

---

# Systeme d'Exploitation et Outils de programmation Partie 1

# Introduction

---

- Le système d'exploitation (SE) est un ensemble de programmes qui
  - Réalise l'interface entre le matériel de l'ordinateur et les utilisateurs
  - Prend en charge la gestion des ressources de la machine et le partage de celles-ci
    - ▣ **physiques**: mémoire, unités E/S, UCT...
    - ▣ **Logiques = virtuelles**: fichiers et bases de données partagés, canaux de communication logiques, virtuels...
    - ▣ les ressources logiques sont bâties par le logiciel sur les ressources physiques

# Pourquoi étudier les SE?

---

- Logiciel très important...
  - tout programme roule sur un SE
  - interface usager-ordinateur
- Les SE utilisent beaucoup d'algorithmes et structures de données intéressants
  - Les techniques utilisées dans les SE sont aussi utilisées dans nombreuses autres applications informatiques
    - ▣ il faut les connaître

# Développement de la théorie des SE

---

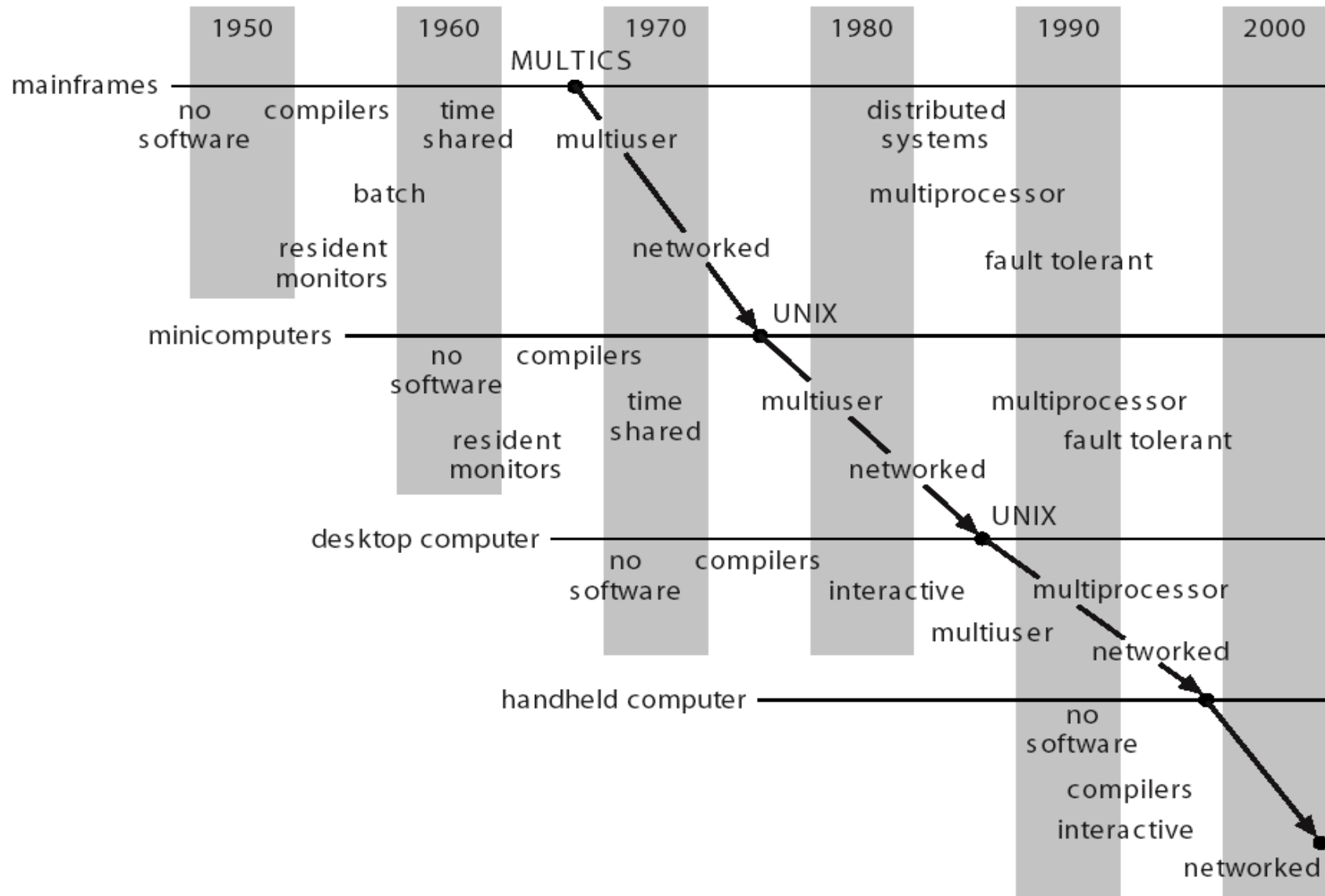
- La théorie des SE a été développée surtout dans les années 1960 (!!)
- A cette époque, il y avait des machines très peu puissantes avec lesquelles on cherchait à faire des applications comparables à celles d'aujourd'hui (mémoire typique: 100-500K!)
- Ces machines devaient parfois desservir des dizaines d'utilisateurs!
- Dont le besoin de développer des principes pour optimiser l'utilisation d'un ordinateur.
- Principes qui sont encore utilisés

