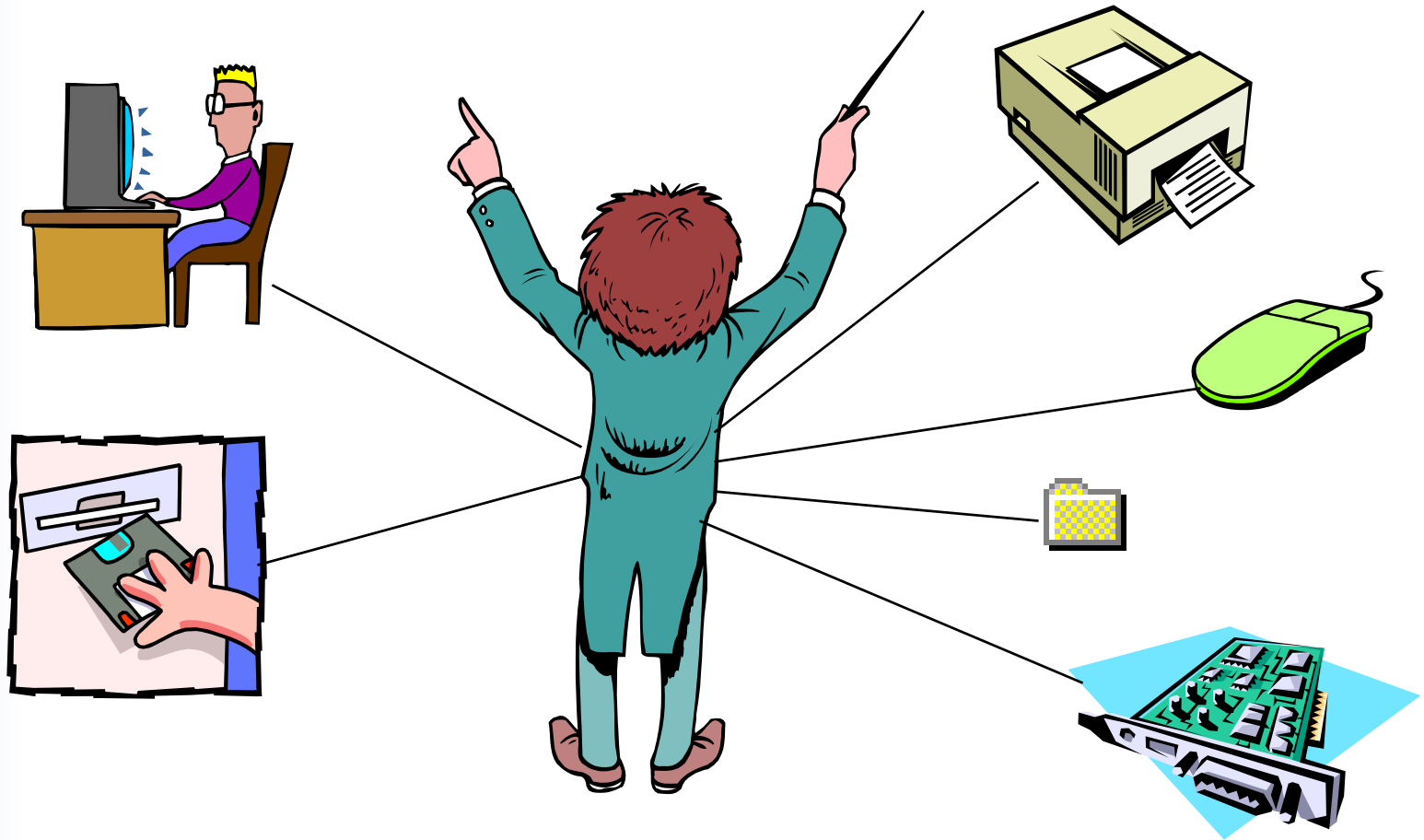


Introduction au système d'exploitation (SE) *Operating System (OS)*

Le Système d'Exploitation



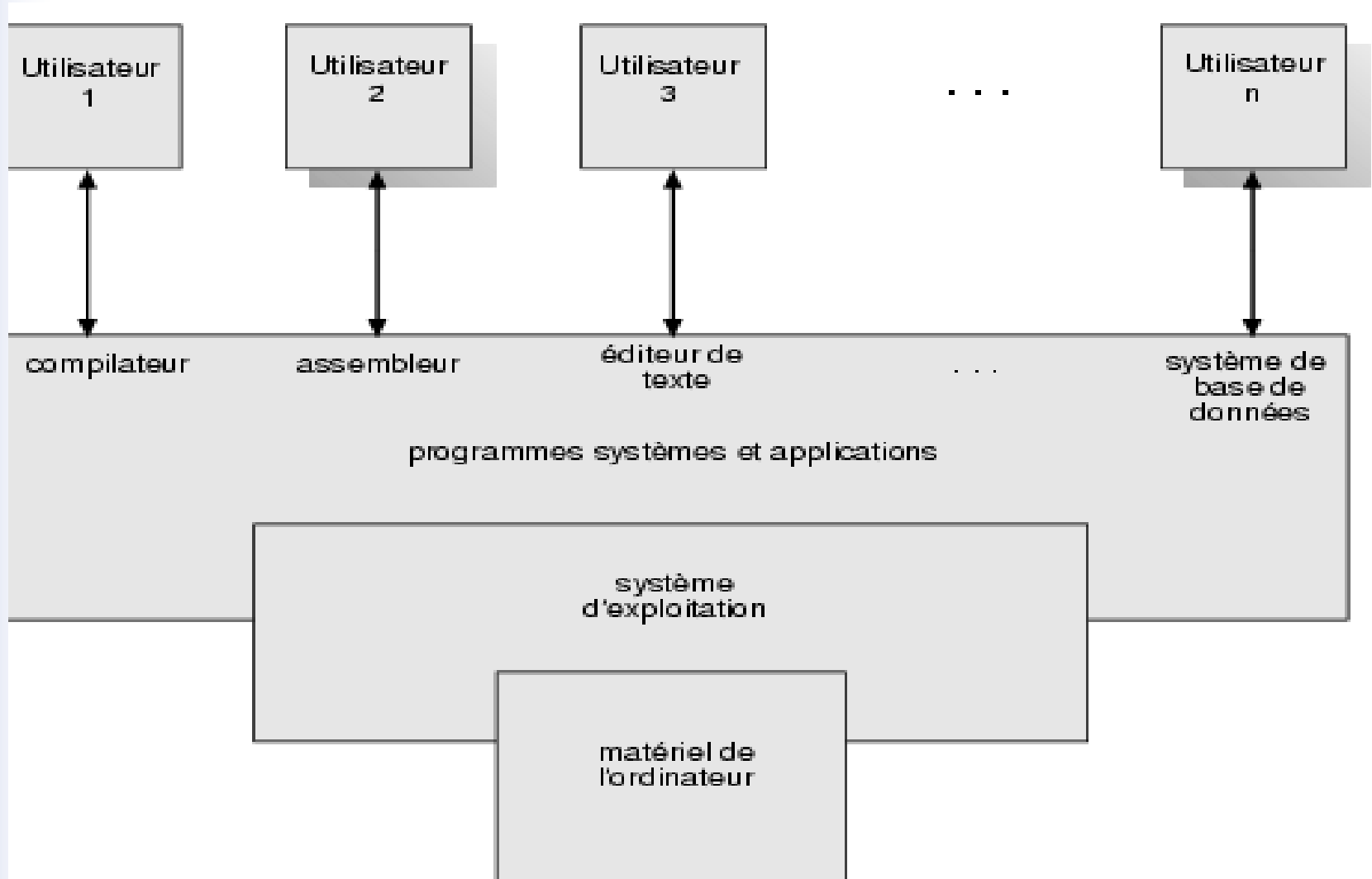
Concepts importants

- **Que c'est que un SE**
- **Évolution historique**
 - ◆ Par lots
 - ◆ Multiprogrammés – balance de travaux
 - ◆ À partage de temps (time-sharing)
 - ◆ Parallèles:
 - ☞ Fortement couplés
 - Symétriques,
 - Asymétriques: maître-esclave
 - ☞ Faiblement couplés:
 - Répartis
 - Réseaux
- **Caractéristiques de matériel et logiciel requises pour cette évolution**
- **Systèmes à temps réel: durs, souples**

