



Système d'Exploitation et Outils de programmation

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

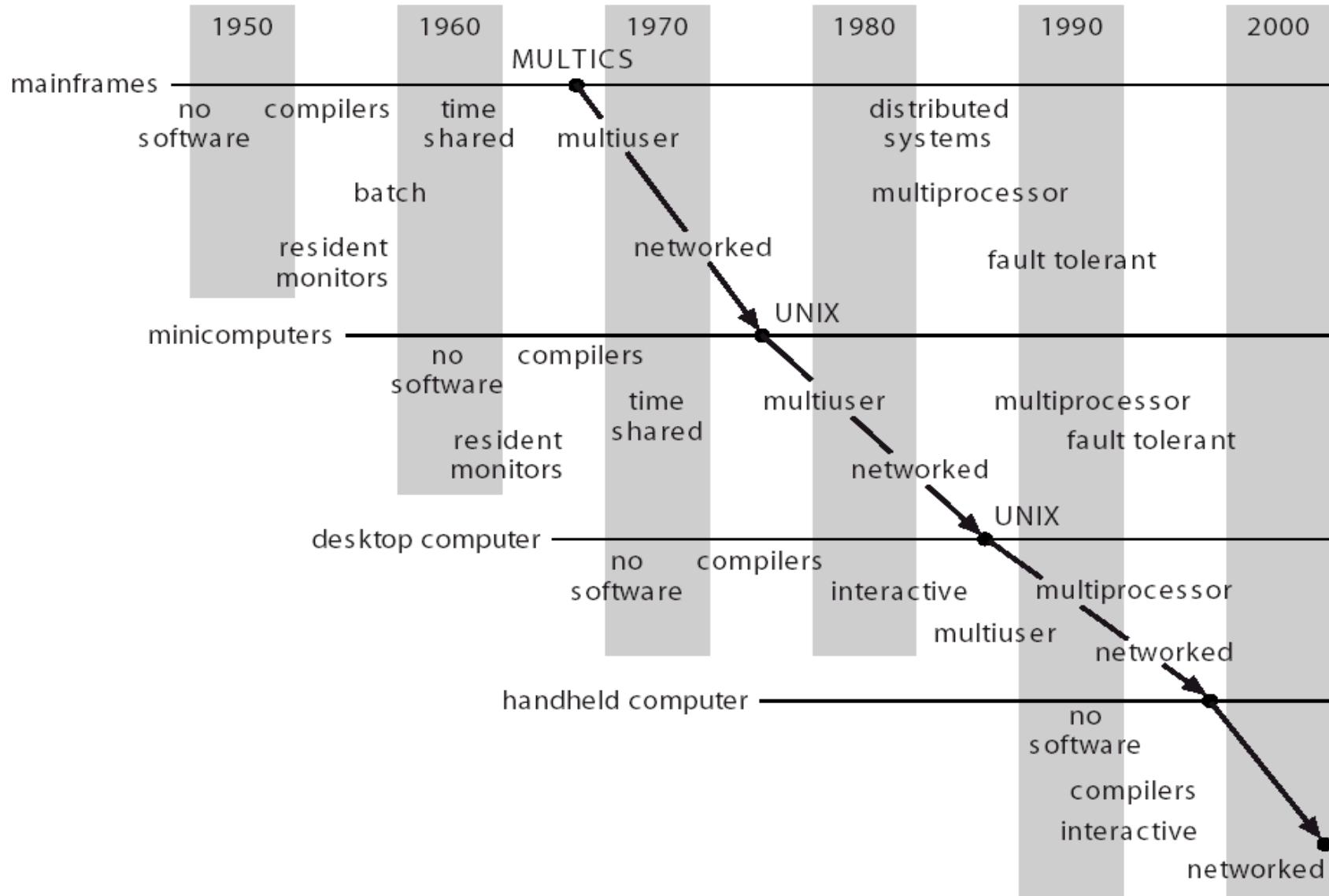
Développement de la théorie des SE

- 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'usagers!
- 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)

Linux
(1991)

Mac/OS
(1984)

Windows NT
(1988)

Windows 2000
Windows XP

Ordinateurs Personnels

MS-DOS
(1981)

Windows
(1990)

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

Systèmes 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'usager 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**