# Évolution historique des SE

---

- Le début: routines d'E/S, amorçage système
- Systèmes par lots simples
- Systèmes par lots multiprogrammés
- Systèmes à partage de temps
- Ordinateurs personnels
- SE en réseau
- SE répartis
- ❖ *Les fonctionnalités des systèmes simples se retrouvent dans les systèmes complexes.*
- ❖ *Les problèmes et solutions qui sont utilisés dans les systèmes simples se retrouvent souvent dans les systèmes complexes.*

# Évolution des SE



# Une synthèse historique

---

## *Mainframes et grands serveurs*

Multics et beaucoup d'autres

(1960s)

Unix

(1970)

Solaris (1995)

## *Ordinateurs Personnels*

Mac/OS

(1984)

Linux

(1991)

MS-DOS

(1981)

Windows NT

(1988)

Windows

(1990)

Windows 2000

Windows XP

# Types de systèmes

---

- Mono utilisateur, monotâche
  - (single user, single tasking)
- Mono utilisateur, multitâches
  - (single user, multitasking)
- Multi utilisateurs, multitâches



# Multi utilisateurs, multitâches

---

- L'utilisateur de la machine peut interagir avec l'exécution de son programme =>
- Le but principal poursuivi par les systèmes interactifs
  - Offrir pour chaque exécution le plus petit temps de réponse possible
  - Travail en temps partagé
  - Exemple: système Unix ou Linux

# « Network operating systems »

---

- SEs en réseau fournissent:
  - Partage de fichiers (systèmes client-serveur)
  - Patrons de communication (protocoles)
  - Autonomie des ordinateurs

# Systemes temps réel

---

- Liés au contrôle de procédé
- Les exécutions de programmes sont soumises à des contraintes temporelles
- Exemple: LynxOS, VxWORKS

# Types de systèmes

---

- Systèmes embarqués
  - Système temps réel dédié pour une application particulière
    - ▣ Injection automatique pour une auto
    - ▣ Programmation micro-onde
- Systèmes distribués - répartis
  - il y a un SE qui fonctionne entre ordinateurs
  - l'utilisateur voit les ressources éloignées comme si elles étaient locales

# Services et facilités

---

- Le SE réalise une couche logicielle placée entre la machine matérielle et les applications
- Le SE typique fournis (+ ou -)
  - Interface usager-ordinateur
  - Gestion des fichiers
  - Support des entrées-sorties
  - Exécution de programmes: chargement, exécution (load, run), gestion de la concurrence

# Services et facilités

---

- Détection et traitement d 'erreurs
- Services Réseau et communication
- Gestion de la mémoire
- Gestion de la protection
- Support pour l'administration du système

# Services et facilités

---

- Support des entrées-sorties
  - Accès aux périphériques, la liaison entre les appels de haut niveau des programmes utilisateurs et les opérations de bas niveau de l'unité d'échange responsable du périphérique (pilote d'entrées-sorties)
  - Logiciels de contrôleur de périphérique ( I/O device driver), certains sont en ROM

# Services et facilités

---

- Services Réseau et communication  
(Network and Communications Support Services)
  - Fonctions nécessaire à l'interconnexion de l'ordinateur dans un réseau (protocole TCP-IP)
  - Programmes d'applications et extension (ex. E-mail, login, etc.)
  - Pilotes pour modems, communication sans fils, etc.



