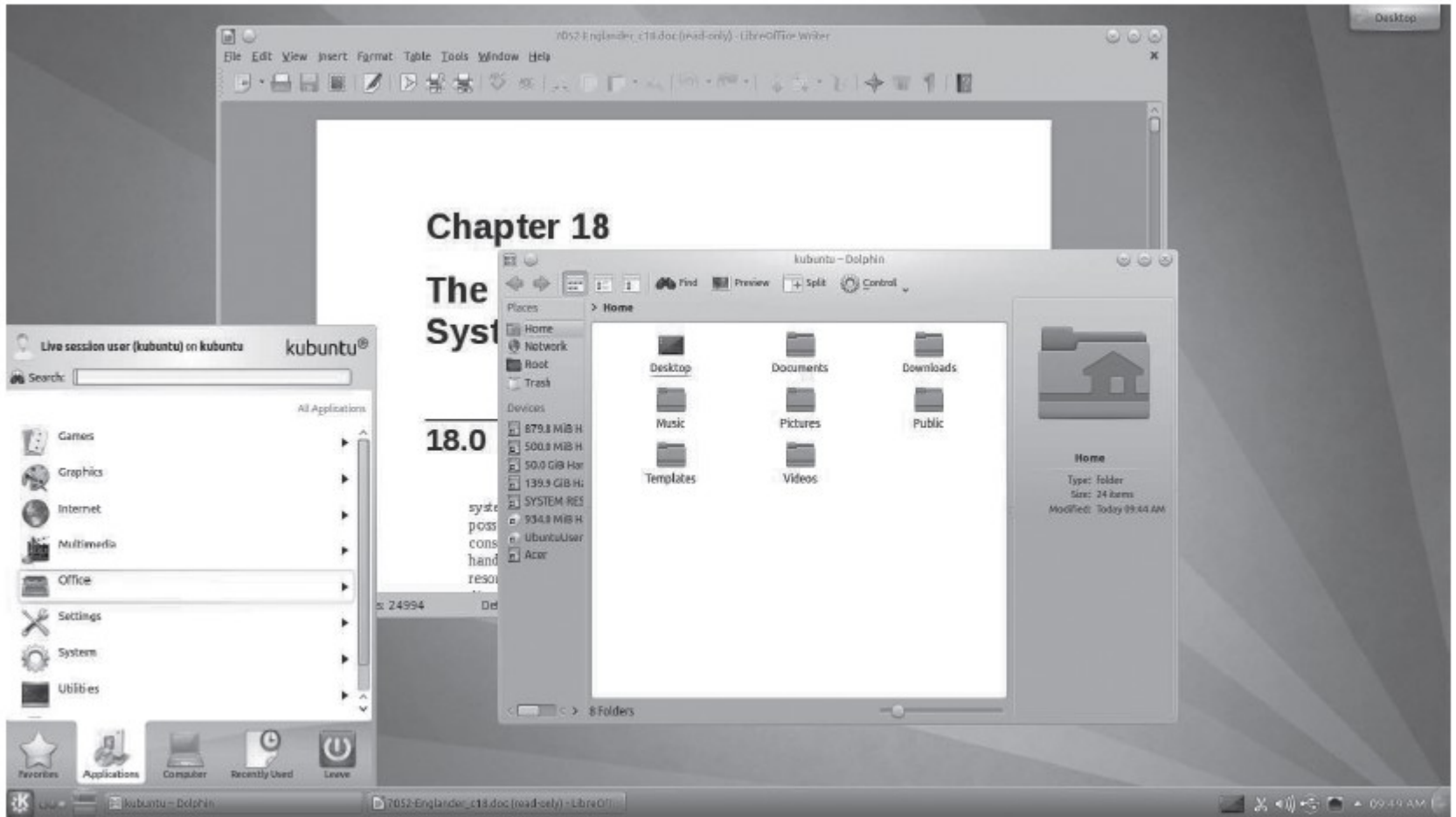


# GUI – Windows 7





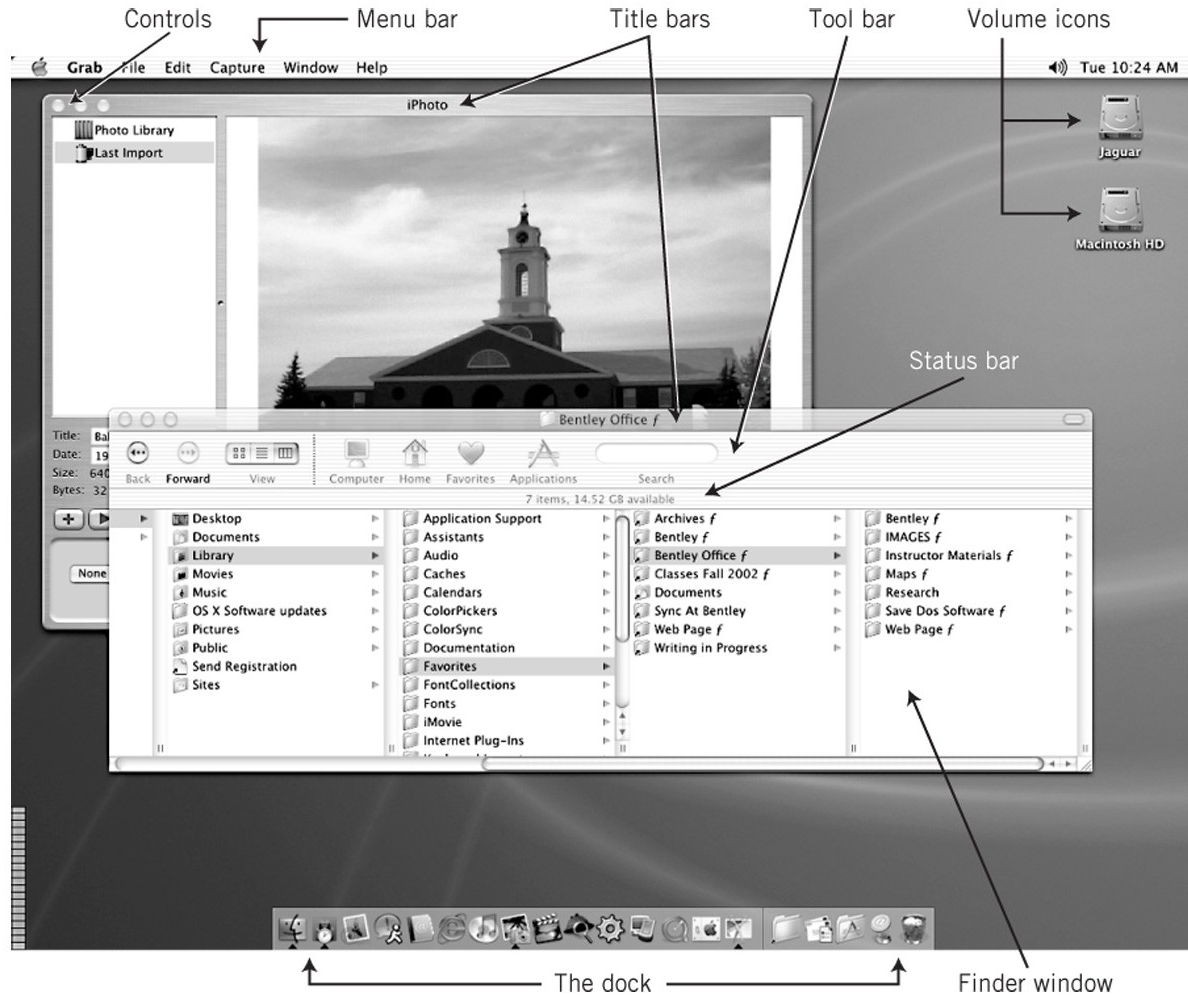
# GUI Interface – Linux KDE







# GUI – Macintosh OSX





# Interfacce gestuali e vocali

---

- L'uso di comandi vocali e di gesti (senza contatto fisico) sta crescendo rapidamente