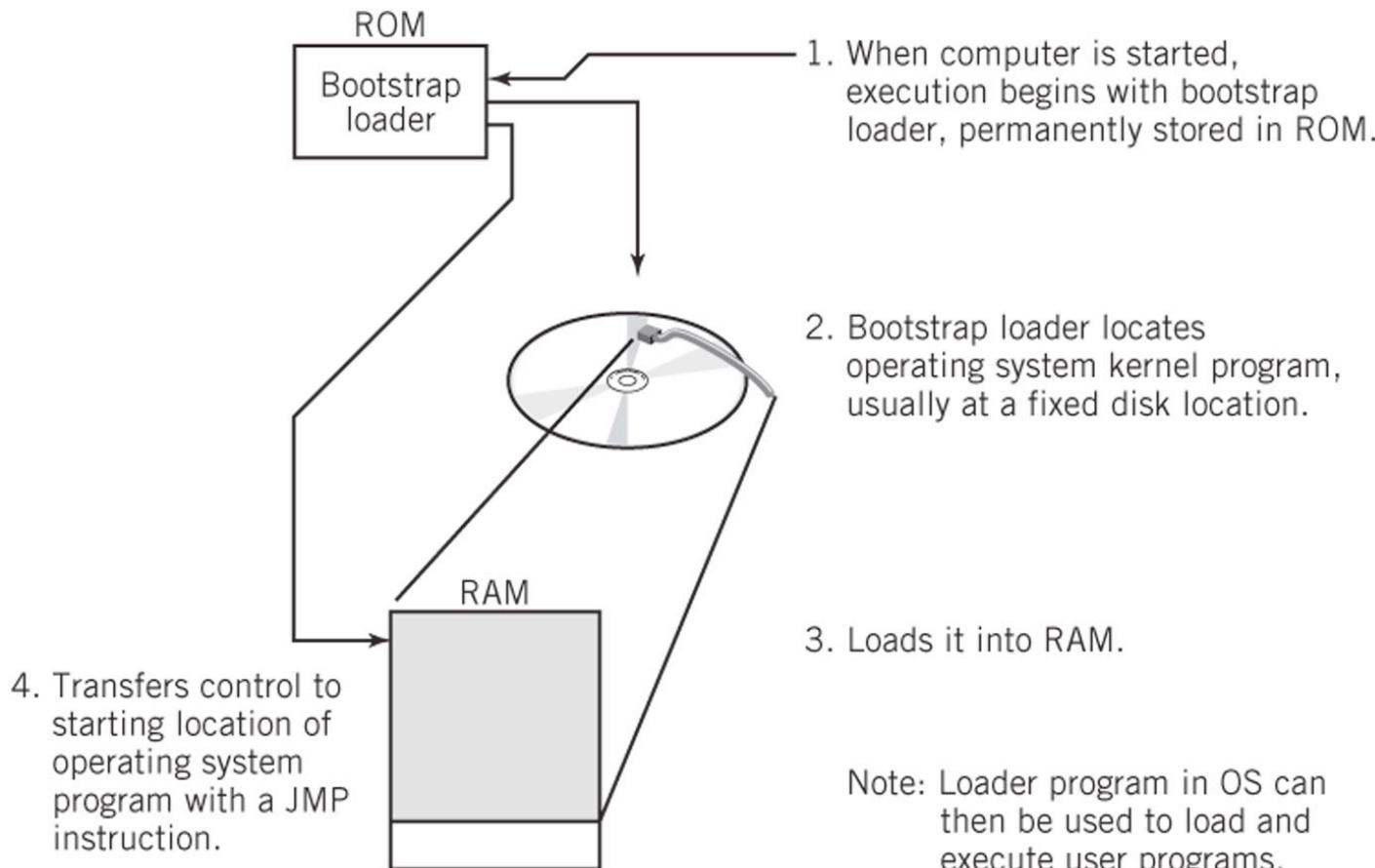
SE

- 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

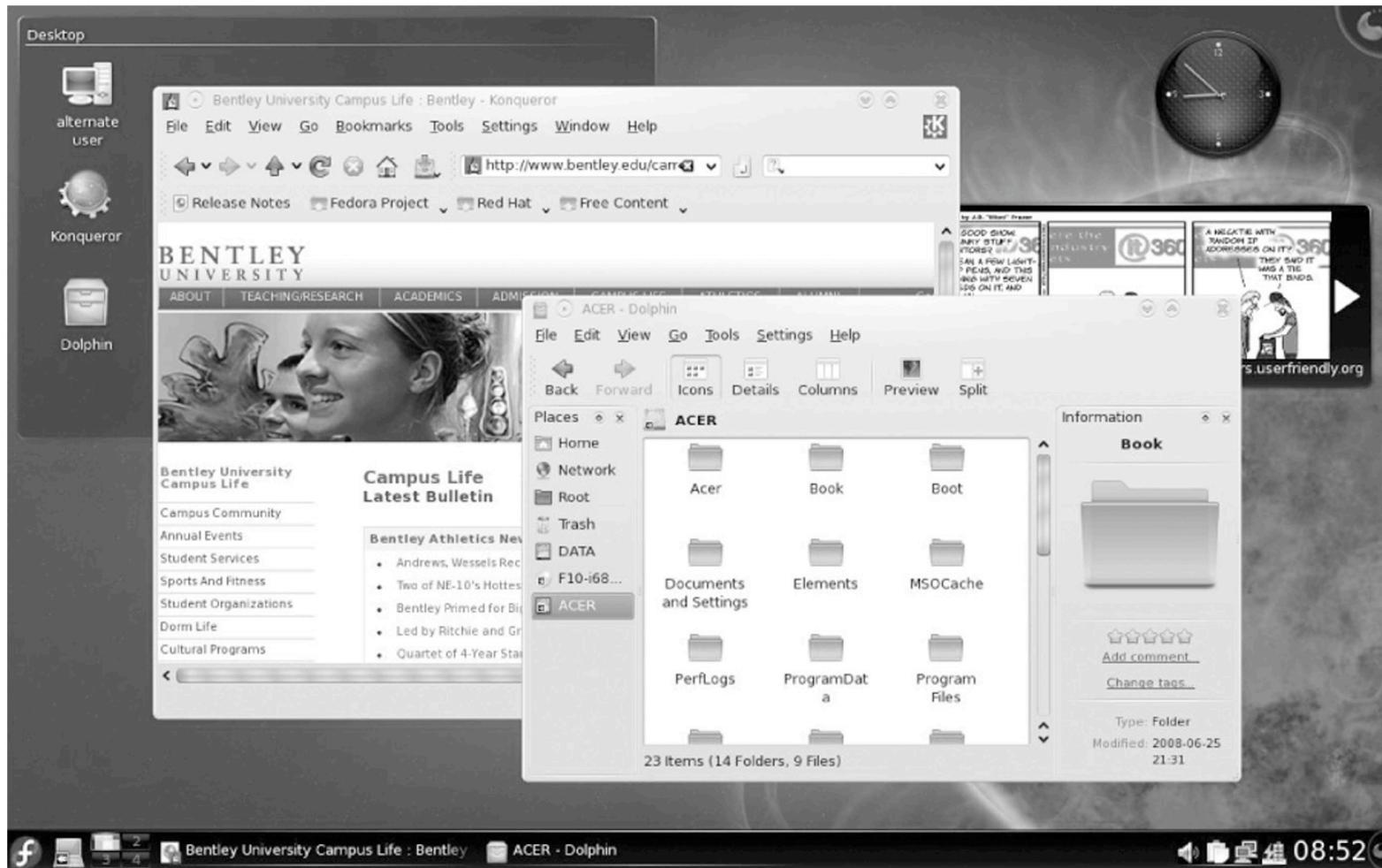
Langage de commandes

- 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

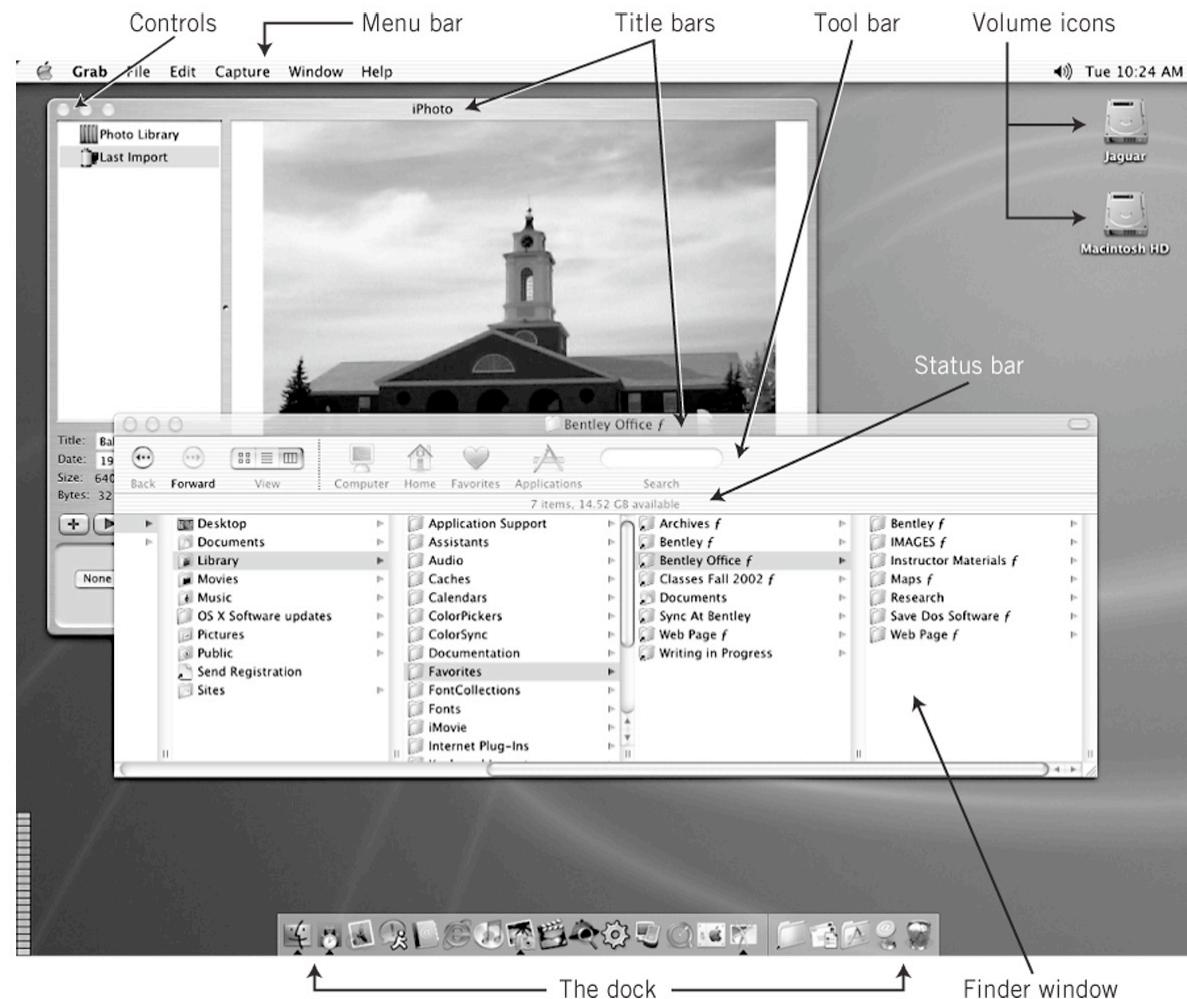
GUI Interface – Windows Vista



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
 - Requiert 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

Gestion des fichiers

- Allocation des mémoires de masse ainsi que l'accès aux données stockées (système de gestion de fichiers et notion de fichier)
- Assure la conservation des données sur un support de masse non volatile

Système de gestion de fichiers

- Offre à l'utilisateur une unité de stockage indépendante des propriétés physiques des supports de conservation: le fichier
 - Fichier logique (vue de l'utilisateur)
 - Fichier physique
- Assure la correspondance entre le fichier logique et le fichier physique
 - Structure de répertoire

Système de gestion de fichiers

- Le fichier logique
 - Un type de données standard défini dans les langages de programmation sur lequel un certain nombre d'opérations peuvent être réalisées
 - Création, ouverture, fermeture, destruction
 - Les opérations de création ou d'ouverture effectuent la liaison du fichier logique avec le fichier physique
 - Un ensemble d'enregistrements, un type de données regroupant des données de type divers liées entre elles par une certaine sémantique inhérente au programme qui les manipule

Système de gestion de fichiers

- Fichier physique
 - Correspond à l'entité allouée sur le support permanent et contient physiquement les enregistrements définis dans le fichier logique
 - Le fichier physique est constitué d'un ensemble de blocs physiques qui doivent être alloués au fichier logiques

Fichier physique

- Différentes méthodes d'allocation de la mémoire secondaire
 - Allocation contiguë
 - Allocation par zones
 - Allocation par blocs chaînés
- Pour pouvoir allouer des blocs aux fichiers il faut connaître à tout moment l'ensemble des blocs libres et donc gérer l'espace libre sur le disque
 - **Liste d'espace libre**

Exemple

- Allocation par blocs chaînes (DOS/Windows)
 - L'ensemble des chaînages des blocs de fichiers est regroupé dans une table FAT(File Allocation Table)
 - Nombre d'entrées = Nombre de blocs de données sur le disque
 - Chaque entrée de la table correspond à un bloc du disque et contient:
 - **Si le bloc appartient à un fichier et n'est pas le dernier bloc de ce fichier => le numéro du bloc suivant**
 - **Si le bloc appartient à un fichier et est le dernier bloc de ce fichier => une valeur de fin de fichier**
 - **Si le bloc n'appartient pas à un fichier => une valeur de bloc libre**

Répertoire

- Correspondance fichier logique – fichier physique
 - Effectue par le biais d'une table appelée répertoire qui contient des informations de gestion des fichiers
 - Le nom logique du fichier
 - Le type du fichier
 - Codé dans son nom logique à l'aide d'une extension
 - L'adresse physique du fichier
 - Dépend de la méthode d'allocation mise en œuvre sur le disque
 - La taille en octets ou en blocs du fichier
 - Le nom du propriétaire
 - Les protections appliquées au fichier