# Services et facilités

---

- Gestion de la protection
  - Mécanismes garantissant que les ressources de système ne peuvent être utilisées que par les programmes auxquels les droits nécessaires ont été accordés (protection de système et la machine des programmes utilisateurs)
    - ▣ Protège SE des programmes d'autres utilisateurs
    - ▣ Protège un utilisateur d'un autre utilisateur
  - Empêche les entrées non-autorisées dans le système (service de login)

# Services et facilités

---

- Support pour l'administration du système
- L'administrateur du système:
  - Configuration du système
  - Ajout ou enlèvement des utilisateurs/privilèges
  - S'occupe de la sécurité du système
  - Gère le système de fichier
  - S'occupe des sauvegardes, installation, mise à jour des logiciels et de SE
  - Optimise le système

# SE

---

- SE s'interface avec les applications du niveau utilisateur par le biais de fonctions prédéfinies – routines systèmes
- Les appels des routines systèmes peuvent se faire par biais
  - D'un appel système (API)
  - D'une commande du langage de commandes
- L'exécution des routines systèmes s'effectue sous un mode privilégié – **mode superviseur**
- Un programme utilisateur s'exécute par défaut selon un **mode utilisateur**

# SE

---

- Mode utilisateur : le jeu d'instructions utilisables par le programme est réduit
- Mode superviseur: aucune restriction
- Passage du mode utilisateur au mode superviseur constitue une **commutation de contexte**
- Trois causes majeures provoquent le passage du mode utilisateur au mode superviseur
  - Utilisateur appelle une fonction système
  - L'exécution d'une opération illicite (trappe ou l'exception)
  - La prise en compte d'une interruption par le matériel et SE