Système d'exploitation (SE)

- Fournit l'interface usager/machine:
 - Masque les détails du matériel aux applications
 - Le SE doit donc traiter ces détails
- Contrôle l'exécution des applications
 - Le fait en reprenant périodiquement le contrôle de l'UCT
 - Dit à l'UCT quand exécuter tel programme
- Il doit optimiser l'utilisation des ressources pour maximiser la performance du système

Vue abstraite d'un SE



Ressources et leur gestion

- Ressources:
 - 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
- Allocation de ressources: gestion de ressources, leur affectation aux usagers qui les demandent, suivant certains critères

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
- 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.*

Les débuts

- Au début, on a observé qu'il y avait des fonctionnalités communes à tous les programmes
- il fallait les pré-programmer et les fournir au programmeur à moyen d'instructions d'appel:
 - amorçage du système
 - entrée/sortie

Systèmes de traitement par lots (*batch*) simples

- Sont les premiers SE (mi-50)
- L'utilisateur soumet un *job* (ex: sur cartes perforées) à un opérateur
- L'opérateur place un *lot* de plusieurs jobs sur le dispositif de lecture
- Un programme, le *moniteur*, gère l'exécution de chaque programme du lot
- Le *moniteur* est toujours en mémoire et prêt à être exécuté
- Les utilitaires du moniteur sont chargés au besoin
- Un seul programme à la fois en mémoire, programmes sont exécutés en séquence
- La sortie est normalement sur un fichier, imprimante, ruban magnétique...

Un ordinateur principal (mainframe) du milieu des années '60



disques

rubans

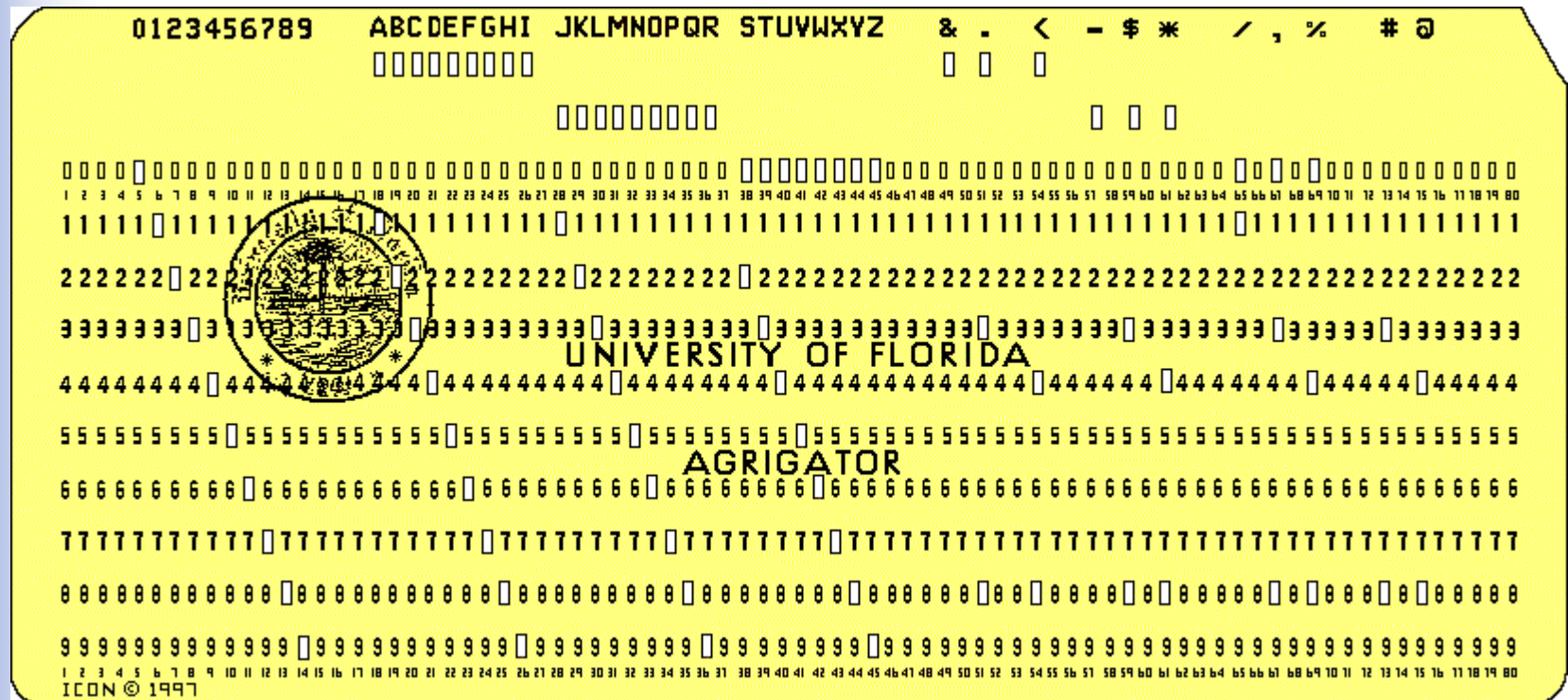
lecteur de cartes

UCT
(mémoire probablen.
autour de 500K)

console opérateur



Oui, cartes perforées...



Une ligne de données ou de programme était codée dans des trous qui pouvaient être lus par la machine

Opérateur lisant un paquet de



Source: http://www.tietokonemuseo.saunalahti.fi/eng/kuva_32_eng.htm

Finnish Data Processing Museum Association

Langage de contrôle des travaux (JCL)

- Utilisé pour contrôler l'exécution d'une job
 - le compilateur à utiliser
 - indiquer où sont les données
- Exemple d'une job: ----->>
- \$JOB initialise la machine
- \$FTN charge le compilateur et initie son exécution
- \$LOAD charge le pgm objet (à la place du compilateur)
- \$RUN transfère le contrôle au programme usager
- les données sont lues par le moniteur et passées au progr. usager

\$JOB

\$FTN

...

**Programme
FORTRAN**

...

\$LOAD

\$RUN

...

Données

...

\$END

\$JOB

...

(job suivi¹⁵.)

Caractéristiques désirables du matériel

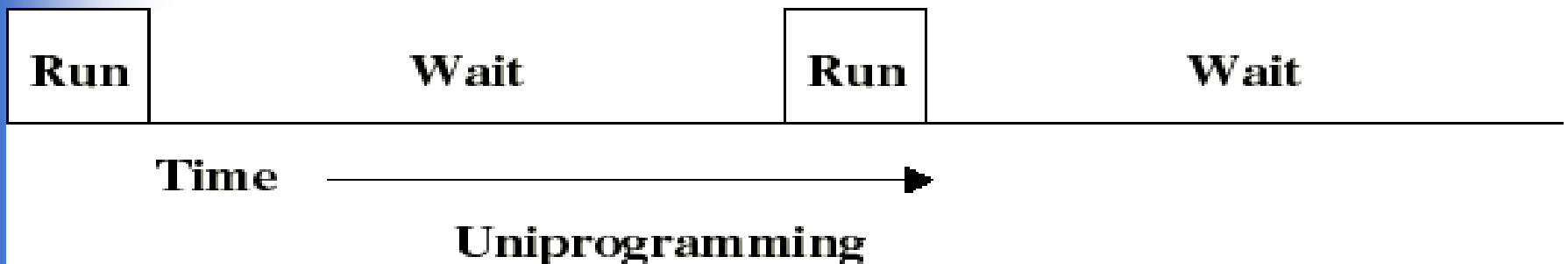
- Protection de la mémoire
 - ne pas permettre aux programmes usager d'altérer la région de la mémoire où se trouve le moniteur
- Minuterie
 - limite le temps qu'un job peut exécuter
 - produit une interruption lorsque le temps est écoulé

