

**-VARI-**

# **Systemes d'exploitation**

**Catherine Coquery  
coquery@cnam.fr**

**février 2008**

# Plan de l'exposé 1

- I Système d'exploitation : première approche**
- II Services du système à l'utilisateur**
- III L'évolution historique- Introduction des concepts**

## Bibliographie

- **CARREZ C. Les systèmes informatiques coursA photocopié CNAM**
- **DELACROIX J. Cours CNAM NFA 003**
- **TANENBAUM A. Systèmes d'exploitation**  
**Pearson Education France 2003**
- **CAZES A. DELACROIX J. Architecture des machines et des**  
**Systèmes informatiques Dunod 2003**

# I Système d'exploitation : première approche

## Rôle d'un ordinateur

Traitement automatique d'informations, par exécution d'un **programme** (description du traitement) sur des **données** (des informations) pour fournir des **résultats** (de nouvelles informations).

## Fonctions fondamentales d'un ordinateur

Traitement des informations  
Mémorisation des informations  
Transmission des informations

## Informations

Programme  
Données

## Système informatique

### **Matériel**

processeurs, mémoire centrale, périphériques d'entrée/sortie, interfaces réseaux, bus de communication

### **Logiciel**

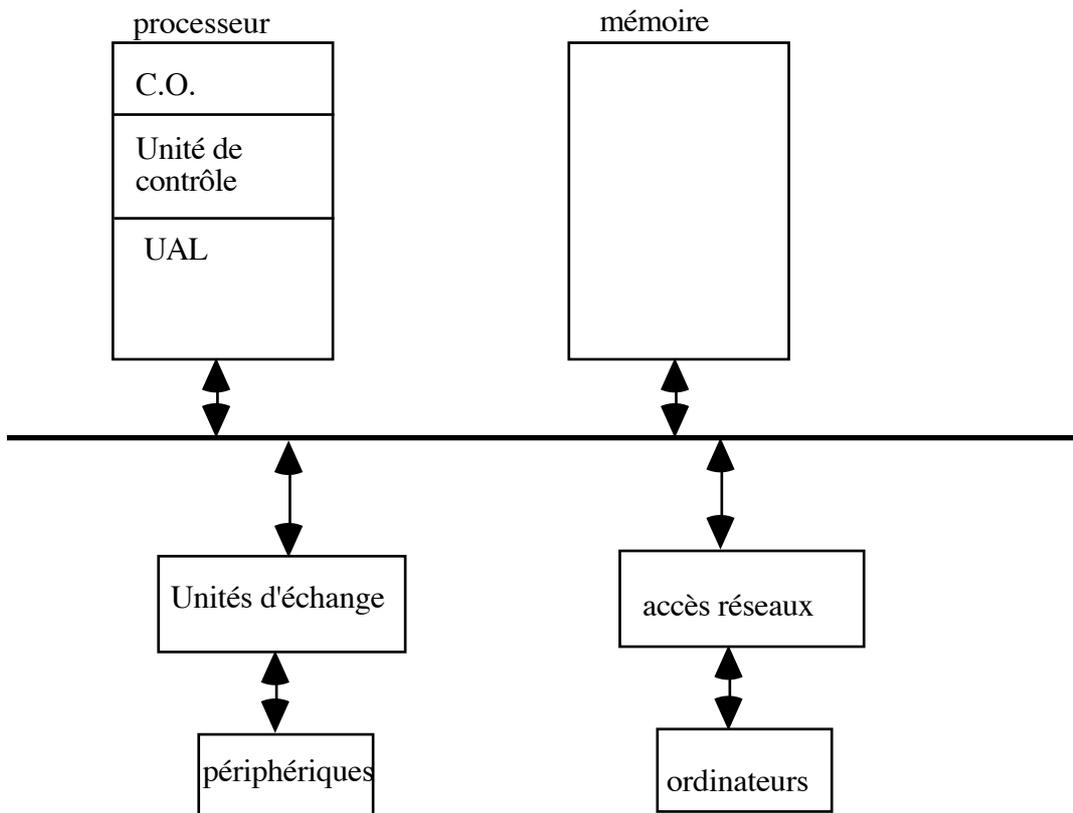
Programmes des utilisateurs

Programmes applicatifs

Programmes système d'exploitation

# Architecture matérielle

Fonctionnement : principe de Von Neumann  
Exécution séquentielle d'un programme enregistré en mémoire centrale



## Les composants

- Le **processeur** est chargé d'exécuter les instructions placées en mémoire centrale. Le processeur est cadencé par une **horloge**.
- La **mémoire centrale** contient les instructions et les données des programmes à exécuter. La mémoire **cache** permet d'accélérer l'accès à la mémoire centrale. Le **disque** permet de stocker des données. La **bande magnétique** permet d'archiver des données. Ces organes constituent **un système de hiérarchie de mémoire**.
- Les **unités d'échange** réalisent l'interface entre le processeur et les périphériques.
- Tous les composants de la machine communiquent par l'intermédiaire d'un **bus**.

## Fonctionnement du processeur

Registre CO (**compteur ordinal**) adresse de la prochaine instruction à exécuter.

### Cycle

Transfert depuis la mémoire de l'instruction à exécuter (CO) ;

Décodage de l'instruction ;

Incrémentation du CO ;

Recherche éventuelle de l'opérande en mémoire;

Exécution de l'instruction ;

Fin cycle

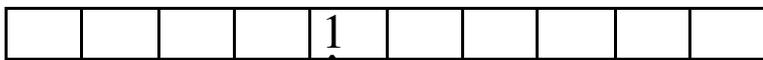
# Notion d'interruption

## Rôle

Arrêt momentané de l'exécution d'un programme pour permettre à un dispositif externe de faire un traitement spécifique.

## Mécanisme matériel

Registre interruption RIT



I cause d'interruption

Registre mot d'état programme PSW



Le cycle du processeur devient :

cycle

décoder et exécuter une instruction;

scruter RIT;

si interruption alors

sauvegarde du PSW en mémoire;

charger PSW du sous-programme d'IT;

adresse sous-programme d'IT --->CO;

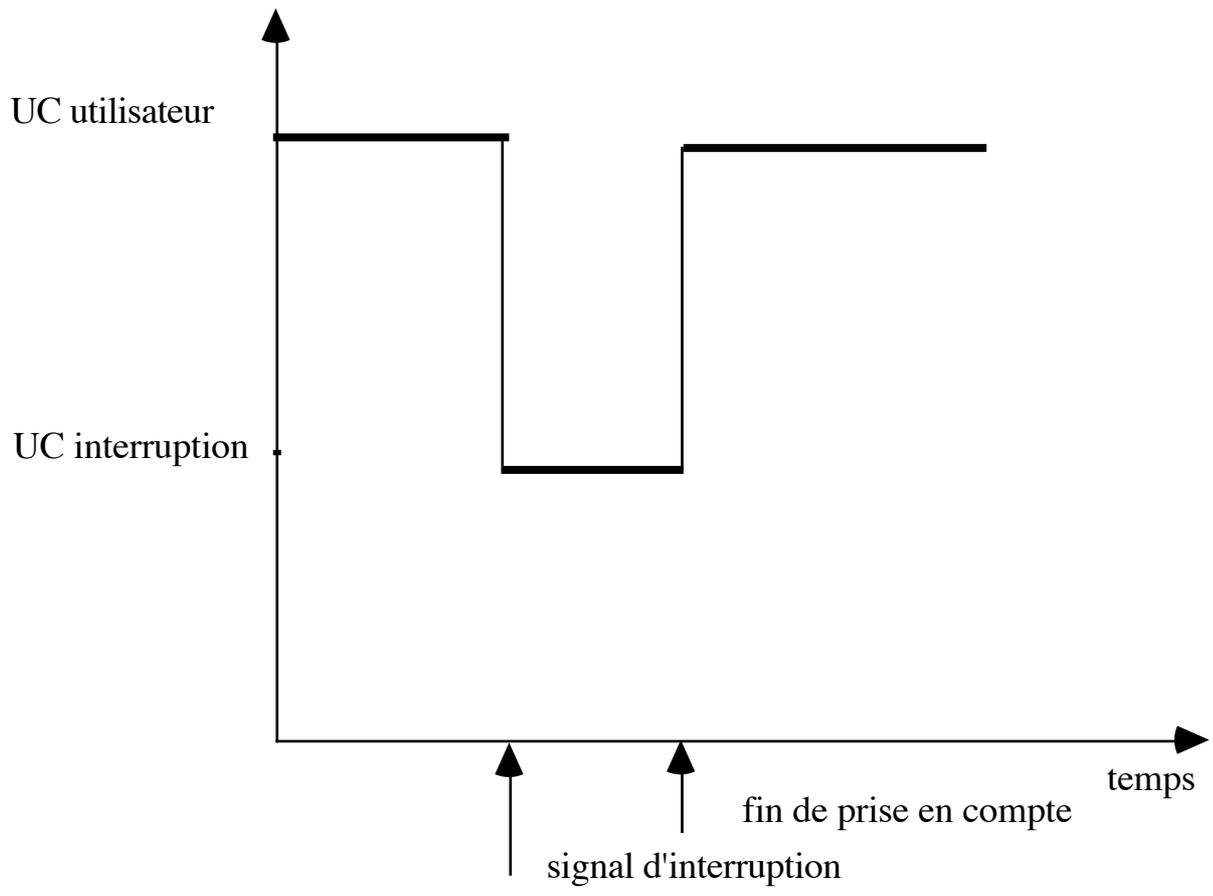
exécution du sous-programme d'interruption;