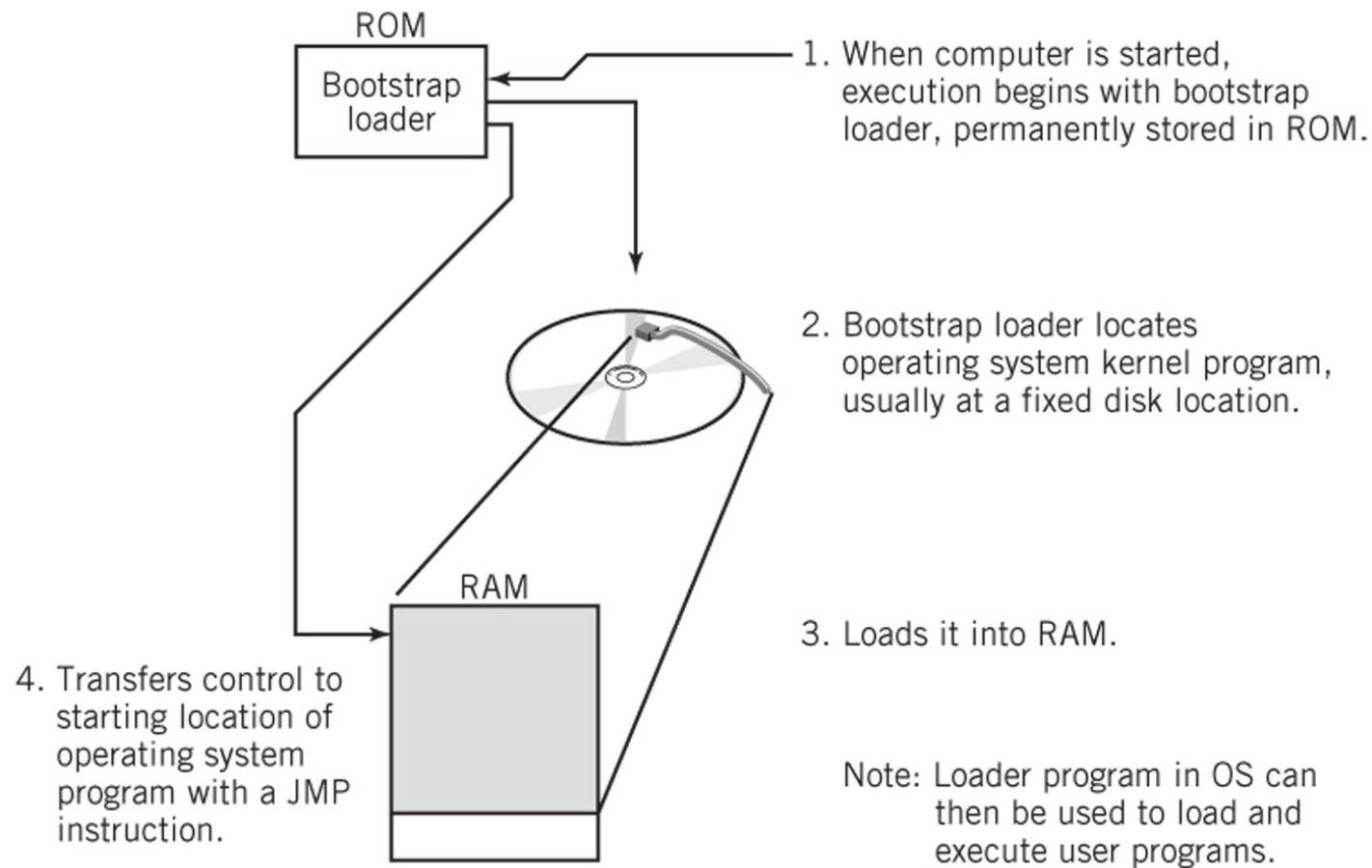
# Chargement d'un SE

---

- Les programmes composant le SE sont conservés sur un support de masse
- Au démarrage l'ordinateur exécute un code placé dans une zone de mémoire morte (ROM), programme d'amorçage ou bootstrap (test du matériel de la machine, chargement d'un programme d'amorce plus sophistiqué)
- Une fois cette amorce placée en mémoire centrale, elle s'exécute et charge les programmes du SE.

# Bootstrapping

---



# Le noyau (kernel) du SE

---

- La partie résidente (toujours en RAM) du SE est appelée *Kernel = noyau*
- Les autres parties sont amenées en RAM au besoin
- Contient les fonctionnalités critiques du SE: elles doivent toujours être prêtes à l'utilisation
  - traitement d'interruptions
  - gestion de UCT
  - gestion mémoire
  - communication entre processus
  - etc.
- À part ça, quoi exactement mettre dans le kernel est une question pour les concepteurs des SE
- La plupart des fonctionnalités discutées dans ce cours sont normalement dans le kernel

# Interface usager-ordinateur

---

- Types d'interfaces utilisateur
  - Interface de ligne de commande
    - ▣ CLI – Command Line Interface
      - ▣ Windows command prompt
      - ▣ UNIX/Linux command prompt
  - Interface graphique
    - ▣ GUI – Graphical User Interface
      - ▣ Apple Macintosh, Windows, stations de travail Sun, les différentes versions de Linux