Les systèmes par lots

- Ont été les premiers systèmes d'exploitation.
- Ils sont associés aux concepts suivants:
 - langage de contrôle de travaux
 - système d'exploitation résident en mémoire (*kernel = noyau*)
 - protection de mémoire
 - instructions privilégiées
 - modes usager-moniteur
 - interruptions
 - minuterie
- Toutes ces caractéristiques se retrouvent dans les systèmes d'aujourd'hui

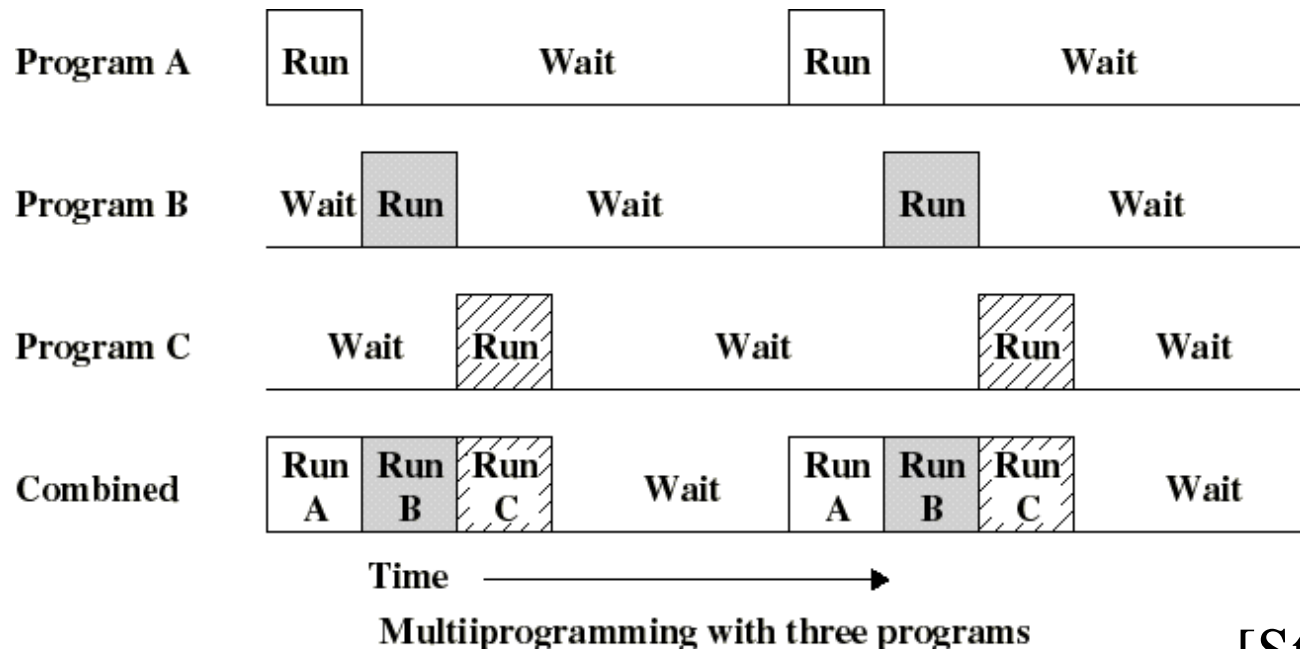
Traitement par lots multiprogrammé

- Les opérations E/S sont extrêmement lentes (comparé aux autres instructions)
- Même avec peu d'E/S, un programme passe la majorité de son temps à attendre
- Donc: pauvre utilisation de l'UCT lorsqu'un seul pgm usager se trouve en mémoire

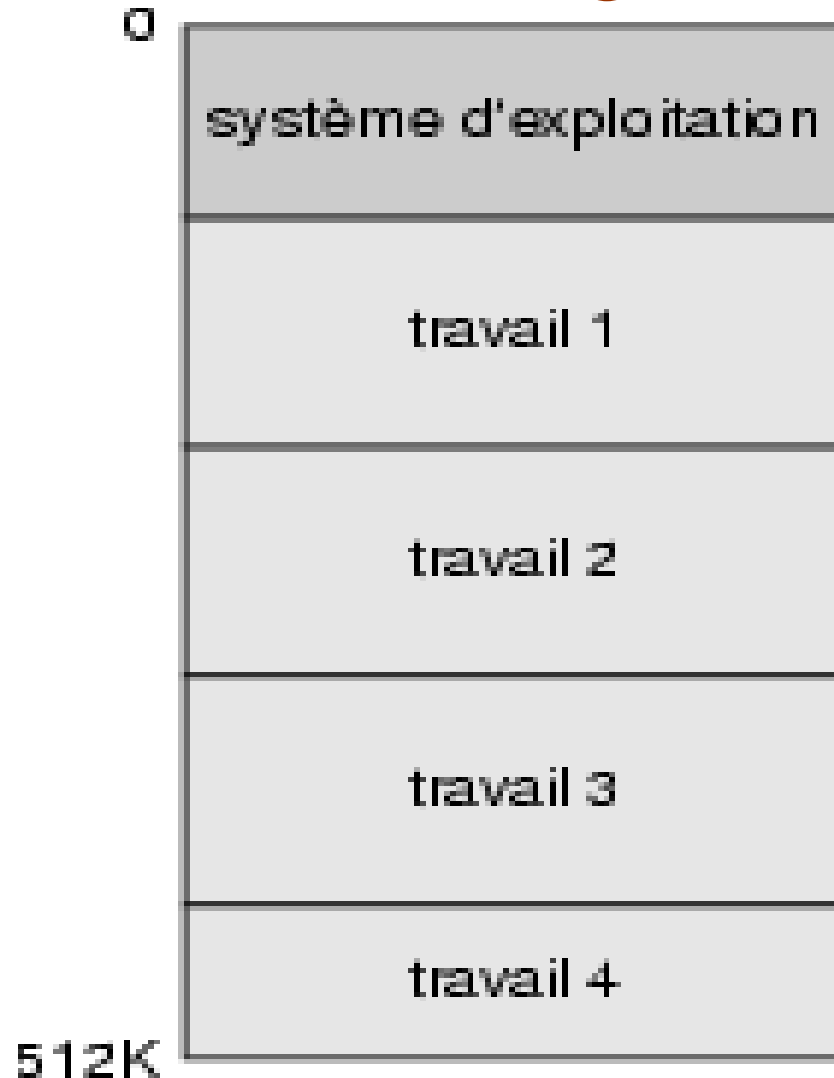


Traitement par lots multiprogrammé

- Si la mémoire peut contenir +sieurs pgms, l'UCT peut exécuter un autre pgm lorsqu'un pgm attend après E/S
- C'est de la multiprogrammation