restitution du PSW;

fincycle

## Autres considérations

- Masquage des interruptions => retarder prise en compte d'une interruption
- Hiérarchie des interruptions selon priorité



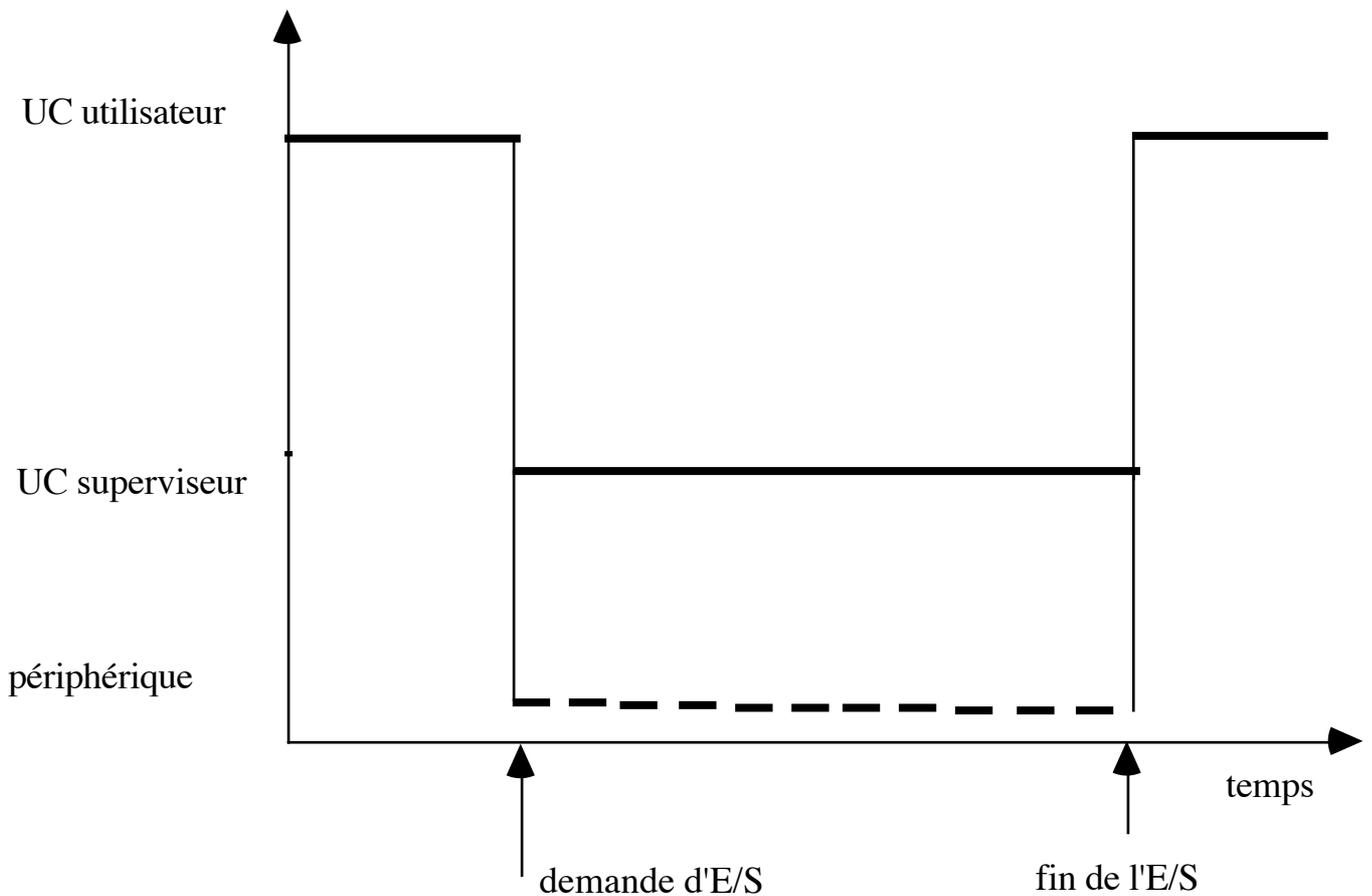
# Les entrées-sorties

Transfert d'information entre mémoire centrale et périphérique

## Les entrées-sorties programmées

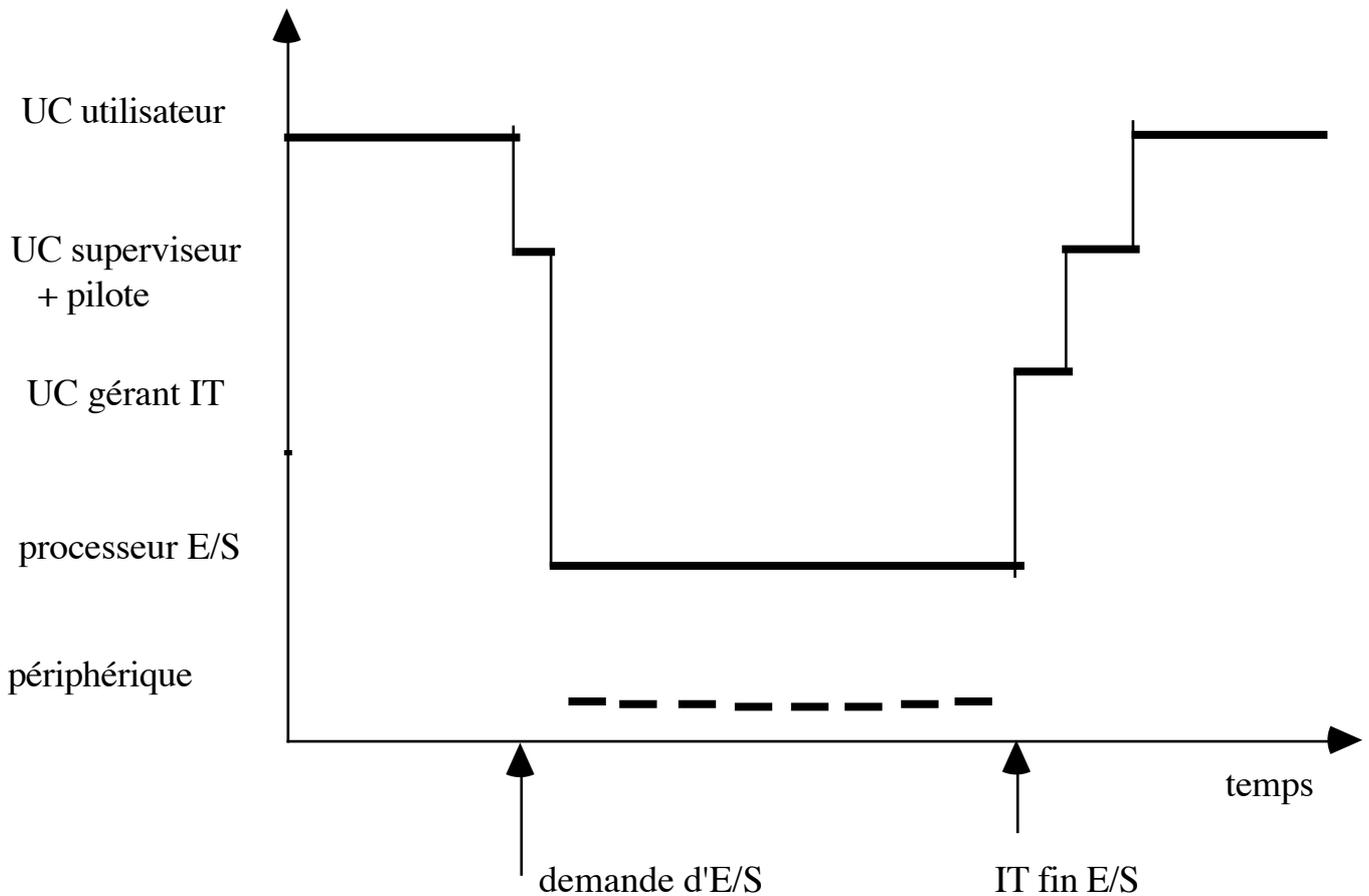
Échange sous contrôle du superviseur d'entrées- sorties