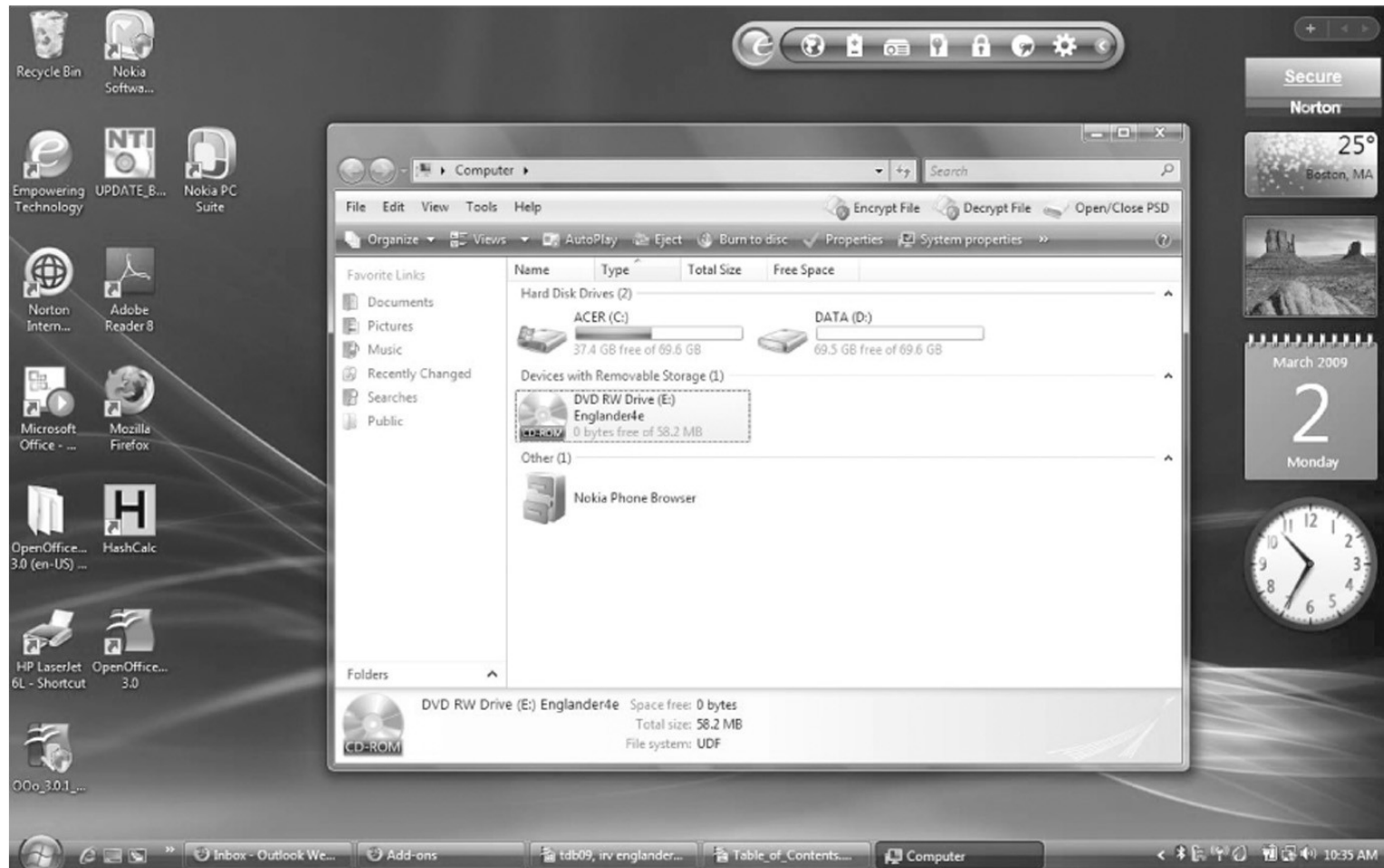


# Langage de commandes

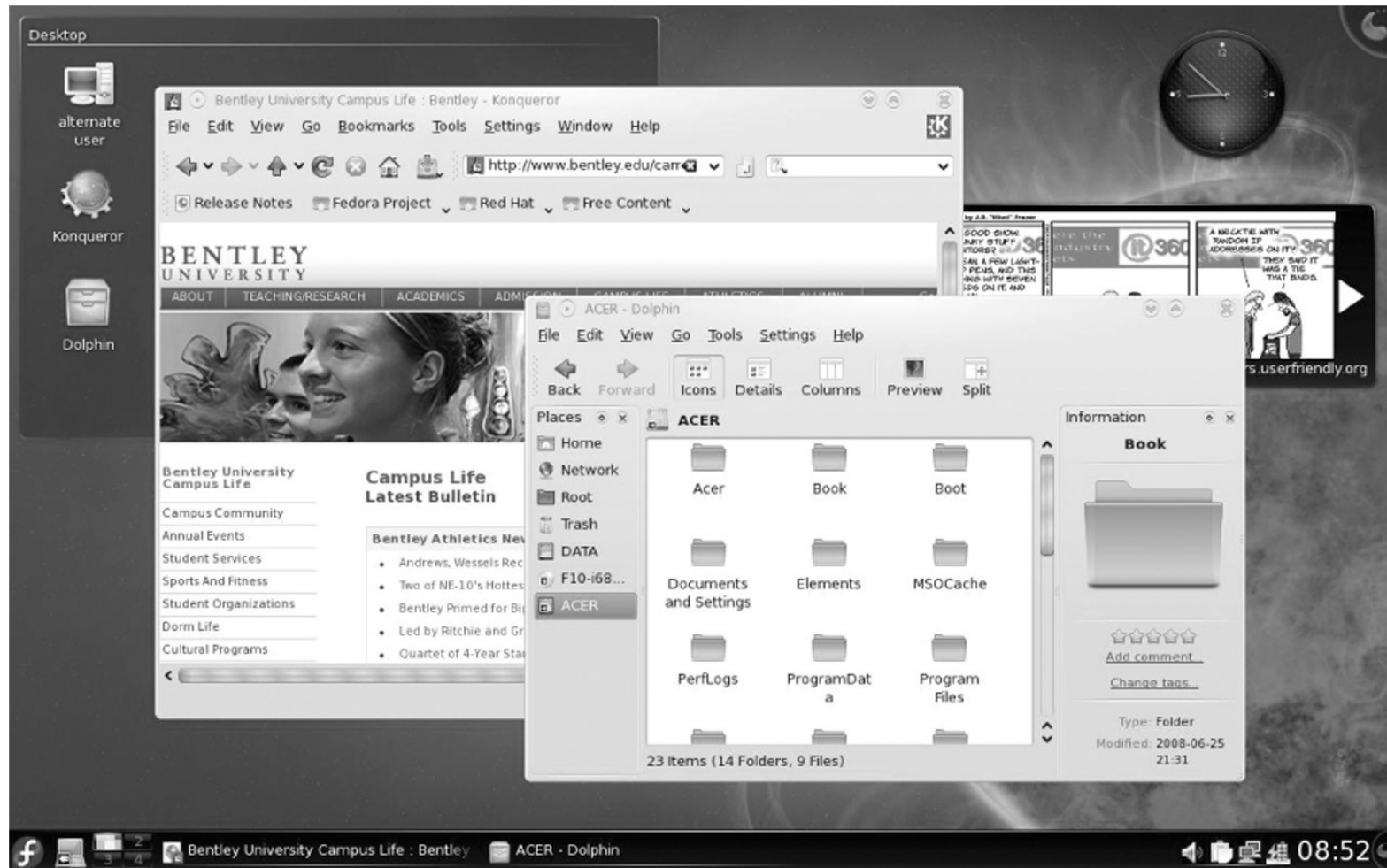
---

- Les ordinateurs modernes ont la possibilité de regrouper les commandes CLI en miniprogrammes
  - Langage de commandes (scripts)
    - ▣ Les commandes sont analysées par l'outil interpréteur de commande qui appelle la routine système appropriée en assurant le passage des paramètres
  - Chaque SE a son langage de commandes propre
    - ▣ JCL (Job Control Language) de SE MVS
    - ▣ Langage Shell de SE Unix