Plusieurs programmes en mémoire pour la multiprogrammation



Spoule ou spooling

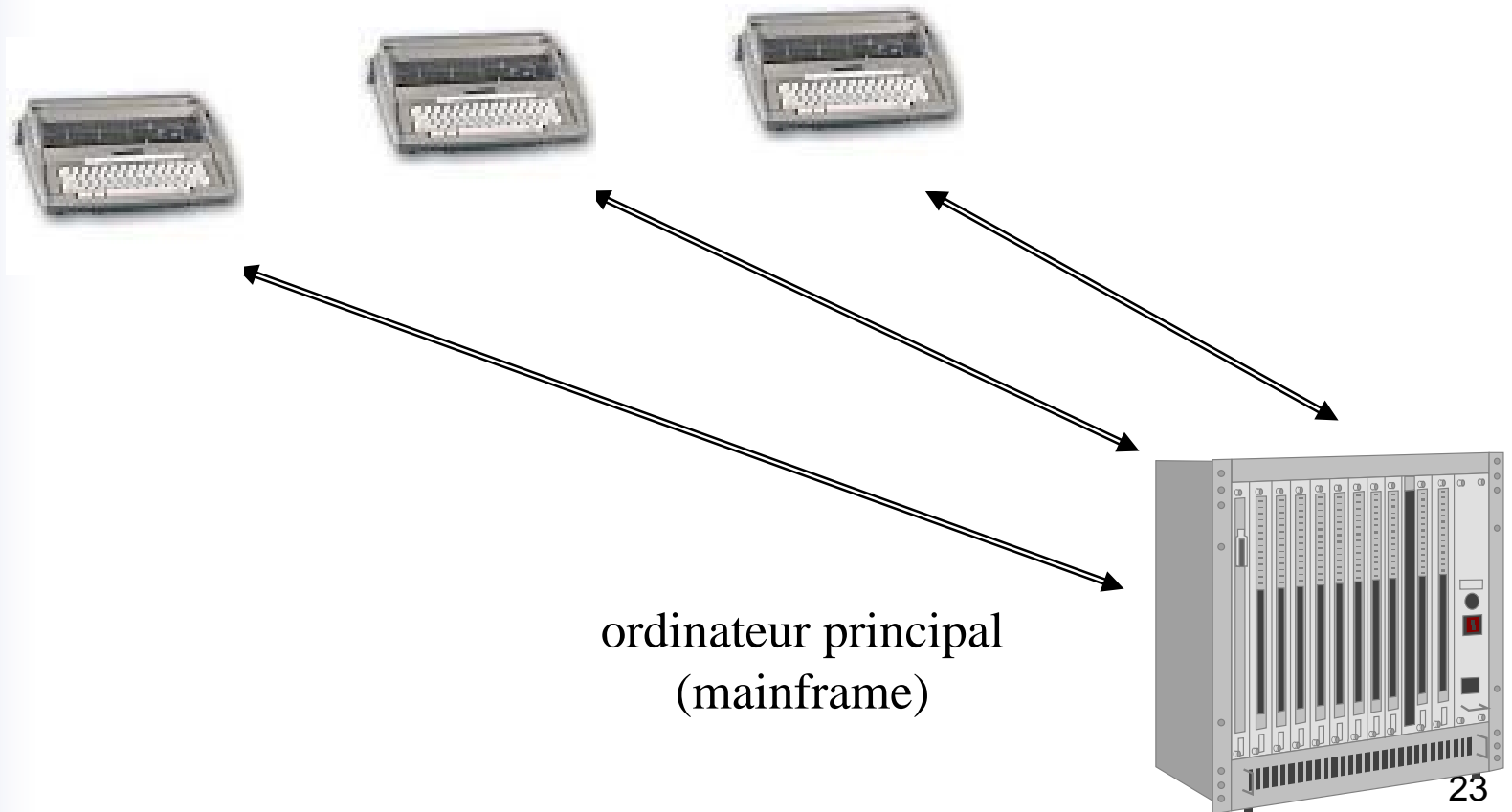
- Au lieu d'exécuter les travaux au fur et à mesure qu'ils sont lus, les stocker à l'avance sur une mémoire secondaire (disque)
- Puis choisir quels programmes exécuter et quand
- La mémoire secondaire contenait aussi les données d'E/S

Équilibre de travaux

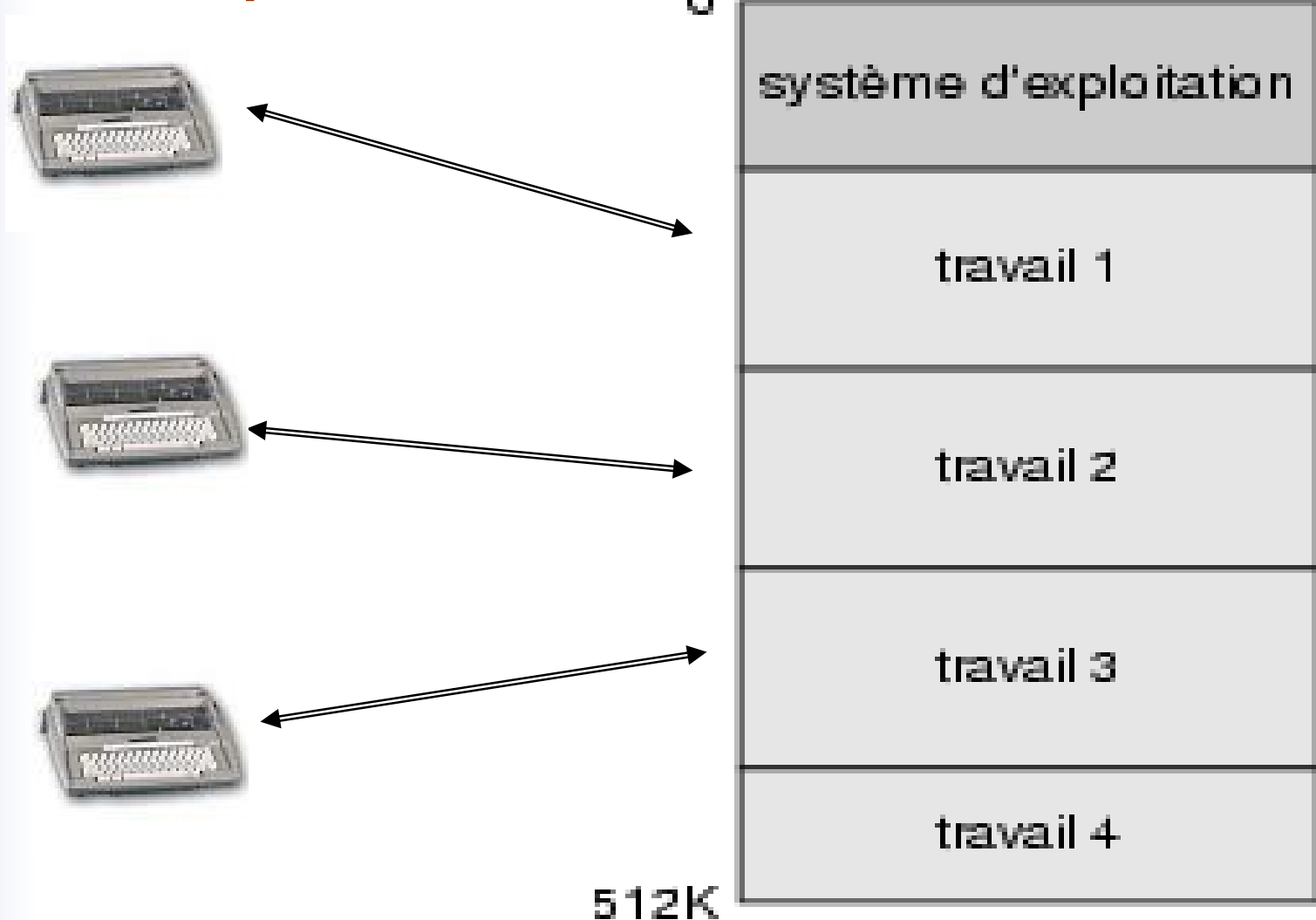
- S'il y a un bon nombre de travaux à exécuter, on peut chercher à obtenir un équilibre
- Travaux qui utilisent peu l'UCT, beaucoup l'E/S, sont appelés tributaires de l'E/S
- Nous parlons aussi de travaux *tributaires de l'UCT*
- Le temps d'UCT non utilisé par des travaux trib. de l'E/S peut être utilisé par des travaux trib. de l'UCT et vice-versa.
- L'obtention d'un tel équilibre est le but des ordonnanceurs à long terme et à moyen terme.
- Dans les systèmes de multiprog. on a souvent coexistence de travaux *longs et pas urgents* avec travaux *courts et urgents*
- Le SE donne priorité aux deuxièmes et exécute les premiers quand il y a du temps de machine disponible.