- Originariamente sviluppati per i videogiochi, ma ora applicati in altri ambiti
- Esempi:
  - Movimenti del corpo nei giochi basati su Microsoft Kinect per XBox
  - Comandi vocali Siri per gli Apple iPhone
  - Display che segue il movimento degli occhi nei Samsung S4



# GUI vs. CLI

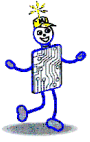
---

## GUI

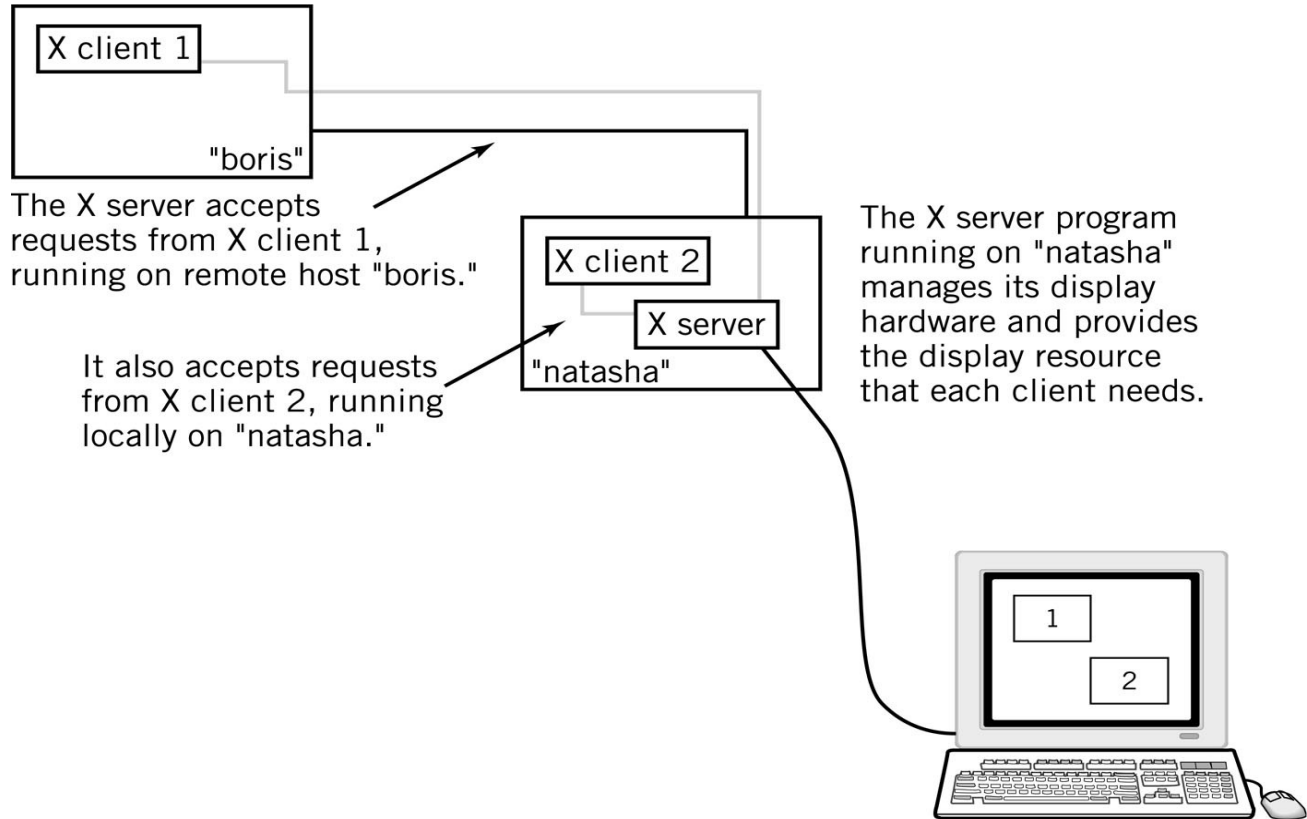
- Vantaggi
  - Facile da usare e imparare
  - Molto adatta al multi-tasking
- Svantaggi
  - Difficile da implementare
  - Richiede più potenza HW
  - Richiede molta memoria
  - Il software è più complesso da scrivere