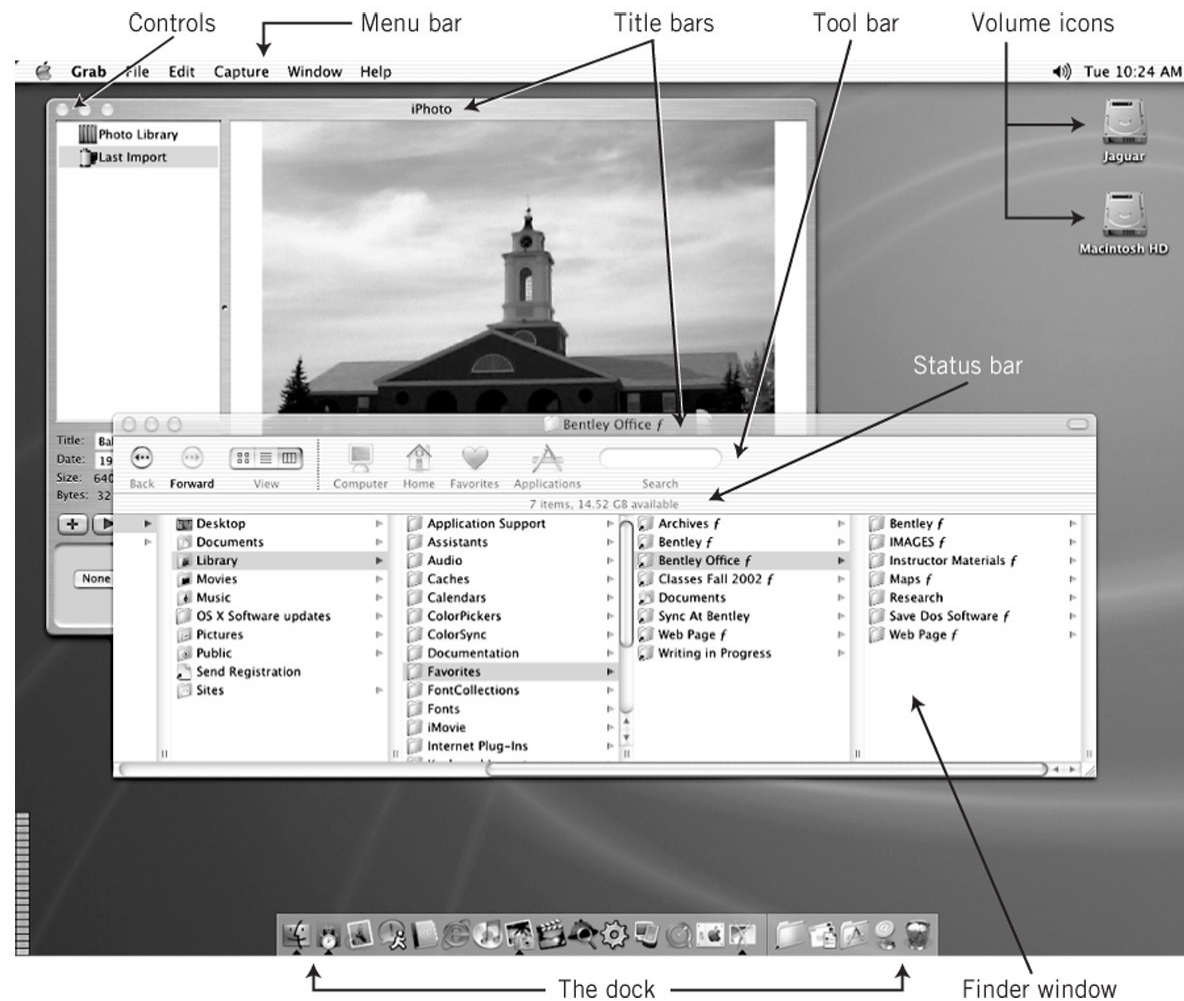
# GUI Interface – Windows Vista



# GUI Interface – Linux KDE



# GUI Interface – Macintosh OSX



# GUI vs. CLI

---

## GUI

- Avantages
  - Facile à apprendre et utiliser
- Désavantages
  - Plus difficile à implémenter
  - Plus exigeant en terme de HW/SW
  - Requier beaucoup de mémoire
  - SW est complexe et difficile à écrire

## CLI

- Avantages
  - Plus flexible et puissant
  - Plus performant pour utilisateurs expérimentés
  - Peut combiner des commandes
- Désavantages
  - Plus difficile à apprendre et utiliser