Répertoire

- Le système de gestion de fichiers offre des primitives permettant de manipuler les répertoires
- Les différentes structures de répertoires existantes se distinguent par le nombre de niveaux
 - Structure en arbre est composée d'un répertoire initial (la racine) et d'un ensemble de nœuds constitués par l'ensemble de sous-répertoires et d'un ensemble de feuilles qui sont les fichiers eux-mêmes
- Le nom complet d'un fichier (path name) est constitué de son nom précédé du chemin dans la structure de répertoires depuis la racine

Partitions

- Gérer des milliers de fichiers dans un seul ensemble – difficile
- Solution – diviser l'ensemble du système de gestion de fichiers en morceaux indépendants – partitions
 - Partition constitue un disque virtuel auquel est associé un répertoire qui référence l'ensemble des fichiers présents sur la partition
 - Chaque partition est repérée par un nom – label
 - La partition doit être connectée à l'arborescence de fichiers de la machine

Services et facilités

- Gestion du processeur
 - Allocation du processeur aux différents programmes: ordonnancement
 - Selon le type de SE l'algorithme d'ordonnancement répond à des objectifs différents
- Gestion de la concurrence
 - Communication entre plusieurs programmes, synchronisation de l'accès aux données partagées (outil de communication et de synchronisation entre programmes)

Notion de Processus

- Définitions
- États d'un processus
- Bloc de contrôle du processus
- Opérations sur les processus

Notion de Processus: Définitions

- Un programme en cours d'exécution auquel est associé un environnement processeur (PC, registres généraux, ...) et un environnement mémoire (zone de code, de données et de pile) appelés contexte du processus
- Instance dynamique d'un programme et incarne le fil d'exécution de celui-ci dans un espace d'adressage protégé

Notion de Processus: États d'un processus

- Au fur et à mesure qu'un processus exécute, il est caractérisé par un état
 - Lorsque le processus obtient le processeur et s'exécute, il est dans l'état **élu**. L'état **élu** est l'état d'exécution du processus
 - Lors de l'exécution, le processus peut demander à accéder à une ressource. Il quitte alors le processeur et passe dans l'état **bloqué**. L'état **bloqué** est l'état d'attente d'une ressource autre que le processeur

Élu

Bloqué

Notion de Processus: États d'un processus

- Lorsque le processus est passé dans l'état bloqué, le processeur a été alloué à un autre processus. Le processeur n'est donc pas forcément libre. Le processus passe dans l'état **prêt**. L'état **prêt** est l'état d'attente du processeur.

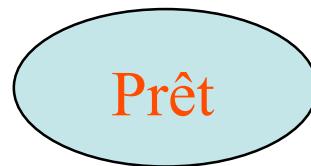
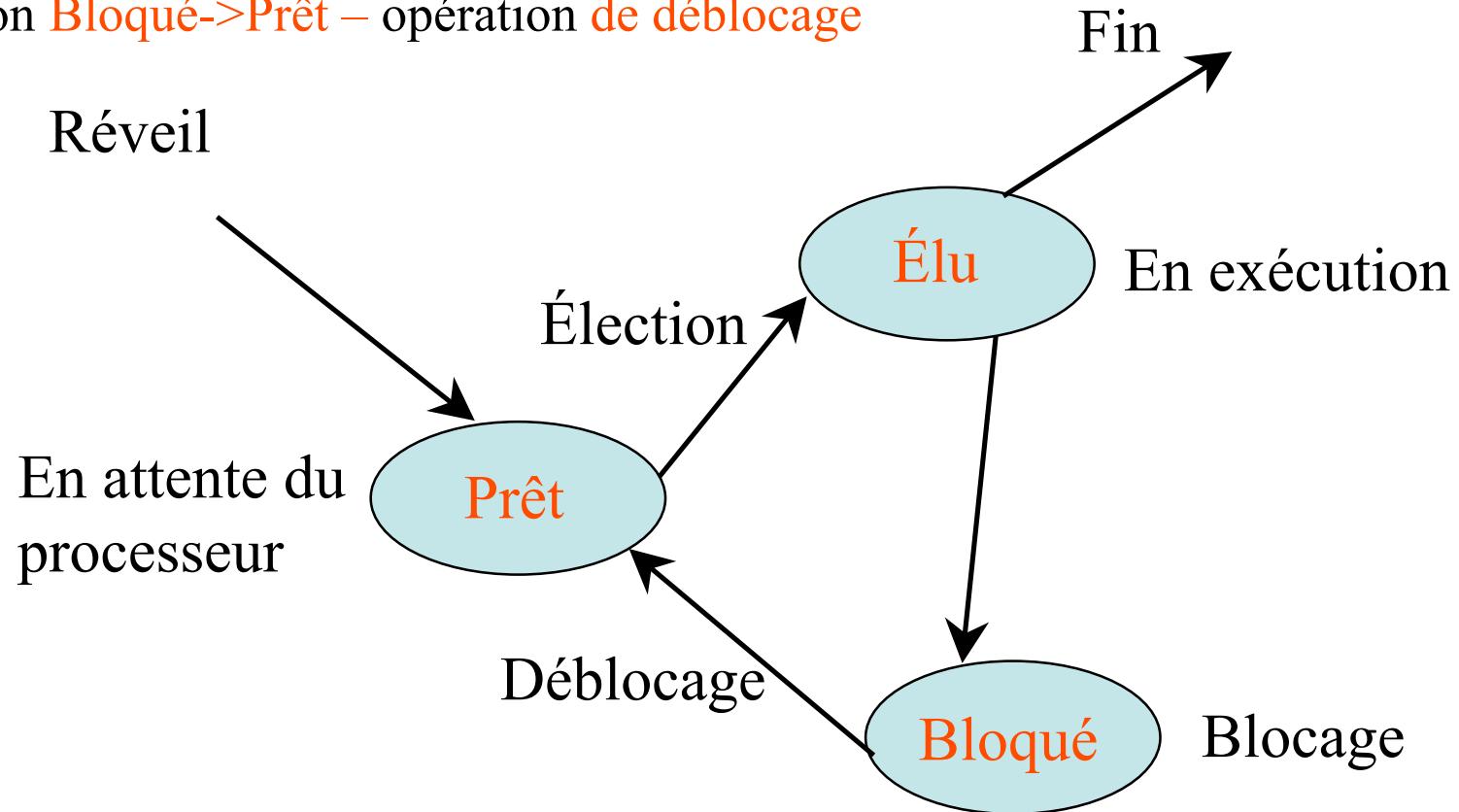


Diagramme d'états d'un processus

Transition Prêt → Élu – opération d'élection

Transition Élu → Bloqué – opération de blocage

Transition Bloqué → Prêt – opération de déblocage



Création d'un processus = état Prêt

Bloc de contrôle du processus

- PCB (Process Control Block) – une structure de description du processus associé au programme exécutable
- PCB permet la sauvegarde et la restauration du contexte mémoire et du contexte processeur lors des opérations de commutations de contexte

Identificateur processus
État du processus
Compteur ordinal
Contexte pour reprise (registres et pointeurs, piles ...)
Chaînage selon les files de l'Ordonnanceur
Priorité (ordonnancement)
Informations mémoire (limites et tables pages/segments)
Informations sur les ressources utilisées fichiers ouverts, outils de synchronisation, entrées-sorties
Informations de comptabilisation

Opérations sur les processus

- SE offre les opérations de gestion des processus
- Opérations pour la gestion des processus
 - Création de processus
 - Destruction de processus
 - Suspension de l'exécution
 - Reprise de l'exécution

Opérations sur les processus

- Création de processus
 - Un processus peut créer un ou plusieurs autres processus en invoquant un appel système de création de processus
 - Le processus créateur – processus père
 - Les processus créés – processus fils
 - Au fil des opérations de création initiées par les processus, se développe un arbre de filiation entre processus

Opérations sur les processus

- Création de processus
 - Les opérations de création de processus admettent les variantes suivantes selon les SE
 - Le processus créé hérite ou non de données et du contexte de son processus créateur
 - Le processus créé peut s'exécuter parallèlement à son père. Dans certains systèmes, le processus père doit attendre la terminaison de ses fils pour pouvoir reprendre sa propre exécution

Opérations sur les processus

- Destruction de processus
 - Processus a terminé son exécution
 - Le processus s'autodétruit en appelant une routine système de fin d'exécution
 - Processus commet une erreur irrécouvrable
 - Le processus est terminé par le système
 - Autre processus demande la destruction du processus
 - Appel à une routine système
- Le contexte du processus est démantelé
 - Ressources allouées au processus sont libérées
 - Bloc de contrôle est détruit

Opérations sur les processus

- Suspension d'exécution – momentanément arrêter l'exécution d'un processus pour la reprendre ultérieurement
 - Le contexte du processus est sauvegardé dans son PCB
 - Le processus passe dans l'état bloqué
 - Reprise d'exécution
 - Transition de déblocage, le processus entre dans l'état prêt