tant qu'il y a des données à lire faire  
tester l'état du périphérique;  
si périphérique non prêt alors attendre;  
lire la donnée;  
traiter la donnée;  
fait



# Les entrées sorties par DMA et interruption

**Principe** : le superviseur d'entrées-sorties(UC) initialise l'entrée-sortie, puis le dispositif d'accès direct à la mémoire (DMA) effectue l'entrée - sortie et envoie une interruption traitée par l'UC.



Pendant l'exécution de l'E/S, l'UC est disponible pour d'autres travaux ; elle peut en particulier exécuter un autre programme utilisateur.

# Notion d'appel système

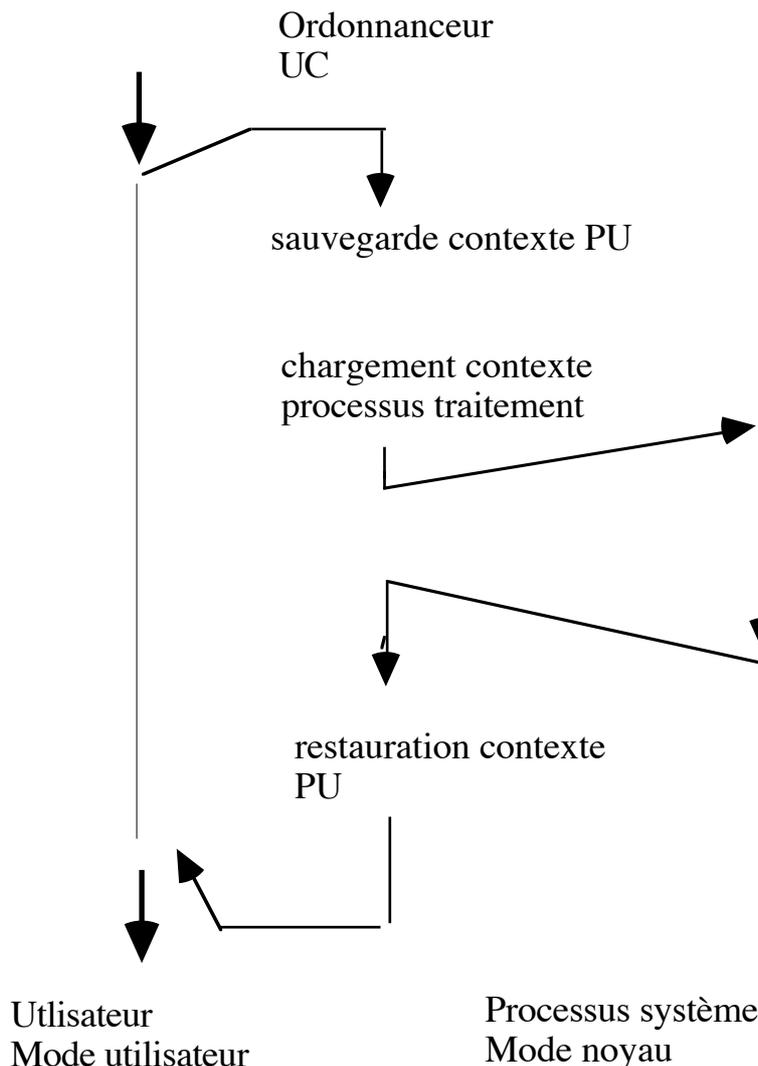
Nécessité de contrôler les actions d'un programme vis à vis de son environnement.

Deux modes de fonctionnement :

- le mode système(maître) le programme a accès à toutes les ressources;
- le mode utilisateur(esclave) le programme ne peut exécuter directement certaines instructions.

Un indicateur du mode de fonctionnement du programme est positionné dans le PSW(mot d'état programme), le processeur vérifie que l'instruction est autorisée. Si non, un sous-programme en mode système prend en charge l'exécution de l'instruction : **déroutement**

## Appel système



## Types d'appels systèmes Exemple d'UNIX

### *Manipulation de fichiers*

accès par un chemin dans l'arborescence des répertoires Unix

- périphériques /etc/ttys
- programmes systèmes /usr/bin
- fichiers utilisateurs /usr/...

appels : create, open, read, write, close etc...

### *Gestion des processus*

fork, exec, exit , kill, wait, sleep,wake-up...

### *Signaux*

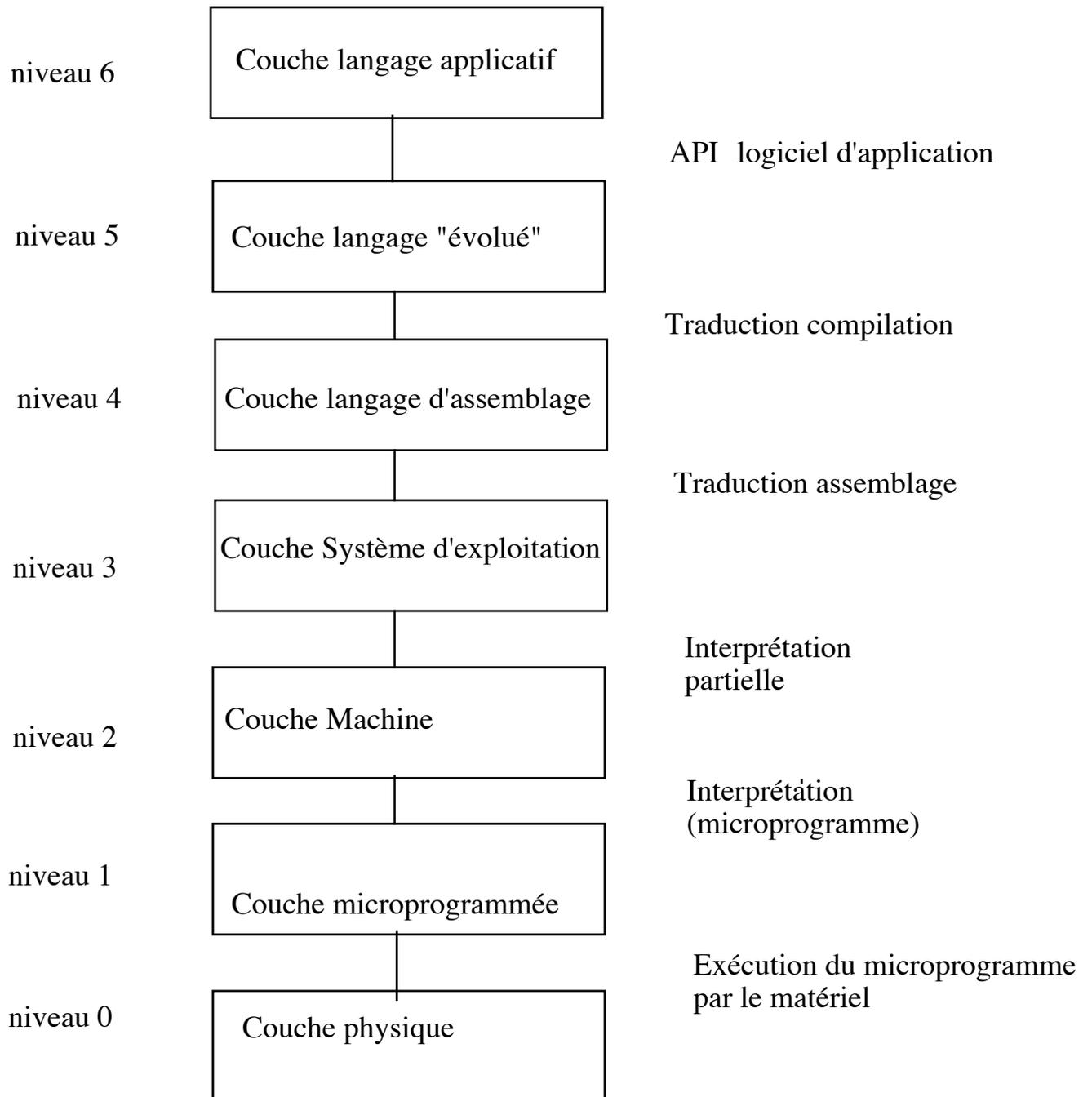
conditions exceptionnelles --> interruption logicielle