Systemes à temps partagé (TSS)

Terminaux
'stupides'



Chaque terminal a sa propre partition de mémoire



Systèmes à temps partagé (TSS)

- Le traitement par lots multiprogrammé ne supporte pas l'interaction avec les usagers
 - **excellente utilisation des ressources mais frustration des usagers!**
- TSS permet à la multiprogrammation de desservir plusieurs usagers simultanément
- Le temps d 'UCT est partagé par plusieurs usagers
- Les usagers accèdent simultanément et interactivement au système à l'aide de terminaux

Systèmes à temps partagé (TSS)

- Le temps de réponse humain est lent: supposons qu'un usager nécessite, en moyenne, 2 sec du processeur par minute d'utilisation
- Environ 30 usagers peuvent donc utiliser le système sans délais notable du temps de réaction de l'ordinateur
- Les fonctionnalités du SE dont on a besoin sont les mêmes que pour les systèmes par lots, plus
 - la communication avec usagers
 - le concept de *mémoire virtuelle* pour faciliter la gestion de mémoire
 - traitement central des données des usagers (partagées ou non)

MULTICS et UNIX

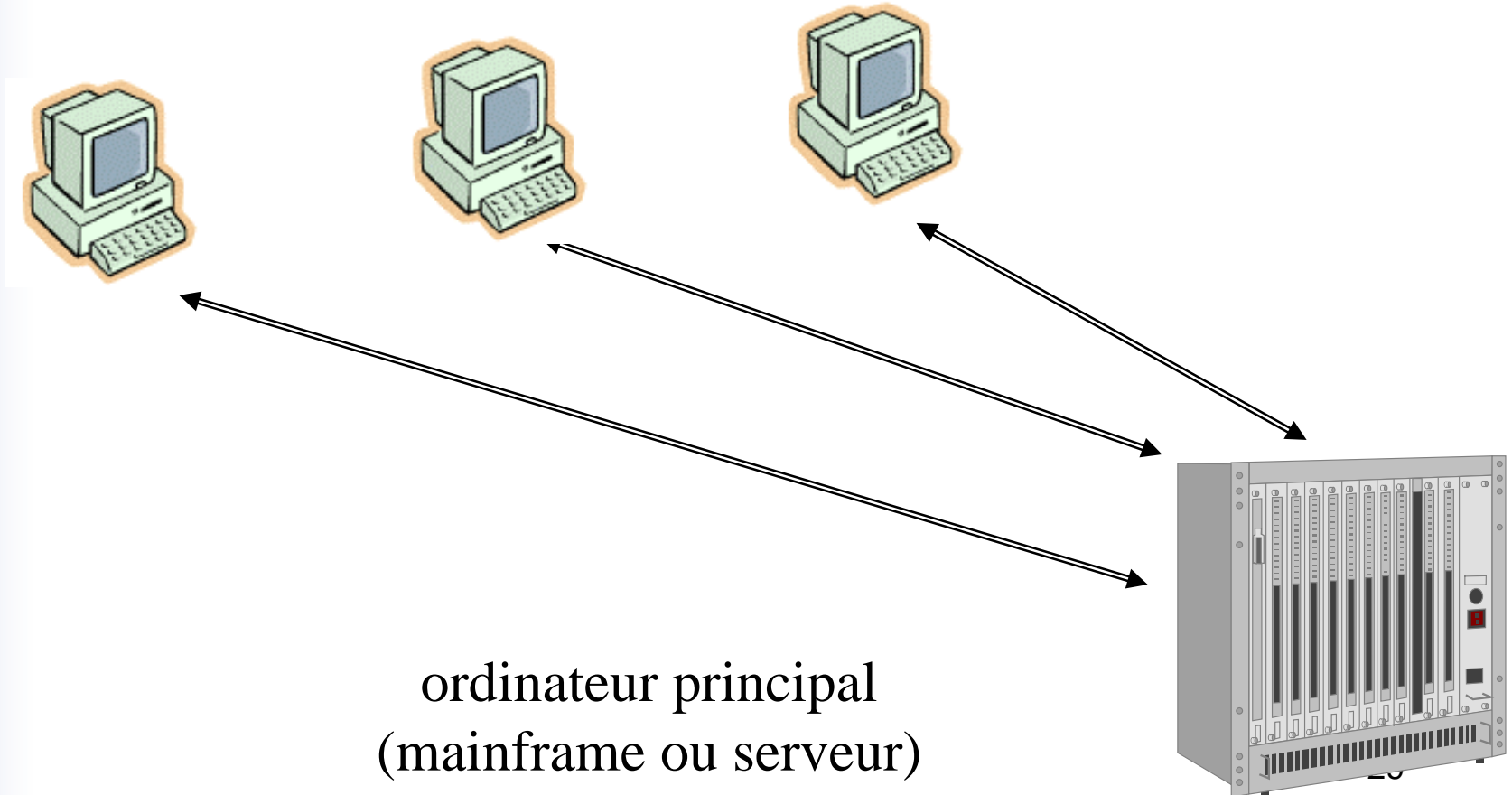
- MULTICS a été un système TSS des années 60, très sophistiqué pour son époque
- Ne réussit pas à cause de la faiblesse du matériel de son temps
- Quelques unes de ses idées furent reprises dans le système UNIX

Ordinateurs Personnels (PCs)

- Au début, les PCs étaient aussi simples que les premiers ordinateurs
- Le besoin de gérer plusieurs applications en même temps conduit à redécouvrir la multiprogrammation
- Le concept de PC isolé évolue maintenant vers le concept d 'ordinateur de réseau (*network computer*), donc extension des principes des TSS.

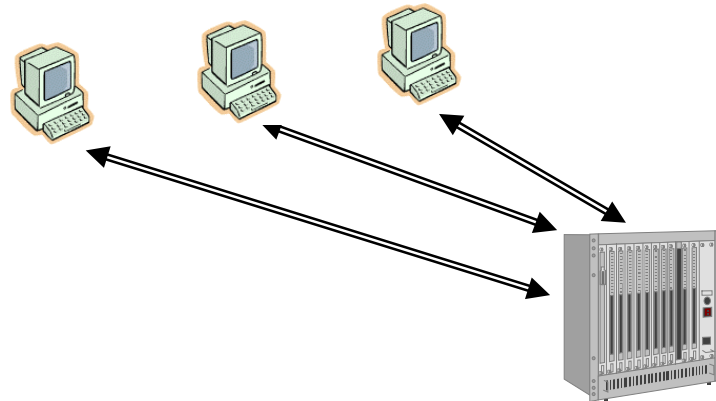
Aujourd'hui

Terminaux
'intelligents' (PCs)