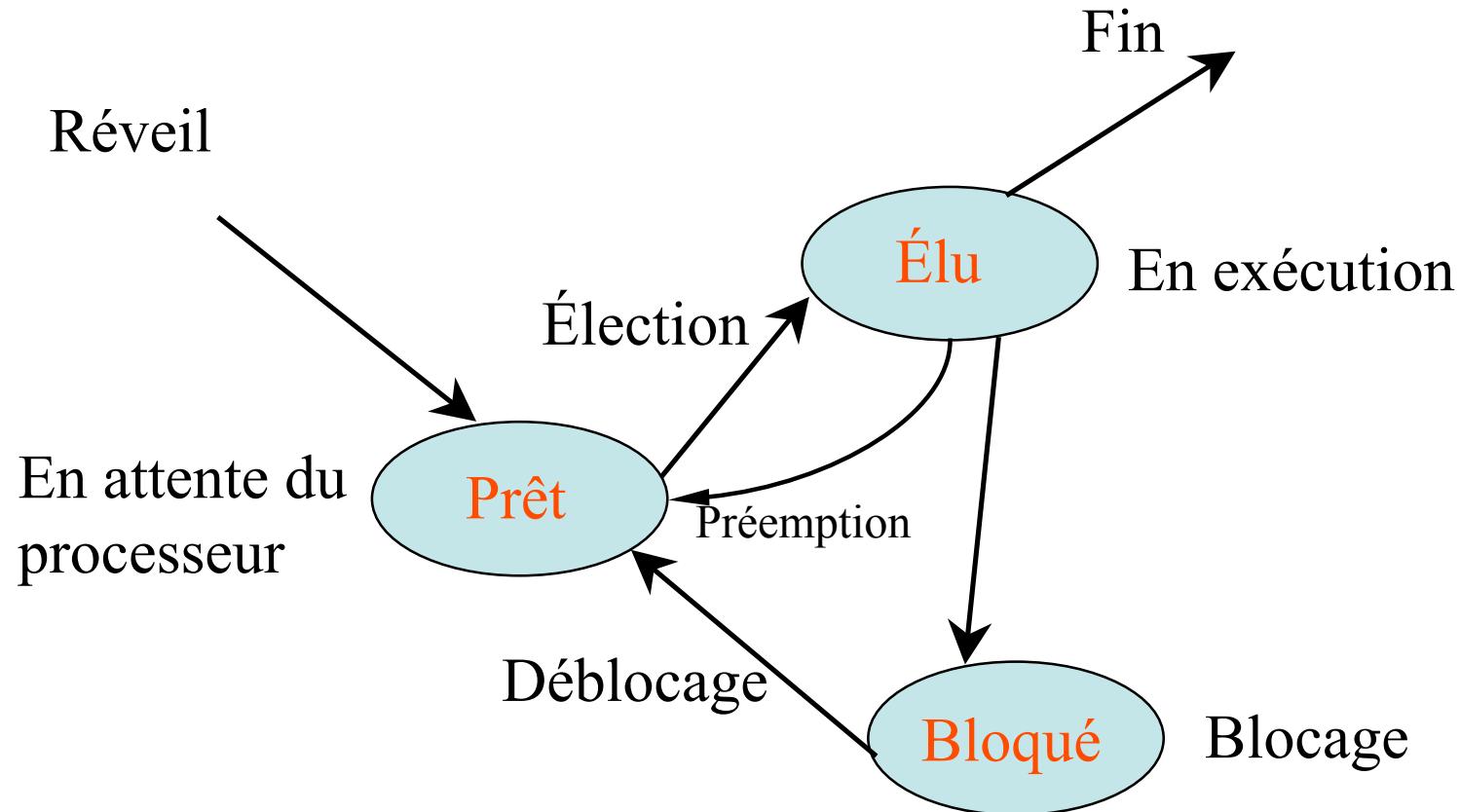
Ordonnancement sur l'unité centrale

- Ordonnancement préemptif et non préemptif
- Entités systèmes responsables de l'ordonnancement
- Politiques d'ordonnancement
- Exemples
 - Ordonnancement sous Linux

Ordonnancement sur l'unité centrale

- La fonction d'ordonnancement gère le partage du processeur entre les différents processus en attente pour s'exécuter, c'est-à-dire entre les différents processus qui sont dans l'état prêt.

Ordonnancement préemptif et non préemptif

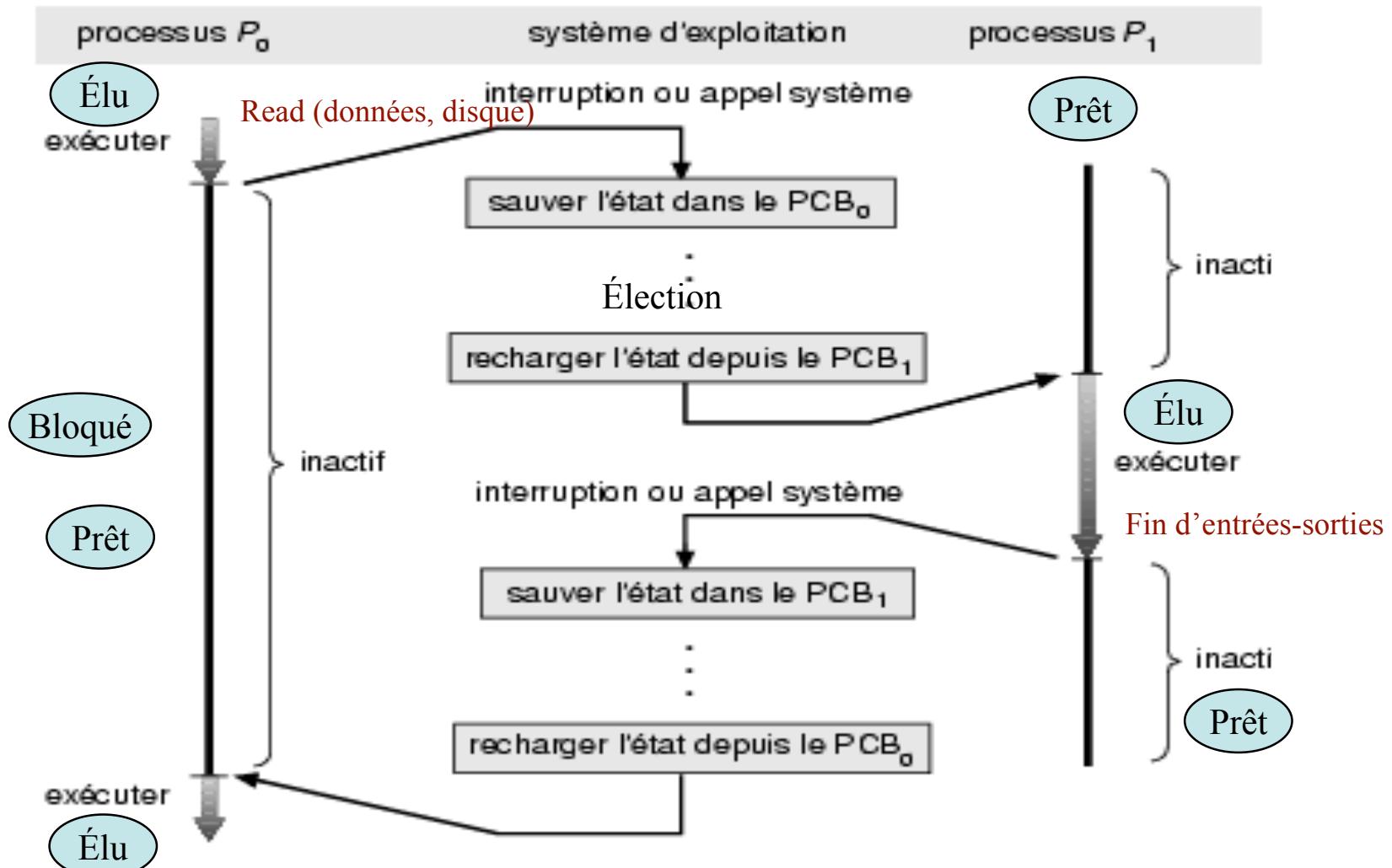


Transition élu->prêt – réquisition du processeur, préemption

Ordonnancement préemptif et non préemptif

- Selon si l'opération de réquisition est autorisée ou non, l'ordonnancement est qualifié d'ordonnancement préemptif ou non préemptif
 - Ordonnancement non préemptif
 - Transition de l'état élu vers l'état prêt est interdite: un processus quitte le processeur s'il a terminé son exécution ou s'il se bloque
 - Ordonnancement préemptif
 - La transition de l'état élu vers l'état prêt est autorisée: un processus quitte le processeur s'il a terminé son exécution, s'il se bloque ou si le processeur est réquisitionné