référence mémoire invalide, alarme, instruction machine illégale

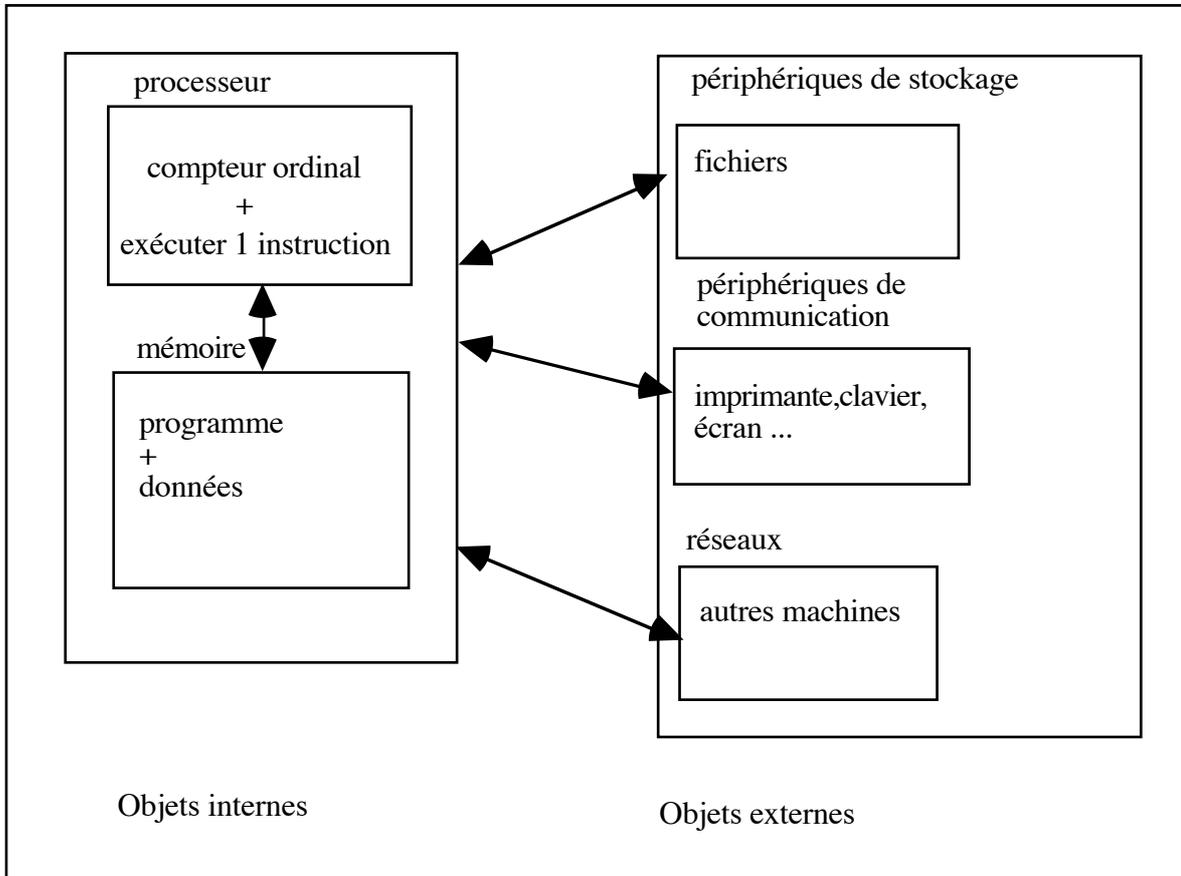
exemple ctrl Z --->SIGINT suspension du processus en cours

# Niveaux d'abstraction

A chaque niveau d'abstraction et à chaque programme correspond une machine virtuelle qui fournit l'environnement d'exécution du programme



# Le point de vue de l'utilisateur



# Rôle du système d'exploitation

Fournir une machine virtuelle à l'utilisateur qui le libère de la gestion du matériel

## **Abstraction du matériel**

Exemple : Uniformisation des mécanismes d'accès aux objets externes par le système de gestion de fichiers

**Adaptation des ressources de la machine aux besoins du programme,** en substituant par des composants logiciels, tout ou partie de composants matériels.

Exemples : processeur, mémoire virtuels

# Les services du système d'exploitation

Un système d'exploitation fournit un ensemble de services à l'utilisateur, qui le libère de la complexité du matériel.

L'utilisateur sollicite ces services par le biais d'instructions spéciales appelées **appels systèmes**.

## **Interface de l'utilisateur**

- langage de commandes
- interface graphique

## **La chaîne de production d'un programme**

- édition de texte
- compilation, assemblage
- édition de liens
- chargement en mémoire