Retour aux concepts de TSS

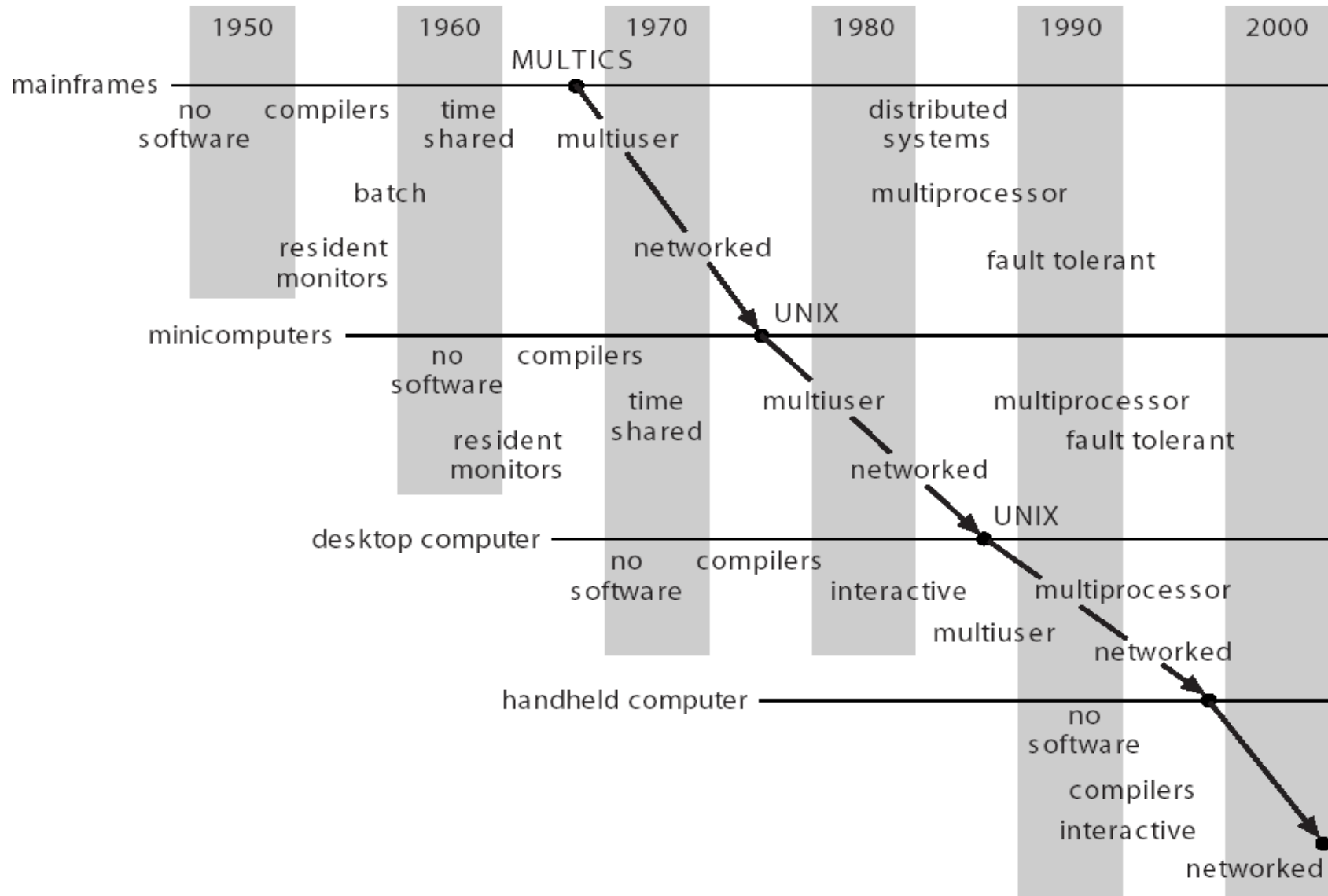
- Plusieurs PC (clients) peuvent être desservis par un ordi plus puissant (serveur) pour des services qui sont trop complexes pour eux (clients/serveurs, bases de données, telecom)
- Les grands serveurs utilisent beaucoup des concepts développés pour les systèmes TSS



Et puis...

- Systèmes d'exploitation répartis:
 - Le SE exécute à travers un ensemble de machines qui sont reliées par un réseau
 - Pas discutés dans ce cours

Évolution des SE



Une synthèse historique

Mainframes et grands serveurs

Multics et beaucoup d'autres

(1960s)

Unix

(1970)

Solaris (1995)

Linux

(1991)

Mac/OS

(1984)

Ordinateurs Personnels

MS-DOS

(1981)

Windows NT

(1988)

Windows

(1990)

Windows 2000

Windows XP

Maintenant, quelques
définitions...

Systèmes parallèles (*tightly coupled*)

- Le petit coût des puces rend possible leur composition dans systèmes multiprocesseurs
- Les ordinateurs partagent mémoire, horloge, etc.
- Avantages:
 - plus de travail fait (throughput)
 - plus fiable

Systèmes parallèles

- Symétriques
 - Tous les UCTs exécutent le même SE
 - Elles sont fonctionnellement identiques
- Asymétrique
 - Les UCTs ont des fonctionnalités différentes, par exemple il y a un *maître* et des *esclaves*.
- Aujourd'hui, tout ordinateur puissant est un système parallèle.

Systèmes distribués (= répartis)

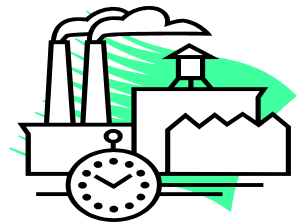
- Les réseaux d'ordinateurs sont en pleine émergence...
- Systèmes multiprocesseurs faiblement couplés (*loosely coupled*)
 - consistent d'ordinateurs autonomes, qui communiquent à travers lignes de communication

Systèmes distribués (= répartis)

- SE répartis
 - il y a un SE qui fonctionne entre ordinateurs
 - l'utilisateur voit les ressources éloignées comme si elles étaient locales
- SE en réseau (*network operating systems*) fournissent:
 - partage de fichiers (systèmes client-serveur)
 - patrons de communication (protocoles)
 - autonomie des ordinateurs

Systèmes temps réel

- Doivent réagir à ou contrôler des événements externes (p.ex. contrôler une usine). Les délais de réaction doivent être bornés
- systèmes temps réel *souples*:
 - les échéances sont importantes, mais ne sont pas critiques (p.ex. systèmes téléphoniques)
- systèmes temps réel *rigides (hard)*:
 - le échéances sont critiques (p.ex. contrôle d'une chaîne d'assemblage)



Terminologie

- Travaux 'en lots' (batch)
 - Travaux non-urgents qui sont soumis au système pour ramasser la réponse plus tard
 - Tri de fichier, calcul d'une fonction complexe, grosses impressions, sauvegarde régulière de fichiers usagers
 - Pour plus d'efficacité, peuvent être groupés et exécutés les uns après les autres
- Interactifs
 - Sont les travaux qui demandent une interaction continue avec l'ordinateur:
 - Édition de documents ou d'un programme
- Les premiers ordinateurs n'avaient pas de mécanismes de communication aisée entre usager et machine, donc normalement les travaux étaient 'par lots'

Systèmes d'exploitation classiques

- On peut diviser les systèmes d'exploitation classiques en quatre parties principales

- Les **processus**
- La **gestion de la mémoire**
- Le **système de fichiers**
- Les **entrées-sorties**

Les systèmes d'exploitation modernes intègrent par ailleurs d'autres caractéristiques:

- L'interconnexion des différentes machines et des différents systèmes par des réseaux locaux ou étendus.
- La disparition des écrans de textes et leur remplacement par des dispositifs à fenêtres multiples disposant de propriétés graphiques.