Déroulement des opérations d'ordonnancement

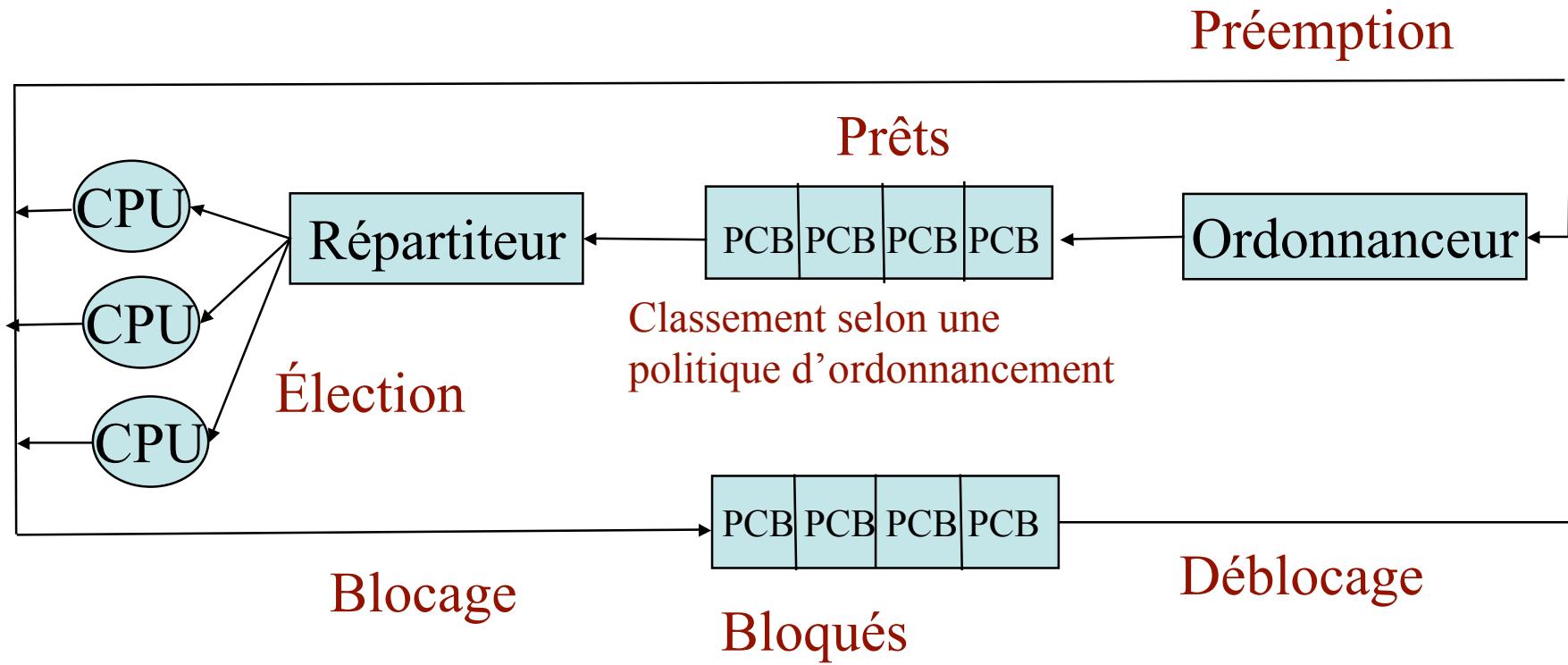


Les opérations d'ordonnancement prennent place lors de tout changement d'états des processus

Entités systèmes responsable de l'ordonnancement

- Les processus prêts et bloqués sont gérés dans deux files d'attentes distinctes qui chaînent leur PCB
 - Le module Ordonnanceur (scheduler) trie la file des processus prêts. Le tri s'appuie sur un critère donné spécifié par la politique d'ordonnancement
 - Le répartiteur (dispatcher) alloue un processeur parmi les processeurs libres (système multiprocesseur) à la tête de file de la liste des processus prêts

Entités système responsable de l'ordonnancement



Ordonnanceur et répartiteur

Politiques d'ordonnancement

- Politique d'ordonnancement détermine quel sera le prochain processus élu
- Selon si la préemption est autorisée ou non, la politique d'ordonnancement sera de type préemptive ou non
- Objectifs d'ordonnancement diffèrent selon les types de systèmes

Politiques d'ordonnancement

- Premier Arrivé, Premier Servi
 - Les processus sont élus selon l'ordre dans lequel ils arrivent dans la file d'attente des processus prêts
 - Il n'y a pas de réquisition
 - Avantages – Simplicité
 - Inconvénient
 - Processus de petit temps d'exécution sont pénalisés en terme de temps de réponse par les processus de grand temps d'exécution qui se trouvent avant eux dans la file d'attente

Politiques d'ordonnancement

- Plus Court d'Abord
 - L'ordre d'exécution des processus est fonction de leur temps d'exécution
 - Sans réquisition
 - Remédie à l'inconvénient pour PAPS
 - Difficulté
 - Connaissance a priori des temps d'exécution des processus

Politiques d'ordonnancement

- Politique par priorité
 - Chaque processus possède une priorité
 - À un instant donné, le processus élu est le processus prêt de plus forte priorité
 - Deux version selon si la réquisition est autorisée ou non
 - Réquisition est admise
 - Le processus couramment élu est préempté dès qu'un processus plus prioritaire devient prêt
 - Risque de famine pour les processus de petite priorité

Politiques d'ordonnancement

- Politique du tourniquet (round robin)
 - Systèmes en temps partagé
 - Le temps est découpé en tranches, quantum de temps (10 – 100 ms)
 - Lorsqu'un processus est élu, il s'exécute au plus durant un quantum de temps. Si le processus n'a pas terminé son exécution à l'issue du quantum de temps, il est préempté et il réintègre la file des processus prêts mais en fin de file
 - La valeur du quantum – un facteur important de performance (commutations de contexte)

Exemple

- Systèmes actuels – combinaison de deux des politiques: celles des priorités fixes et celles du tourniquet
 - La file des processus prêts est divisée en autant de sous files F_i qu'il existe de niveaux de priorité
 - Chaque file est gérée en tourniquet avec un quantum Q_i
 - Pour remédier au problème de famine – mécanisme d'extinction de priorité
 - La priorité d'un processus baisse au cours de son exécution

Services et facilités

- Gestion de la concurrence
 - Communication entre plusieurs programmes, synchronisation de l'accès aux données partagées (outil de communication et de synchronisation entre programmes)

Gestion de la concurrence

- **Les processus peuvent avoir besoin de communiquer entre eux pour échanger des données par le biais**
 - D'une zone mémoire partagée
 - D'un fichier
 - En utilisant les outils de communication offerts par le système d'exploitation
- **Les processus ne sont plus indépendants**
 - Accès concurrents aux ressources logicielles

Gestion de la concurrence

- Une ressource désigne toute entité dont a besoin un processus pour s'exécuter
 - Matérielle (le processeur, un périphérique)
 - Logicielle (variable)
- Une ressource est caractérisée par
 - un état
 - libre
 - occupée
 - Nombre de points d'accès
 - Le nombre de processus pouvant l'utiliser en même temps

Gestion de la concurrence

- **Ressource critique** – ressource ne pouvant être utilisée que par un seul processus à la fois
 - Processeur, imprimante
- **Utilisation d'une ressource**
 - Allocation de la ressource
 - Utilisation
 - Restitution de la ressource

Gestion de la concurrence

- Les phases d'allocation et de restitution d'une ressource doivent assurer que la ressource est utilisée conformément à son nombre de points d'accès
- Étape d'allocation
 - Peut bloquer un processus, si tous les points d'accès sont occupés

Gestion de la concurrence

- Étape de restitution
 - Peut entraîner le déblocage d'un autre processus en attente d'accès
- Synchronisation entre processus doit garantir une bonne utilisation des ressources
 - Une communication cohérente
 - Sans perte de données

Services et facilités

- Gestion de la mémoire
 - Allocation de la mémoire centrale: principe de la mémoire virtuelle, à un instant donné, seules les parties de code et données utiles à l'exécution sont chargées en mémoire centrale

Gestion de la mémoire

- La tâche principale de la gestion de la mémoire est de charger des programmes en mémoire pour qu'ils soient exécuté par le CPU
 - Mémoire virtuelle
 - La taille du programme, des données et de la pile peut dépasser la mémoire disponible. Le SE garde en mémoire les parties du programme qui sont utilisées et stocke le reste dans le disque
 - Cette méthode est basée sur deux principes de gestions, la SEGMENTATION et la PAGINATION

Gestion de la mémoire

■ Pagination

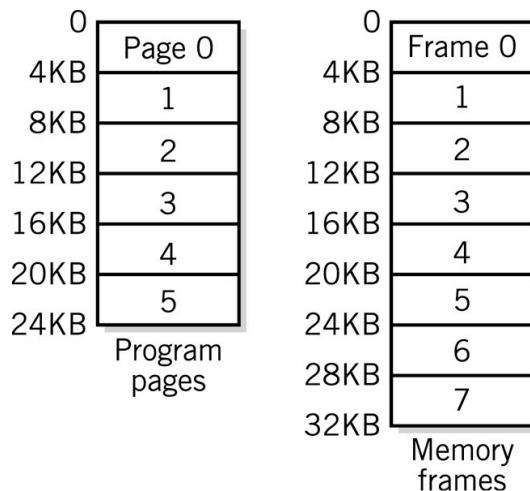
- L'espace d'adressage du programme est découpé en morceaux linéaires de même taille appelés pages
- L'espace de la mémoire physique est lui-même découpé en morceaux linéaires de même taille appelés case
 - Taille page = case – définie par le matériel (selon SE entre 512 octets et 8192 octets)
- Charger un programme en mémoire centrale - placer les pages dans les cases disponibles
 - Pour connaître à tout moment quelles sont les cases libres en mémoire centrale, le système maintient une table des cases
 - Pour chaque case de la mémoire physique, information
 - Libre ou occupée
 - Si occupée, quelle page et quel processus la possèdent

Pages et Cases (1)