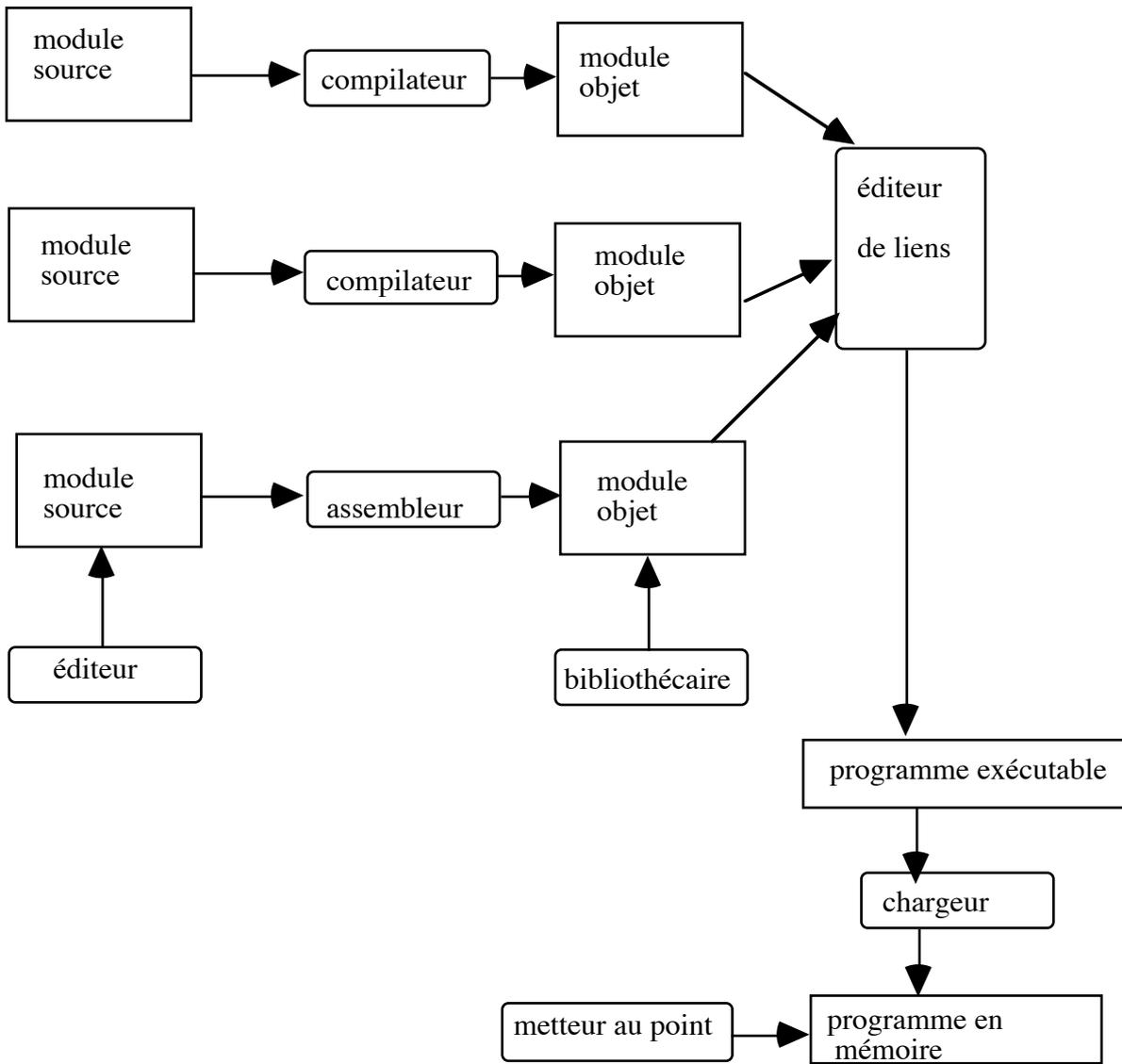
## **Exécution d'un programme**

- exécution
- erreurs
- mise au point

## **Machine virtuelle**

- Opérations d'entrées/sorties
- Gestion des fichiers
- Communication
- Gestion d'erreurs





La chaîne de production d'un programme

## Communication avec les objets externes

- Communication avec les périphériques ou les machines distantes simplifier les accès aux périphériques par des mécanismes uniformes d'accès aux objets externes.
- gérer les objets externes : conservation de l'information, désignation, protection

## Environnement physique

Fournir une machine virtuelle correspondant aux besoins des usagers : indépendance des usagers vis à vis du matériel, éventuellement par simulation des ressources manquantes

## Architecture d'un système informatique

6	utilisateurs					
5	programmes d'application		programmes utilitaires		ateliers de programmation	
4	éditeur de texte	assembleur	compilateur	éditeur de liens	chargeur	metteur au point
3	système d'exploitation					
2	gestion processeur	gestion mémoire	gestion données	gestion périphériques	gestion communications	
1	machine nue					

# Le point de vue du système

## Gestionnaire de ressources

**Ressource** : tout objet matériel ou logiciel nécessaire à l'exécution d'un programme

### **Gestion du partage des ressources entre plusieurs utilisateurs**

Le partage dans le temps

Une ressource est allouée exclusivement pendant un temps défini à un programme actif.

Exemple : la gestion du processeur dans un contexte de machine monoprocesseur

Le partage dans l'espace

Une ressource est partagée simultanément par plusieurs utilisateurs. Le système doit gérer la protection et le partage des informations.

Exemples : gestion de la mémoire centrale, du disque dur.

### **Optimisation de l'utilisation des ressources**

## Le contrôle d'exécution des programmes

Contrôle de l'utilisation des ressources (erreurs)

Gestion de la concurrence

Sécurité

# III Évolution historique des systèmes d'exploitation – Introduction des concepts

## 1. Les systèmes séquentiels (1950-1960)

### 1.1 Programmes autonomes (1950)

Matériel

- Les périphériques : lecteur, perforateur de cartes, imprimante

Interface

- Programmation en binaire

Fonctionnement

Utilisateur

Machine

Mise en route manuelle

Placer le bac de cartes dans le lecteur :

amorce logicielle  
programme exécutable

exécution de l'amorce cablée

=> lire la 1ère carte

exécution de l'amorce logicielle :  
lire carte suivante jusqu' à la fin du programme

lancement de l'exécution du programme

données

lire carte suivante

- Difficultés de la programmation en binaire

Émergence de **langages symboliques** :

Langage d'assemblage propre à la machine

Langage de haut niveau (Fortran), indépendant de la machine

Utilitaires : Compilateur, interpréteur, assembleur

## Concepts

- **Amorçage** ou bootstrap

Lors de la mise en route d'un ordinateur exécution d'un petit programme en ROM(mémoire non effaçable), qui permet de charger le noyau du système à partir d'un périphérique disquette, disque dur.

## Remarques

- enchaînement automatique des instructions (compteur ordinal)
- fonctionnement sur machine "nue"
- enregistrement en mémoire du programme à un emplacement fixe  
=> adresses fixes dans le programme : code absolu

Utilisateur

- machine virtuelle = machine physique

Gestion des ressources

- sous-emploi du matériel dû aux nombreuses manipulations humaines

## 1.2 Moniteur d'enchaînement des travaux (1955)

### - Matériel

L'émergence des supports magnétiques (bandes, puis disques) permet la conservation des programmes binaires importants (traducteurs par exemple).

### - Logiciel

Enchaînement automatique des travaux des utilisateurs pour optimiser l'utilisation de la machine : **moniteur d'enchaînement des travaux.**

### - Fonctionnement

Intermédiaire d'un opérateur pour constituer et placer un train de travaux au lecteur de cartes.

Un travail est constitué d'une suite d'étapes, chaque étape correspond à l'exécution d'un programme, utilitaire ou résultat de l'étape précédente.

```
#JOB nom
#FTN
programme à compiler
#LOAD
#RUN
données du programme
#END
```

exécution de #FTN :

chargement en mémoire du compilateur sur bande

lecture du programme utilisateur sur cartes

exécution du compilateur

stockage du résultat (programme binaire) sur bande

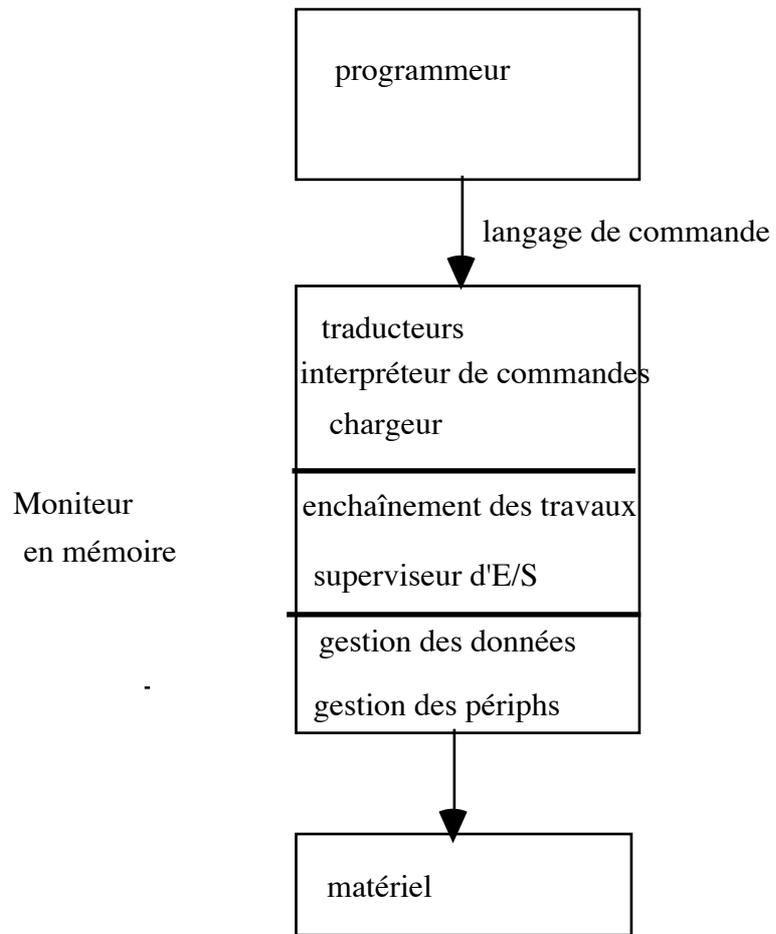
exécution de #END provoque la prise en compte du travail suivant

## Concepts

- **Moniteur** réside en mémoire centrale, protection du moniteur
- **langage de commandes** : interpréteur du langage de commandes
- **superviseur d'entrées/sorties**
  - gestion des périphériques -> **pilotes** (drivers)
  - organiser et retrouver les informations -> **système de gestion de fichiers**

## Conclusion

- Utilisateur
  - machine virtuelle -> langage de commandes
  - objets externes : périphériques, fichiers
  - pas d'accès au programme en cours => intermédiaire d'un opérateur
  - temps de réponse augmenté
- Système
  - interface avec le matériel
  - enchaînement des programmes
  - débit amélioré



### **1.3 Les ordinateurs spécialisés d'entrées sorties (1960)**

#### Matériel

- lenteur des lecteurs de cartes et de l'imprimante par rapport à l'unité centrale : réaliser les opérations d'entrée des programmes et de sortie des résultats "hors-ligne" par des ordinateurs spécifiques
- les disques plus rapides et à adressage direct.