Différents systèmes

- Monde PC

- MS-DOS (MicroSoft-Disk Operating System)
- Windows 98, 2000, XP
- Windows NT (Multi-utilisateurs)

- Monde MAC

- MAC OS 9, OS 10, OS 10.2

- Monde UNIX

- SUN,... LINUX (Multi-utilisateurs)

Multi-tâches

Historique de WINDOWS

- 81 : Système d 'exploitation MS-DOS
 - Lié au PC (IBM)
 - 83 : Création de Windows
 - 85 : Windows 1.0
 - 90 : Windows 3.0
 - 92 : Windows 3.1
 - 95 : Windows 95
 - 98 : Windows 98
 - 2000 : Windows 2000
 - ... Windows XP ...
- Interface graphique pour DOS**
- SE**

L'interface utilisateur et le langage de manipulation

Historiquement 2 classes

- Les langages de commandes textuels
 - MS-DOS, UNIX
 - Chaque action est décrite par une commande textuelle obéissant à une certaine syntaxe
- Les interfaces graphiques
 - Xerox, Macintosh, IBM puis Windows
 - Manipulation directe d'objets graphiques

Exemple de SE : DOS

C:\>cd windows

C:\WINDOWS>dir *.com

commandes

Le volume dans le lecteur C est DISQUE_DUR1

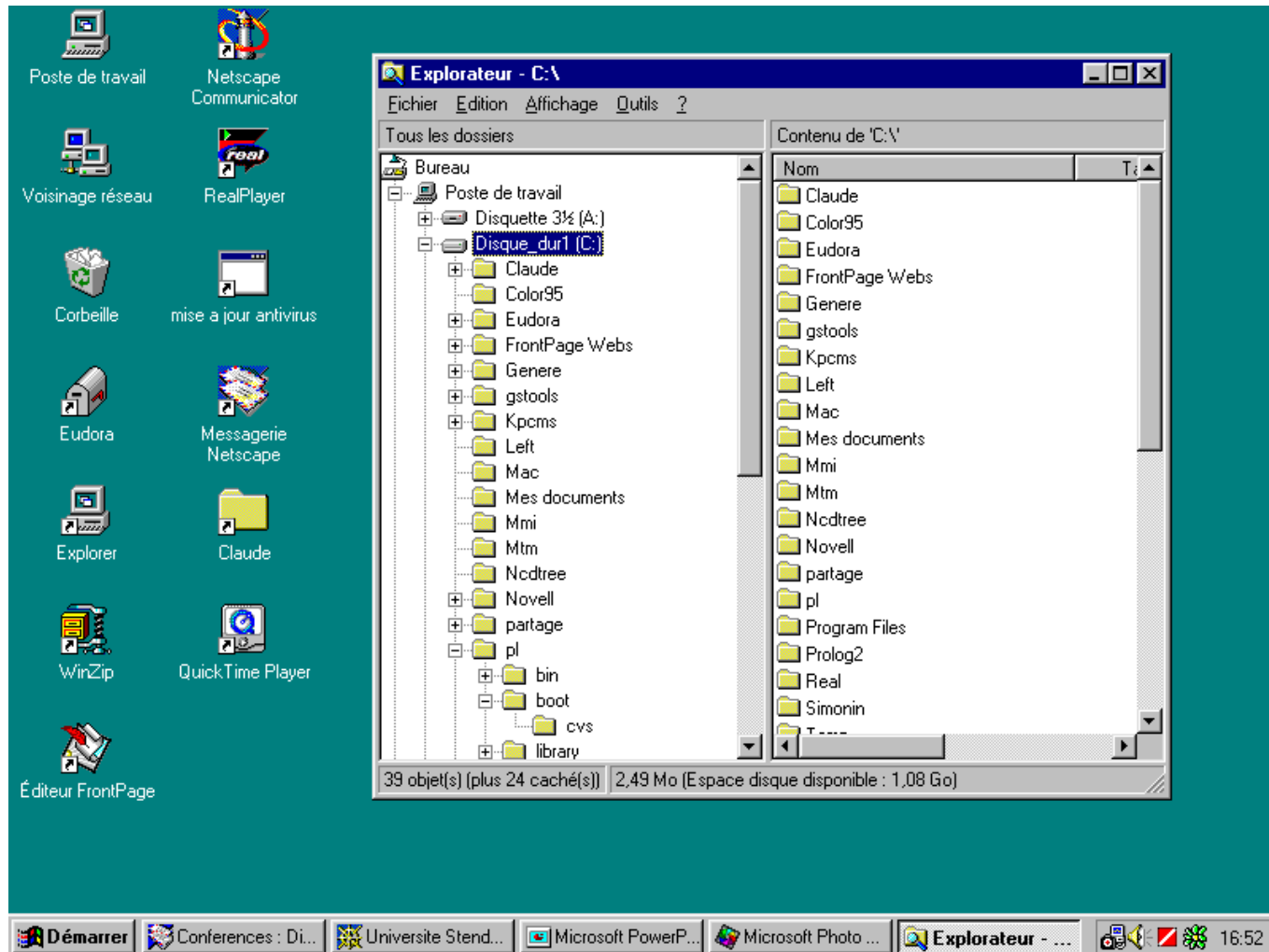
Le numéro de série du volume est 0C36-18CE

Répertoire de C:\WINDOWS

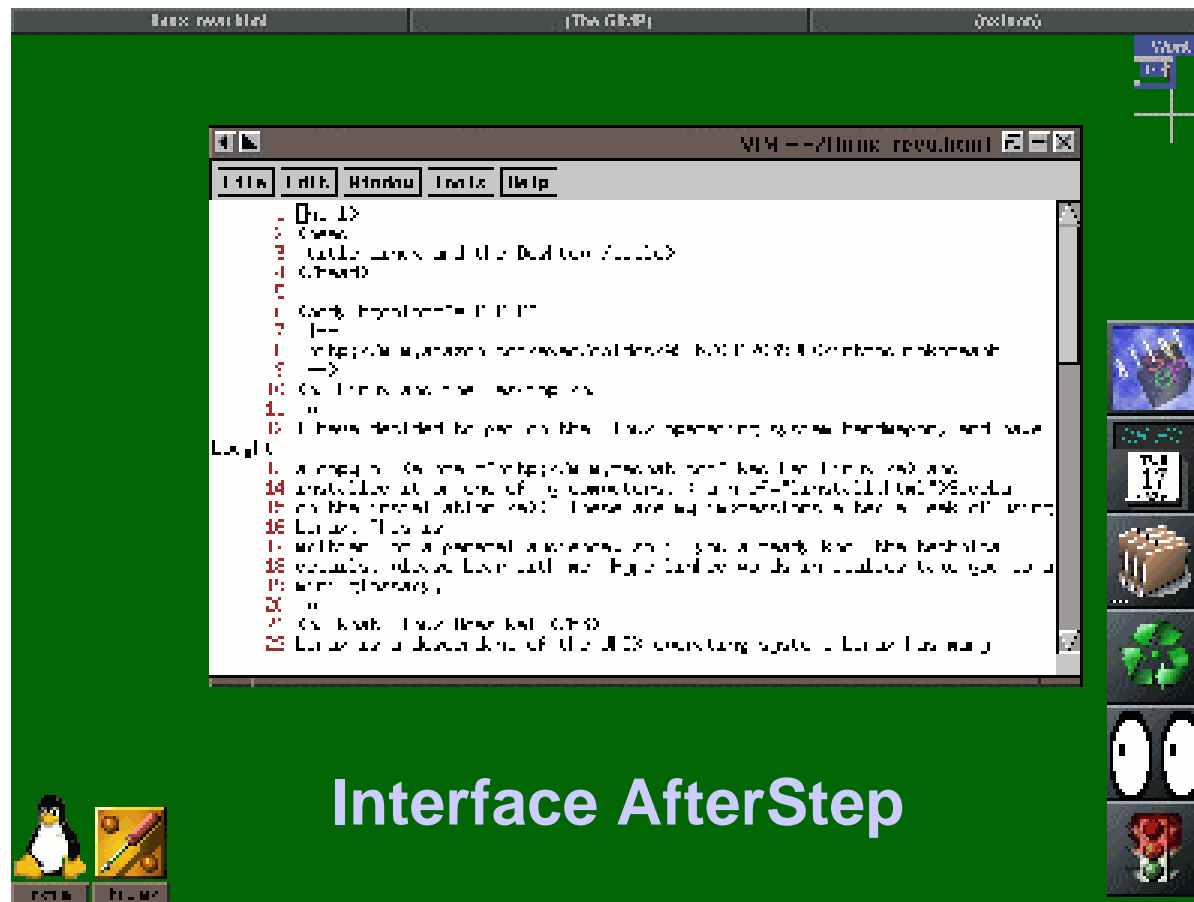
WIN	COM	25 015	24/08/96	11:11	WIN.COM
SMC9432	COM	33 267	20/05/97	13:36	SMC9432.COM
COMMAND	COM	95 910	05/05/99	14:00	command.com
	3 fichier(s)		154	192	octets
	0 répertoire(s)		70 287 360		octets libres