	Programme	Mémoire
Unité	Page	Case
Adresse	Logique	Physique
Taille de l'unité	2 to 4KB	2 to 4KB
Quantité	# de bits dans l'instruction	Mémoire installée

Pages et Cases (2)

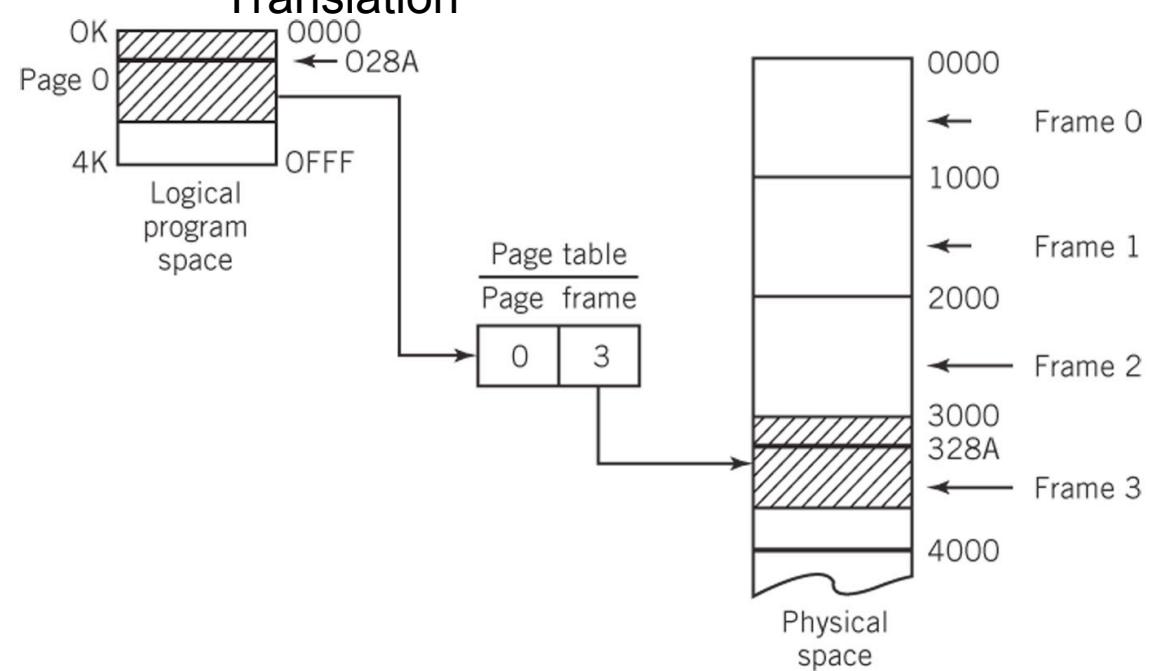
Pages and Frames



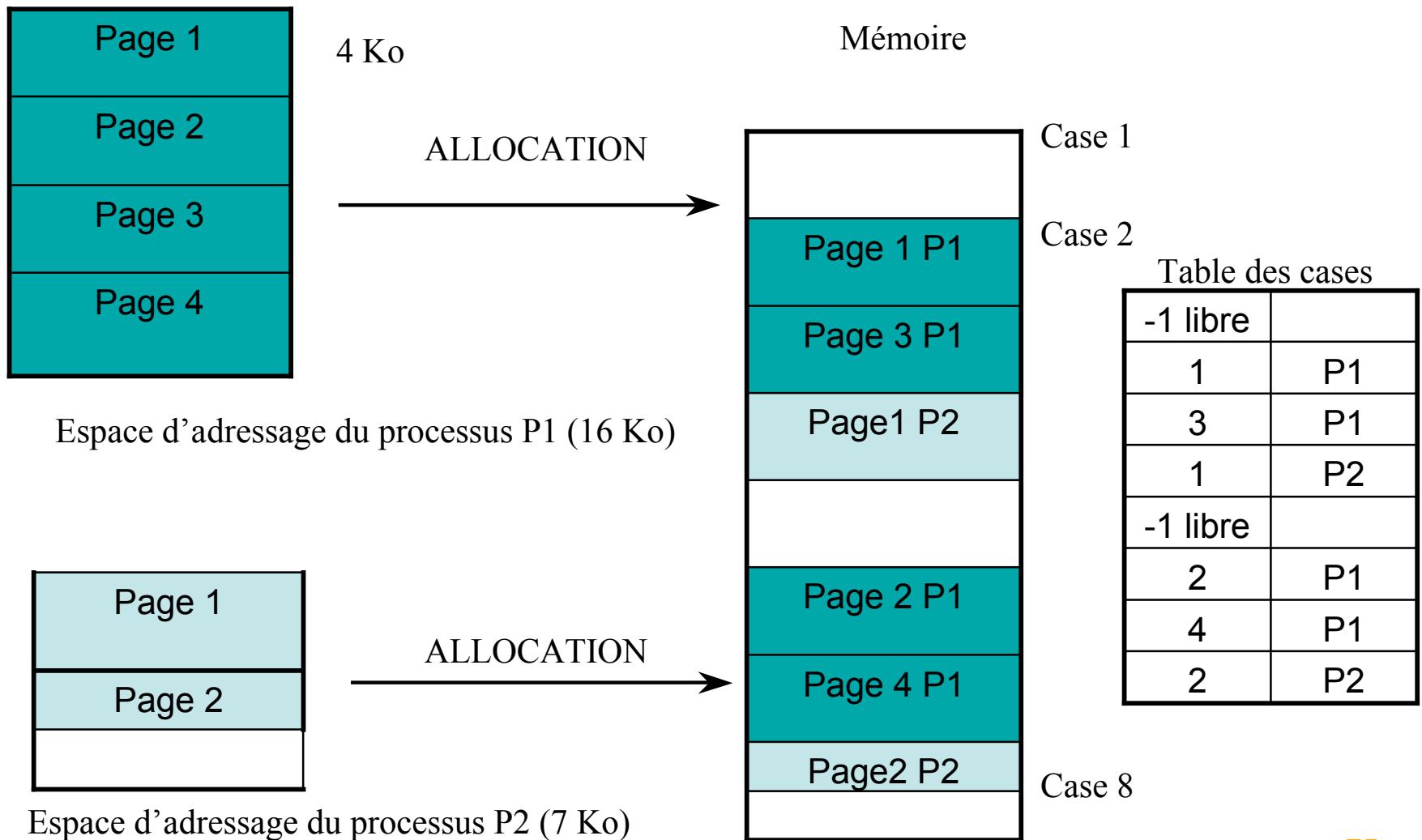
Each program has its collection of pages.

The total number of pages can exceed the number of frames (physical memory).

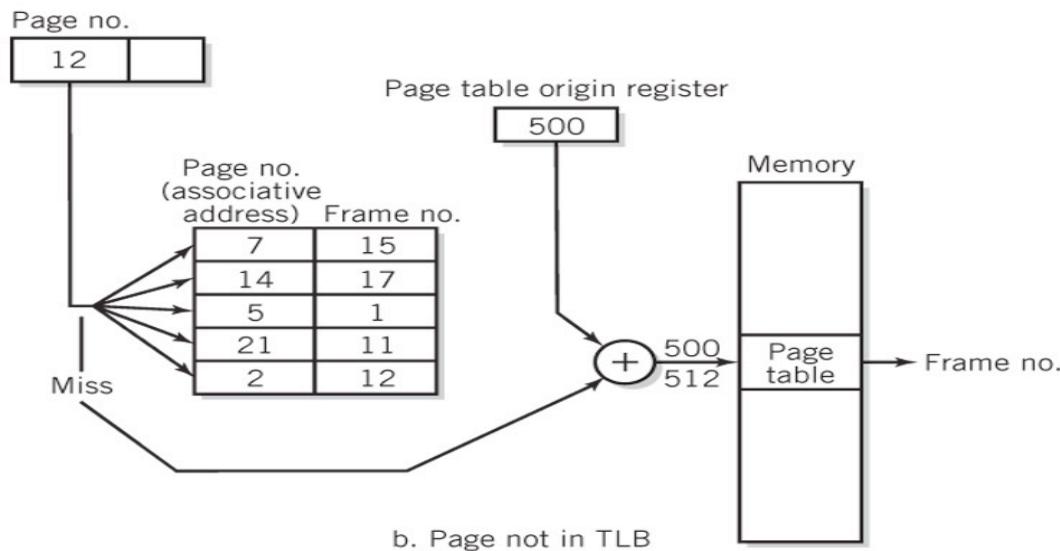
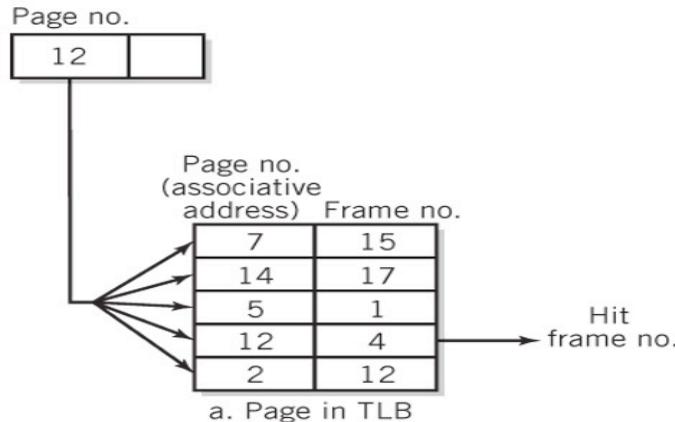
A Simple Page Table Translation