#### Fonctionnement

Trains de travaux sur bande en entrée, trains de résultats sur bande en sortie, conservation des traducteurs et fichiers sur disque.

Moniteur d'enchaînement des travaux

#### **Conclusion**

##### - utilisateur

machine virtuelle : périphériques logiques

temps de réponse augmenté (transfert des bandes)

##### - système

augmentation du débit : parallélisme entre entrée d'un travail, sortie d'un résultat et le travail de l'UC

gestion : périphériques logiques <-> périphériques physiques

## 2 - L'introduction du parallélisme(1960-1980)

### 2.1 Les entrées-sorties tamponnées (1960)

#### Matériel

- *Dispositifs spécialisés* d'E/S autonomes  
canal d'entrées/sorties ou DMA (Dispositif d'accès direct à la mémoire)  
accès direct à la mémoire (sans passer par l'UC)  
gestion et enchaînement des entrées(sorties) élémentaires
- *Interruption* mécanisme permettant de prévenir l'UC de la fin de l'E/S . Une interruption est déclenchée par un dispositif extérieur

#### Logiciel

##### Superviseur d'E/S

###### Initialisation de l'E/S (SIO)

- tester l'état du processeur spécialisé;
- si processeur spécialisé prêt (ou DMA) alors
  - transmettre adresse du tampon de données en MC;
  - transmettre l'adresse du programme de l'E/S en MC;

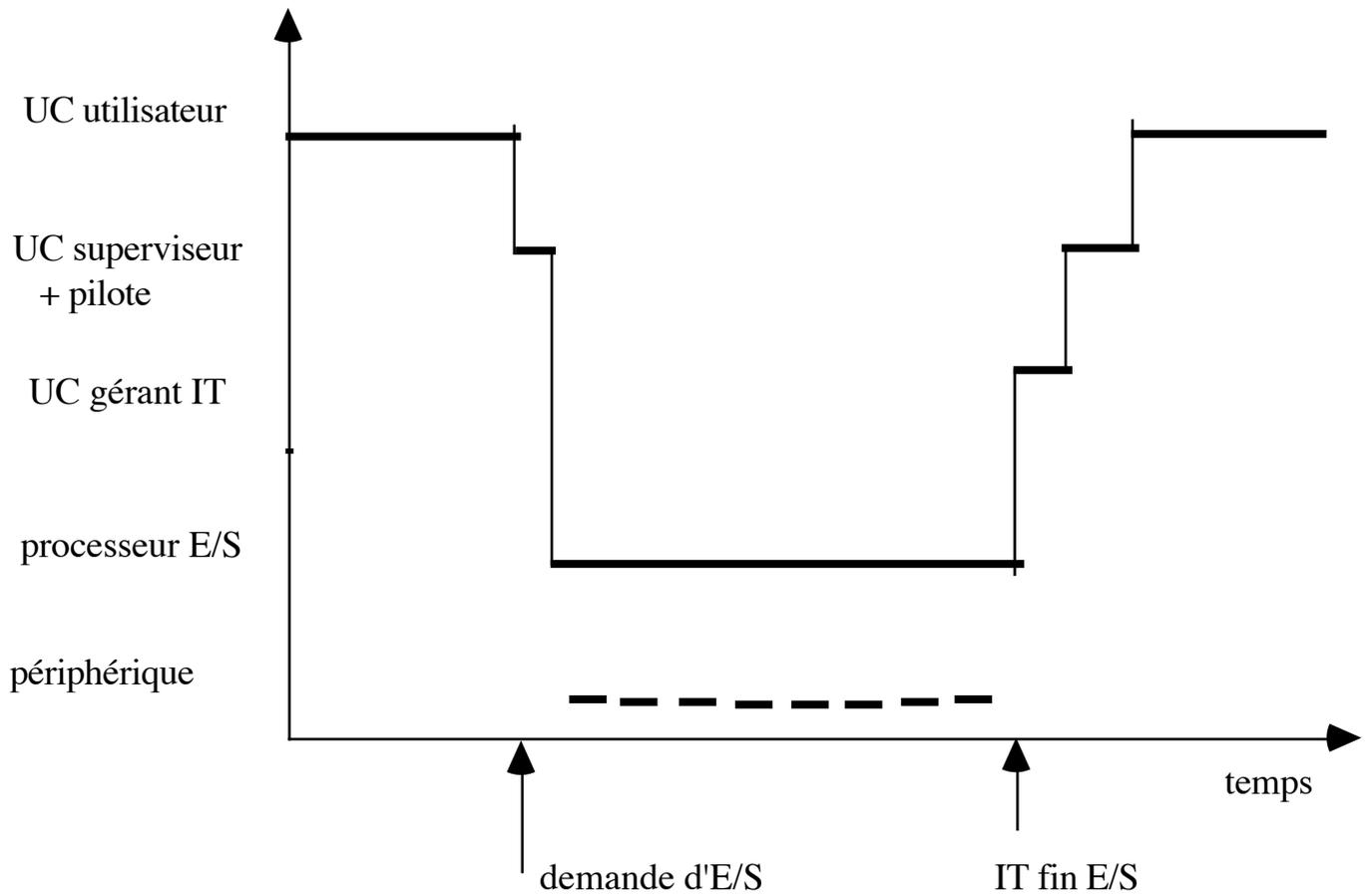
###### Fin d'E/S

- traiter les données;

#### Fonctionnement

- Le superviseur d'E/S initialise la lecture de la carte suivante vers un tampon, l'impression de la ligne suivante à partir d'un tampon
- Le processeur d'E/S effectue l'E/S de manière autonome
- Le traitement pour le programme s'exécute en parallèle, les E/S du programme sont des transferts de MC à MC.

# entrées sorties par processeur spécialisé



## Amélioration

Limitation de la taille des tampons en MC (Coût)  
meilleure utilisation de l'UC et des périphs (problème de flux)  
=>extensions sur disque, les travaux et les résultats sont stockés sur  
disque(files d'attente) mécanisme de *spooling*  
( Simultaneous Peripheral Operation on-line)

## Conclusion

Parallélisme des entrées sorties : pendant l'exécution d'une E/S, l'UC  
peut initialiser une autre E/S.

Transfert automatique des travaux et des résultats  
(plus d'intervention humaine)

Temps de réponse amélioré

Meilleure utilisation du matériel

Disque accès direct

Choix possible par le système du prochain travail à exécuter

=> **ordonnancement des travaux**

Mais système plus complexe gestion des files d'attente, gestion des  
transferts mémoire centrale disques