## CLI

- Vantaggi
  - Più potente e flessibile
  - Più veloce per gli utenti esperti
  - Si possono combinare comandi
  - Facile applicare comandi a file multipli
- Svantaggi
  - Più difficile da usare e imparare



# X-Windows



Source: From *The UNIX Operating System 3e*, K. Christian and S. Richter, copyright © 1994.  
Reprinted by permission of John Wiley & Sons, Inc.





# Multicomputer X-Window Display

The screenshot illustrates a multicomputer X-Window display. It features a local PC window titled "MAIL" with a menu bar (File, Edit, Pick, Create-Send, Read, Maintenance, Customize, Help) and a list of messages. The messages are:

Msg No.	Date	From	Subject
1	21-DEC-1999	EOS::CHEYNE_BOBE	Peer Evaluat.
2	21-DEC-1999	EOS::CHEYNE_BOBE	CS 380 Grade
3	21-DEC-1999	IN@dinius_anne-mari	beneficiary
4	21-DEC-1999	EOS::LABBE_MICH	GROUP EVALUA

Other windows visible include "Document - WordPad" with text: "This is some text being generated on the local PC, to demonstrate XWindow.", "My Computer" showing a file explorer, and "xterm" displaying a directory listing of files like lib%.h, www, e, ireland-xcless, etc. Annotations include "Local PC" pointing to the MAIL window, "Sun workstation via modem" pointing to the xterm window, and "VAX VMS computer on local area network" pointing to the bottom of the MAIL window.



# Linguaggio di Comandi

---

- Fornisce un meccanismo per combinare sequenze di comandi. Gli pseudo-programmi che si ottengono sono noti come **script** o **file batch**.
- File di Startup – configurazione del SO, preferenze utente
- Caratteristiche di un linguaggio di comandi
  - Accetta input/output dall'utente ma anche da dispositivi o file
  - È in grado di creare e manipolare variabili
  - Supporta cicli e comandi condizionali
  - Consente di specificare argomenti ai programmi e agli script, e di accedere agli argomenti tramite variabili
  - Fornisce riconoscimento e recupero di errori



# Windows Program DOWP

---

```
@echo
if '%1' == '' go to error
if not exist %1 goto error
if not exist %1.txt goto newtxt
if exist %1.old copy %1.old e:\%.arc
copy %1.txt %1.old
wordproc %1.txt
goto end
:newtxt
echo This is a new file. Opening word processor....
wordproc
goto end
:error
echo proper command format is 'dowp filename'
echo with no extension.
:end
```



# Esempio: Linux Shell Script

```
export city state lat long port
grep -i "$1 $2" townfile | read city state lat long port
if [ -z "$city" ] then
    echo "this city is not in the file"
elif [ "$port" = "y" ] then
    echo " $city $state has its own airport"
else
    awk '
    BEGIN {close = 10000}
    $5 == "y" {dist = ($3 - '$lat')*($3 - $lat)+($4 - '$long')*($4 - '$long')}
        if (dist < close) {
            close = dist
            ccity = $1
            cstate = $2 } }
    END {print ( "the nearest airport is in " ccity, cstate)
        print ( " approximate distance is " 60* sqrt (close) " miles")
        } ' townfile
fi
```

A typical line in townfile:

Boston MA 42.3333 71.083 y



# 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.