Gestion de la mémoire



Recherche d'une case



Gestion de la mémoire

- Conversion adresse logique – adresse physique
 - L'espace d'adressage du processus étant découpé en pages, les adresses générées dans cet espace d'adressage sont des adresses paginées (relative au début de la page)
 - L'adresse : <numéro de page p, déplacement d relativement au début de la page p>
 - Une adresse logique de m bits, en considérant des pages de 2^n octets, les m-n premiers bits correspondent au numéro de page p et les n bits restants au déplacement d dans la page

Gestion de la mémoire

- Conversion adresse logique – adresse physique
 - Il fait convertir l'adresse paginée générée au niveau du processeur en une adresse physique équivalente
 - L'adresse physique s'obtient à partir de son adresse logique en remplaçant le numéro de page p par l'adresse physique d'implantation de la case contenant la page p et en ajoutant à cette adresse le déplacement d du mot dans la page
 - MMU (memory management unit) – un dispositif matériel (partie de CPU), effectue cette conversion

Gestion de la mémoire

- Conversion adresse logique – adresse physique
 - Pour toute page il faut connaître dans quelle case de la mémoire centrale celle-ci a été placée
 - Correspondance page – case s'effectue grâce à une structure – table de pages
 - La table des pages – contient autant d'entrées que de pages dans l'espace d'adressage d'un processus
 - Chaque processus a sa propre table des pages
 - Chaque entrée – un couple <numéro de page, numéro de case physique dans laquelle la page est chargée>

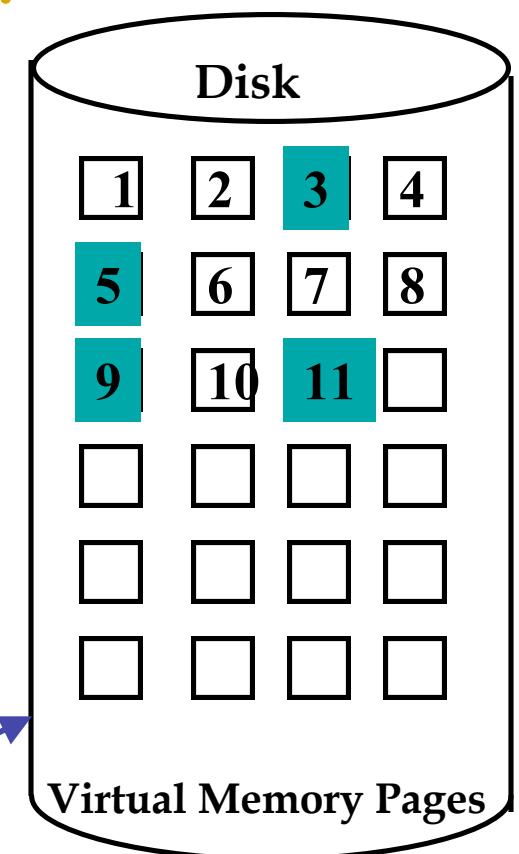
Table de pages

Page Frame

1	6
2	4
3	
4	8
5	
6	10
7	1
8	2
9	
10	7
11	

Pages not in main memory:
page fault when accessed

Swap space



Allocation de la mémoire pour 3 processus

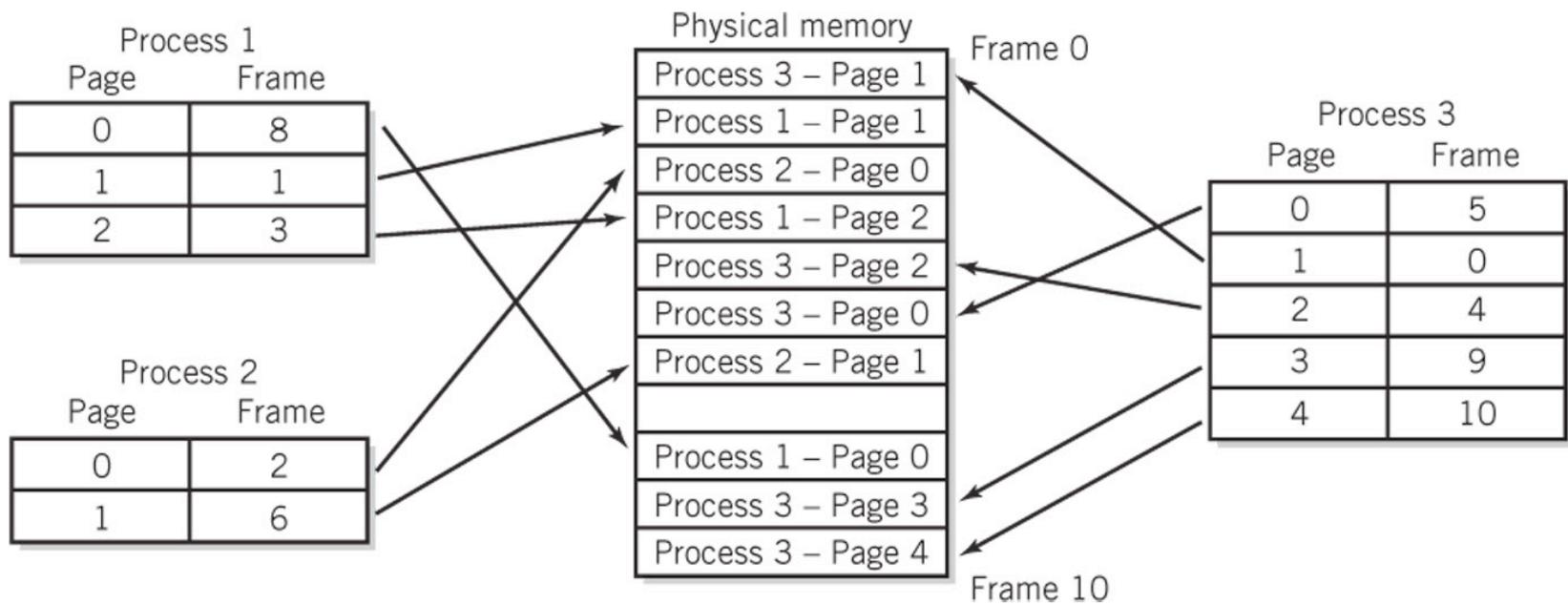


Table de cases

Inverted Page Table for the previous slide

The table represents what is in physical memory

Frame	Process #	Page
0	3	1
1	1	1
2	2	0
3	1	2
4	3	2
5	3	0
6	2	1
7		
8	1	0
9	3	3
10	3	4

Free page frame

Gestion de la mémoire

- Le défaut de page
 - Processus tente d'accéder à une page qui n'est pas en mémoire centrale
 - Accès à l'entrée p de la table des pages du processus actif et test de la valeur du bit de validation V ;
 - Si la valeur du bit V est à 0, alors il y a défaut de page. Une opération d'entrées-sorties est lancée pour charger la page dans la mémoire centrale (l'adresse de la page sur le disque est stockée dans la table des pages)
 - La page est placée dans une case libre, trouvée par l'intermédiaire de la table des cases
 - La table des pages du processus est mise à jour
 - La conversion de l'adresse logique vers l'adresse physique est reprise

Gestion de la mémoire

- Le remplacement de pages
 - Traitement d'un défaut de page pour un processus peut se faire
 - Localement à ce processus
 - Globalement sur l'ensemble des processus (plus souvent mis en œuvre)
 - Algorithmes de remplacement de pages
 - FIFO (First In, First Out)
 - LRU (Least Recently Used)
 - LFU (Least Frequently Used)
 - MFU (Most Frequently Used)