## 2.2 La multiprogrammation (1965)

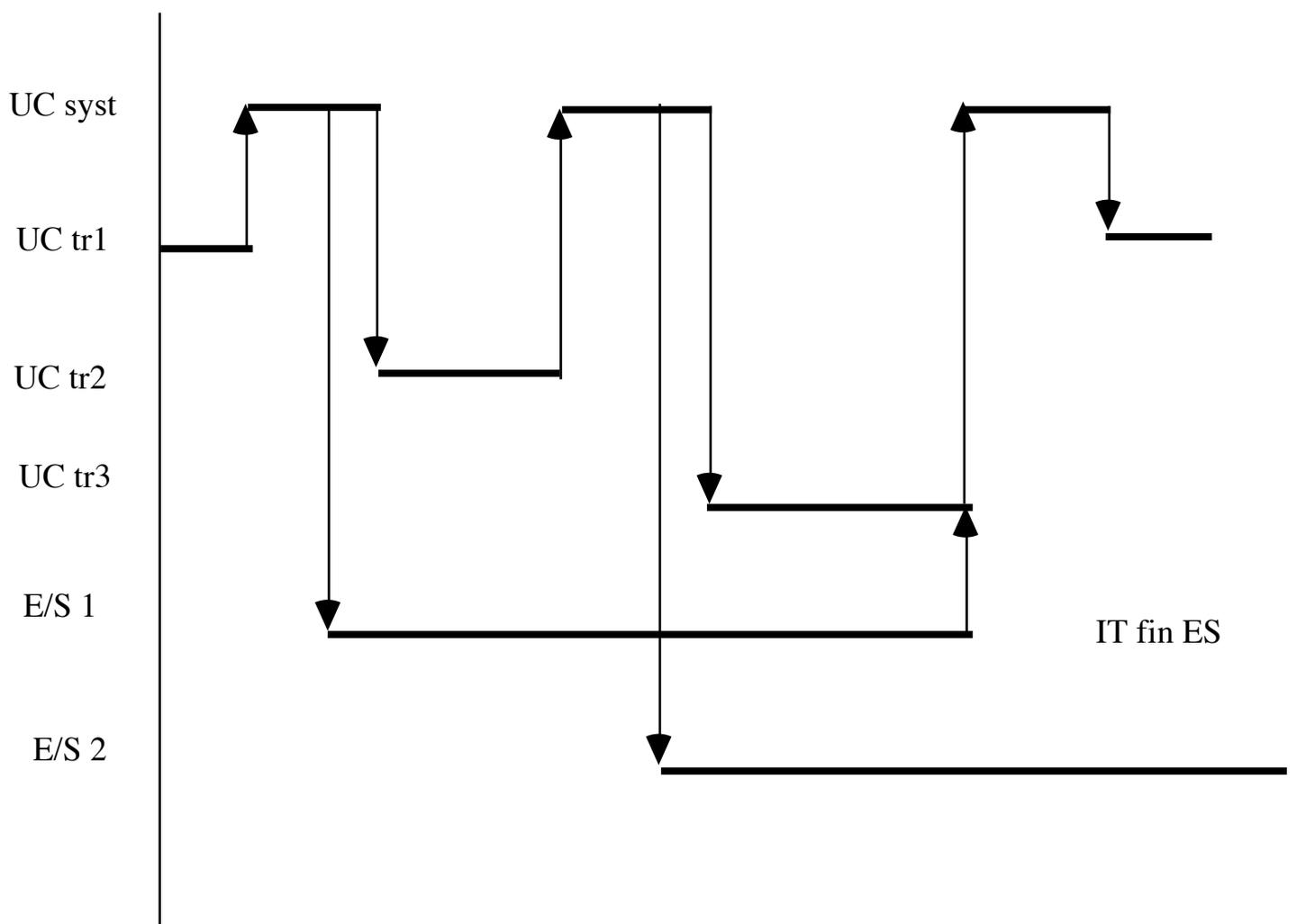
Matériel

Augmentation des tailles de MC grâce à un coût moindre

Fonctionnement

Pendant les E/S d'un travail, UC inutilisée

Plusieurs travaux simultanément en mémoire -> *multiprogrammation*



## Rôle du système d'exploitation

Gérer les *ressources*, ensemble du matériel et du logiciel, partagées entre les utilisateurs simultanés :

allocation, contrôle, restitution

processeur -> ordonnancement des travaux

mémoire -> allocation de l'espace mémoire à chaque travail, protection

périphériques - > ordonnancement des E/S des travaux

contrôle de la concurrence pour l'accès aux ressources

contrôle des interblocages

gestion de la protection

## Conclusion

Amélioration de l'utilisation des ressources

Traitement non séquentiel de la suite des travaux

Système d'exploitation complexe: utilisation par le système d'un temps UC non négligeable

Utilisateur : en général, temps de réponse augmenté

exemple le processeur n'est pas disponible à la fin d'une E/S d'un travail

### 3 Le temps partagé(1965)

Inconvénients du traitement par lots

- temps de réponse souvent important, dépend du nombre et du type de travaux soumis
- aucune interactivité du programmeur avec son programme en exécution : mise au point statique

Matériel

- **Les terminaux** permettent la communication en-ligne avec la machine  
Idée : partager le temps d'UC entre les utilisateurs  
Pendant la réflexion d'un utilisateur, un autre programme utilisateur peut s'exécuter ...  
Le temps de réponse est très amélioré et surtout il y a interactivité
- **L'horloge temps réel** : dispositif matériel externe provoquant périodiquement une interruption.  
Ces interruptions permettent de répartir le temps UC entre les programmes utilisateurs.

Evolution du langage de commandes :

- prompt
- plus de commandes, moins de paramètres :  
commandes de l'interpréteur + tout fichier exécutable

Gestion de l'environnement propre à une session d'un utilisateur et non à chaque commande

## Le problème de la mémoire

la mémoire est limitée en taille, on peut améliorer son utilisation

- **mécanisme de va et vient**

si un utilisateur est inactif pendant un certain temps (réflexion entre deux commandes), placer le programme sur disque et le rappeler ultérieurement en mémoire.

- **mémoire virtuelle**

*principe de localité* : un programme exécute pendant un certain temps un code dans le même espace d'adressage (localité temporelle) sur la même zone de données (localité spatiale)

idée ne charger en mémoire que les parties utiles du code et des données à un instant donné : *mémoire virtuelle*

- **partage de programmes**

l'utilisation d'un même programme par plusieurs utilisateurs (un compilateur par exemple) => *programmes réentrants*

Le code est invariant et partagé, on évite la copie de gros programmes pour chaque utilisateur

## Conclusion

Point de vue utilisateur

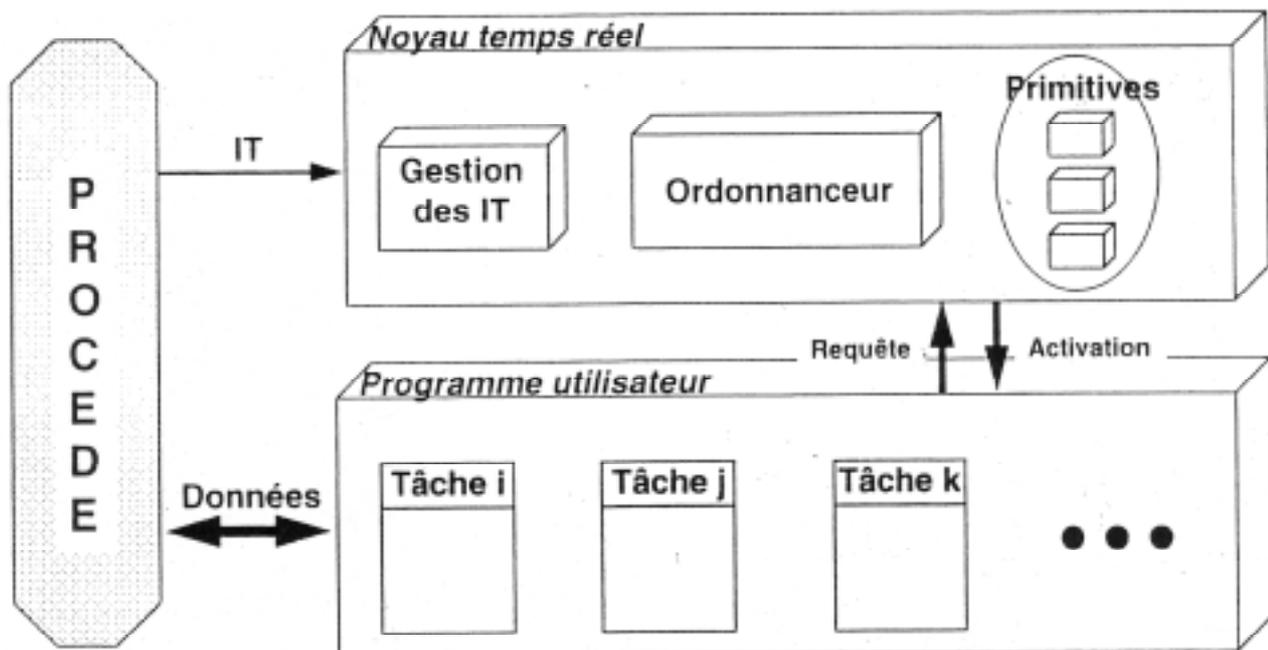
machine virtuelle processeur + ressources