C:\WINDOWS>_

Exemple de SE : Windows



Exemple de SE : Linux



- Logiciel libre (programmes fournis avec leur « code source »)
- Différentes interfaces graphiques disponibles

Fichiers et de répertoires



Fichier (File)

Ensemble d'informations élémentaires stocké de manière contiguë

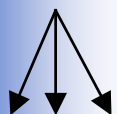
De préférence regroupés dans des répertoires

- **fichier exécutable**
Logiciel d'application (traitement de texte, tableur, etc)
- **fichier de données: Fichier de texte, feuille de calcul, base de données, etc**

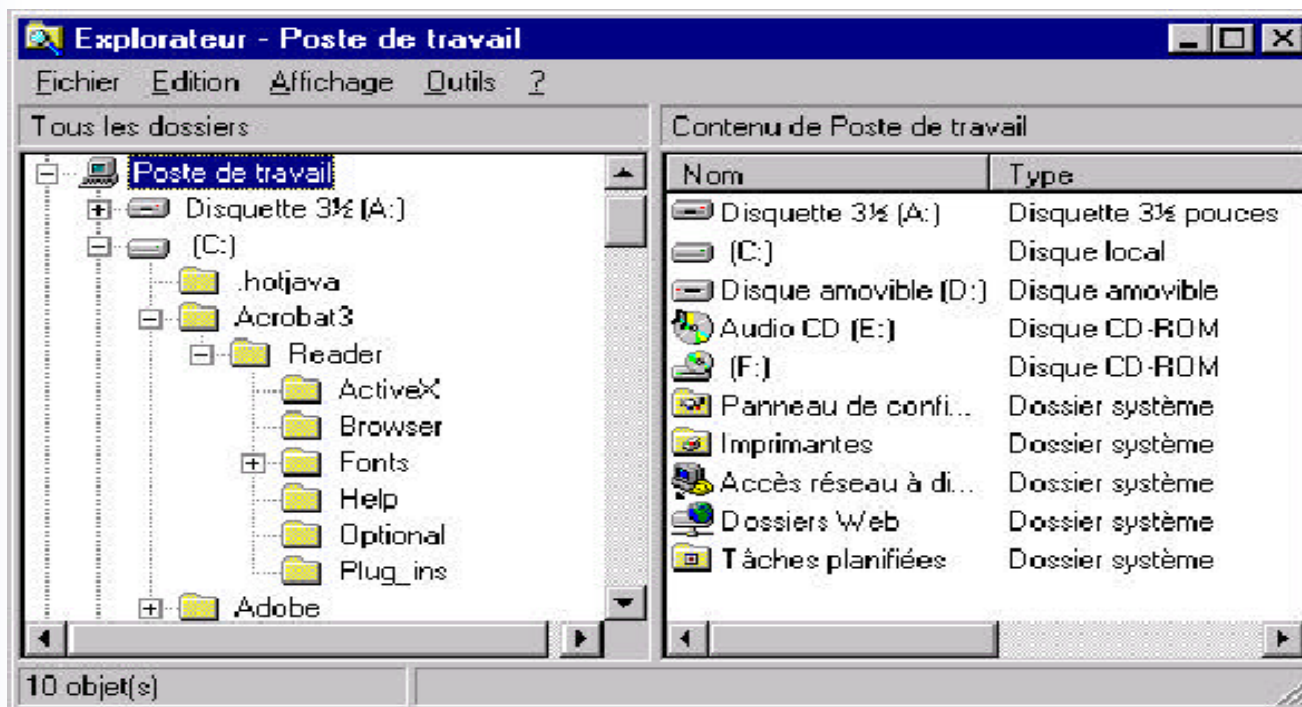


Répertoire (Directory)

Ensemble de fichiers ou d'autres répertoires (sous-répertoires)



Arborescence de fichiers et de répertoires



Structure des fichiers Windows

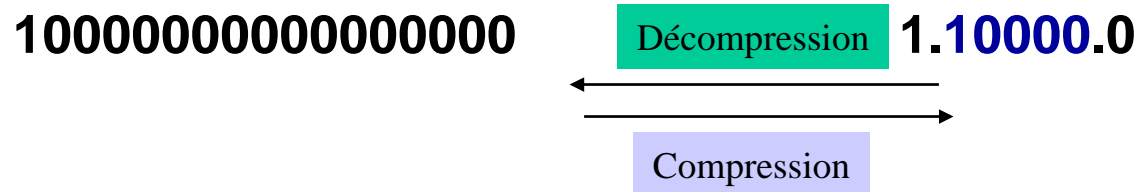
- ➡ Racine : bureau
- ➡ Répertoire : Poste de travail
- ➡ Répertoire Disques A (disquette), B, C (disque dur)

Types de fichiers

- **Nature des données**
 - Texte (Éditeurs : Bloc-Notes, Textpad et Word)
 - programmes
 - voix, image, vidéo
- **Suffixe des fichiers**
 - Texte : .txt, .doc, .html, .ps
 - Image : .gif, .jpeg
 - Vidéo : .ra, .mpeg
 - Programme : .exe

Copier et Installer un programme

- Copier un fichier .ZIP
- Décompresser le fichier (UNZIP)
100000000000000000 : 1 suivi de 16 caractères 0



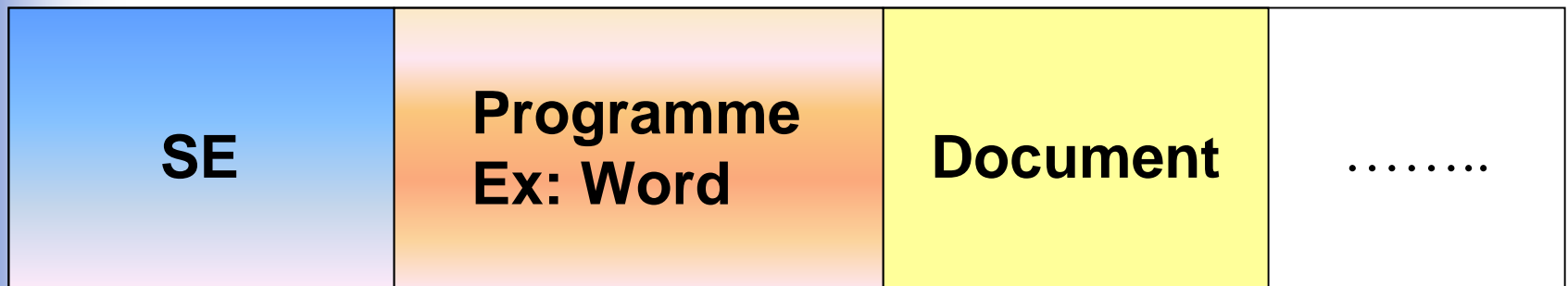
- Installation : Répertoire et sous-fichiers

Fichier, programme et document

- **Fichier** : ensemble logique d'informations stockées par l'ordinateur
- (Fichier) **Programme** : contient une suite d'instructions interprétable par l'ordinateur
- (Fichier) **Document** : contient une description interne des données traitées par un ou plusieurs programmes
- **Dossier** (répertoire) : organisation logique (pour le classement) des fichiers et des dossiers (sous-dossiers)

Programme et document

- Un document est toujours associé à un ou des programmes



**Contenu de la mémoire centrale (vive) lors
d'une session de travail sur un document**

Notion de PATH

C'est la liste des chemins que le système va regarder pour trouver un fichier lorsque on lui donne un nom

Si fich1 est une commande ; si on est sur c:

- 1) si le path est vide, taper fich1 renvoie une erreur
- 2) si le path vaut c:\users\toto, taper fich1 exécute la commande.