Point de vue système

- gestion des terminaux par mécanisme d'interruption + superviseur
- allocation du processeur aux processus par interruption + allocateur du processeur; répartition équitable entre les utilisateurs
- la multiprogrammation permet d'optimiser l'utilisation de l'UC

## 4 Les systèmes dédiés : temps réel (1965)

Utilisé pour le contrôle de procédés industriels par exemple  
- Contrainte de temps : traitement à échéance



## 5 Les ordinateurs personnels (1970 -1990)

Moins chers, plus petits => mono-utilisateur, mono-tâche IBM PC  
Système MS DOS

Evolution dans le temps vers les mêmes fonctionnalités que les gros ordinateurs et leur système (OS/2, Windows, Windows NT)

En 1991, naissance de Linux, UNIX pour PC et Internet

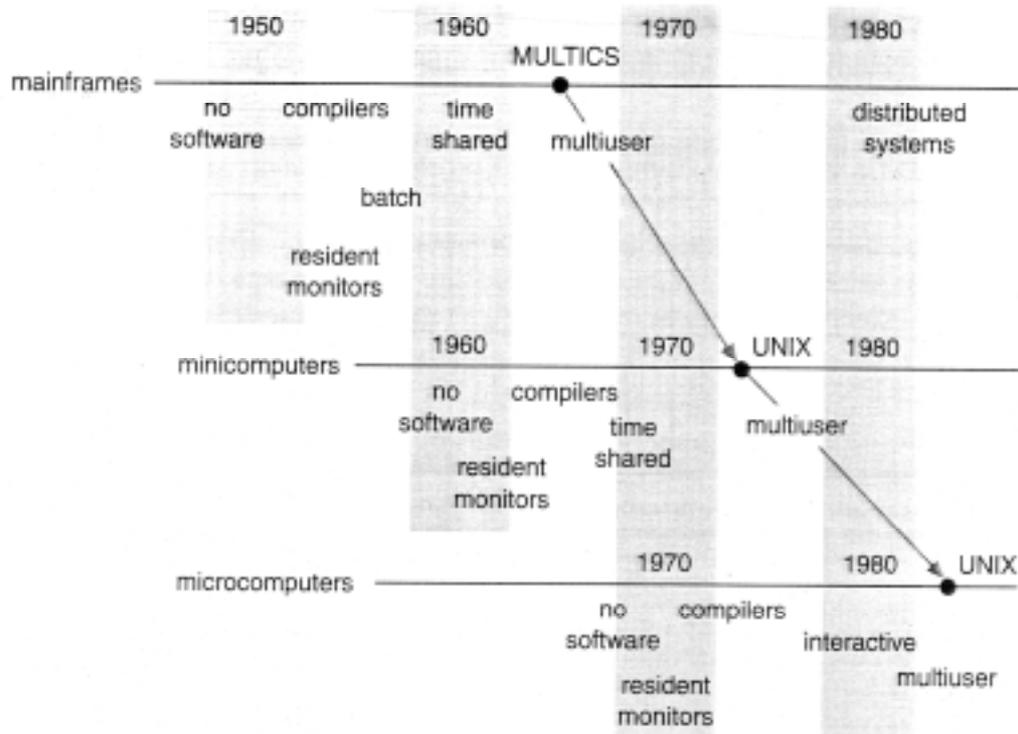


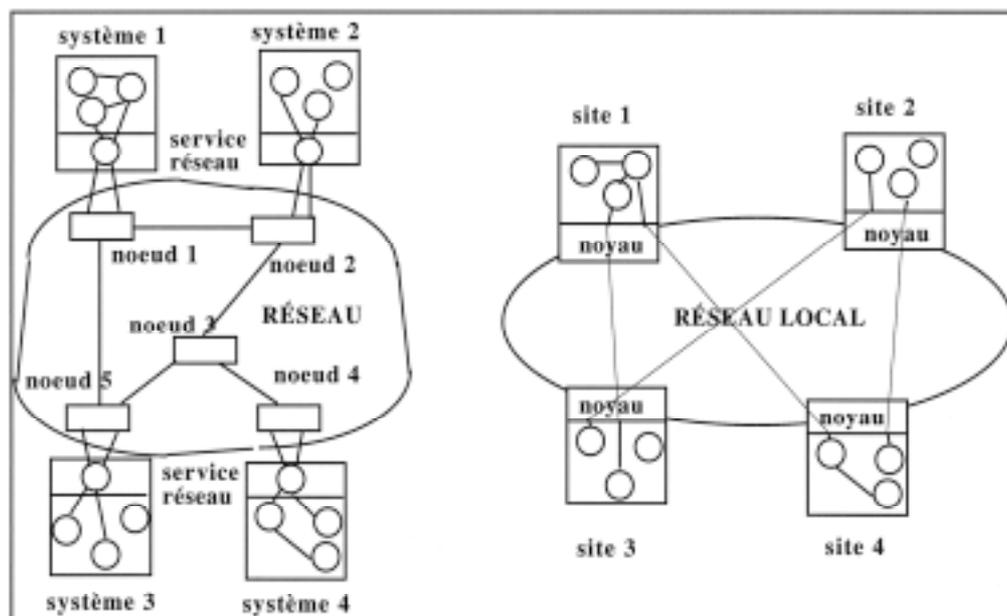
Figure 1.7 Migration of operating-system concepts and features.

## 6 Les systèmes répartis ou distribués

Sites (ordinateurs) reliés par réseau

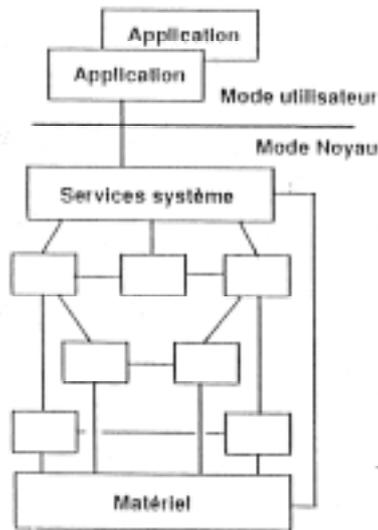
Objectifs d'un système réparti

- Abstraire le réseau pour l'utilisateur
- partager les ressources entre les différents sites
- partager la charge des sites : si un site est trop chargé, migration de processeurs vers d'autres sites, augmentation de la vitesse d'exécution
- fiabilité : en cas de panne de site, le système peut continuer à fonctionner
- communication entre programmes de différents sites : transfert de fichiers (NFS), messagerie électronique

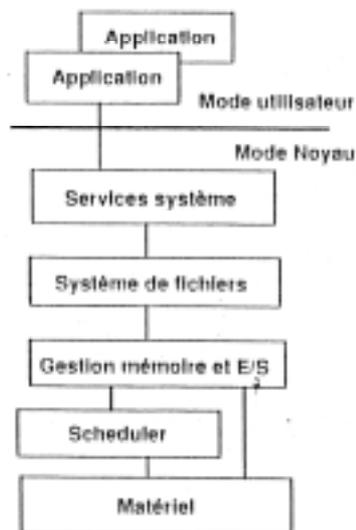


SYSTÈMES INTERCONNECTÉS <-> SYSTÈMES RÉPARTIS  
"network of operating systems" <-> "single system image"

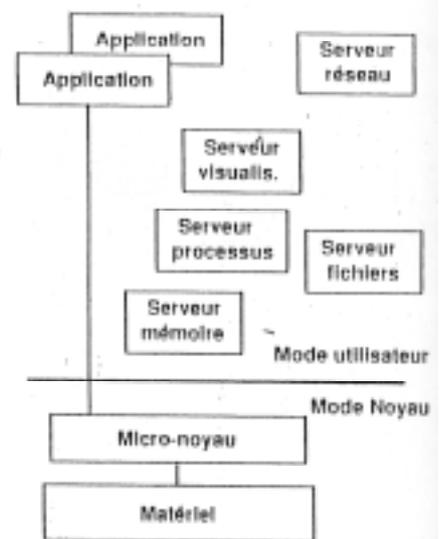
# Modèles de systèmes d'exploitation



Système d'exploitation monolithique



Système d'exploitation hiérarchisé



Système Client/serveur