Gestion de la mémoire

Algorithmes de remplacement de pages

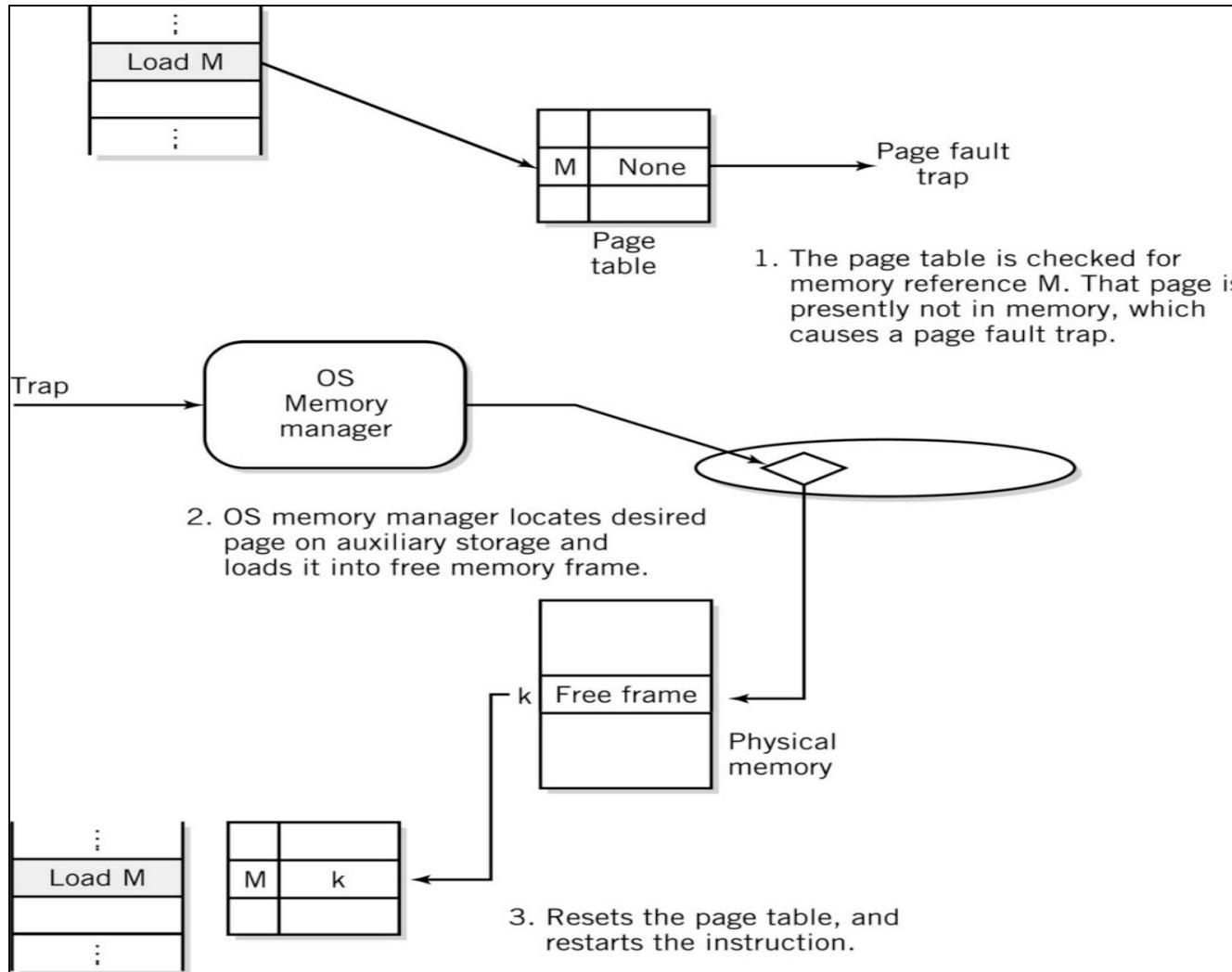
- FIFO (First In, First Out)
 - La page la plus anciennement chargée qui est remplacée
 - Performances ne sont pas toujours bonnes (l'âge ne reflète pas d'utilité)
- LRU (Least Recently Used)
 - La page la moins récemment utilisée qui est remplacée
 - Localité temporelle
 - Le plus utilisé, mais plus coûteuse

Gestion de la mémoire

Algorithmes de remplacement de pages

- LFU (Least Frequently Used)
 - La page la moins fréquemment utilisée est remplacée
 - Problème vis-à-vis des pages abondamment référencées sur un court laps de temps =>compteur pour ces pages – élevé => ne sont pas retirées de la mémoire même si elles ne sont plus jamais référencées
- MFU (Most Frequently Used)
 - La page la plus fréquemment utilisée qui est remplacée
 - L'argument à la base – une page ayant un petit nombre de références vient sans doute d'être chargée en mémoire centrale et doit donc y rester

Étapes lors de défaut de page



Gestion de la mémoire

- Conversion adresse logique – adresse physique
 - Chaque processus dispose de sa propre table de pages => chaque opération de commutation de contexte se traduit par un changement de table des pages de manière à ce que la table active corresponde à celle du processus élu
 - Deux approches de réalisation de la table des pages
 - Structure matérielle réalisée grâce à des registres de la MMU
 - Structure logicielle placée en mémoire centrale

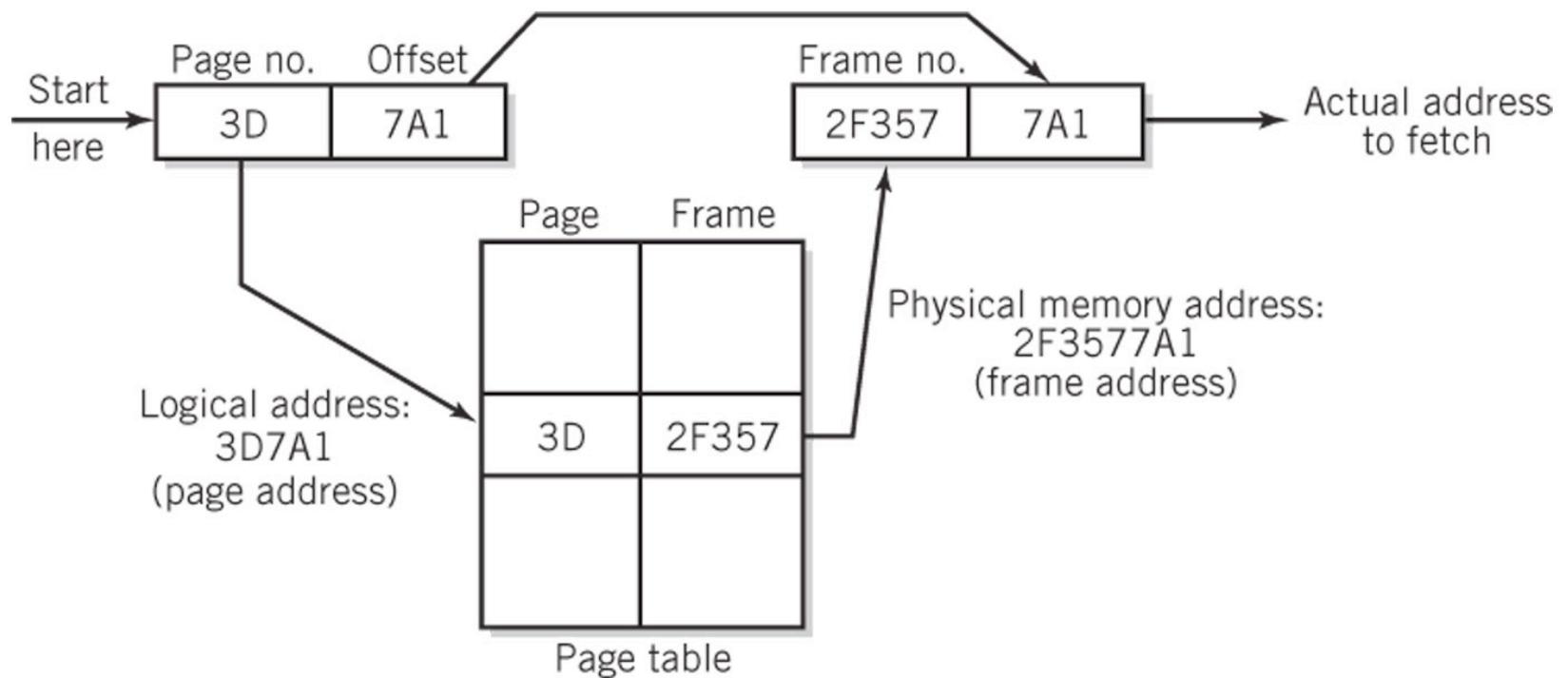
Gestion de la mémoire

- Réalisation de la table des pages
 - Structure matérielle réalisée grâce à des registres de la MMU
 - Nécessite un accès à la mémoire
 - Ne peut convenir que pour de petites tables des pages
 - Structure logicielle placée en mémoire centrale
 - Deux accès à la mémoire
 - Un premier accès permet de lire l'entrée de la table des pages correspondant à la page p cherchée et délivre une adresse physique c de case dans la mémoire centrale
 - Un second accès est nécessaire à la lecture ou l'écriture de l'octet recherché à l'adresse c+d
 - Réalisation des tables de pages de très grande taille

Gestion de la mémoire

- Protection de l'espace d'adressage des processus
 - Des bits de protection sont associés à chaque page de l'espace d'adressage du processus et permettent ainsi de définir le type d'accès autorisés à la page
 - Ces bits de protection sont mémorisés pour chaque page dans la table des pages du processus
 - 3 bits sont utilisés - l'autorisation d'accès en
 - Lecture – r
 - Écriture – w
 - Exécution - x

Translation d'adresse



Gestion de la mémoire

- Protection de l'espace d'adressage des processus
 - Lors d'un accès à une page, la cohérence du type d'accès avec les droits associés à la page est vérifiée et une trappe est levée par le SE si le type d'accès réalisé est interdit
 - Chaque processus ne peut avoir accès qu'à sa propre table des pages =>chaque espace d'adressage est protégé
 - Pour que deux processus puissent partager un ensemble de pages =>référencer cet ensemble dans sa table des pages respective

Gestion de la mémoire

- Pagination
 - Pagination multiniveaux
 - L'espace d'adressages logiques supporté par les SE actuels est très grand (de 2^{32} à 2^{64} octets)
 - La table des pages d'un processus peut devenir également de très grande taille et comporter jusqu'à un million d'entrées
 - Il n'est dès lors plus envisageable de charger de manière contiguë la table des pages d'un processus
 - Solution: paginer la table des pages elle-même

Gestion de la mémoire

- Pagination

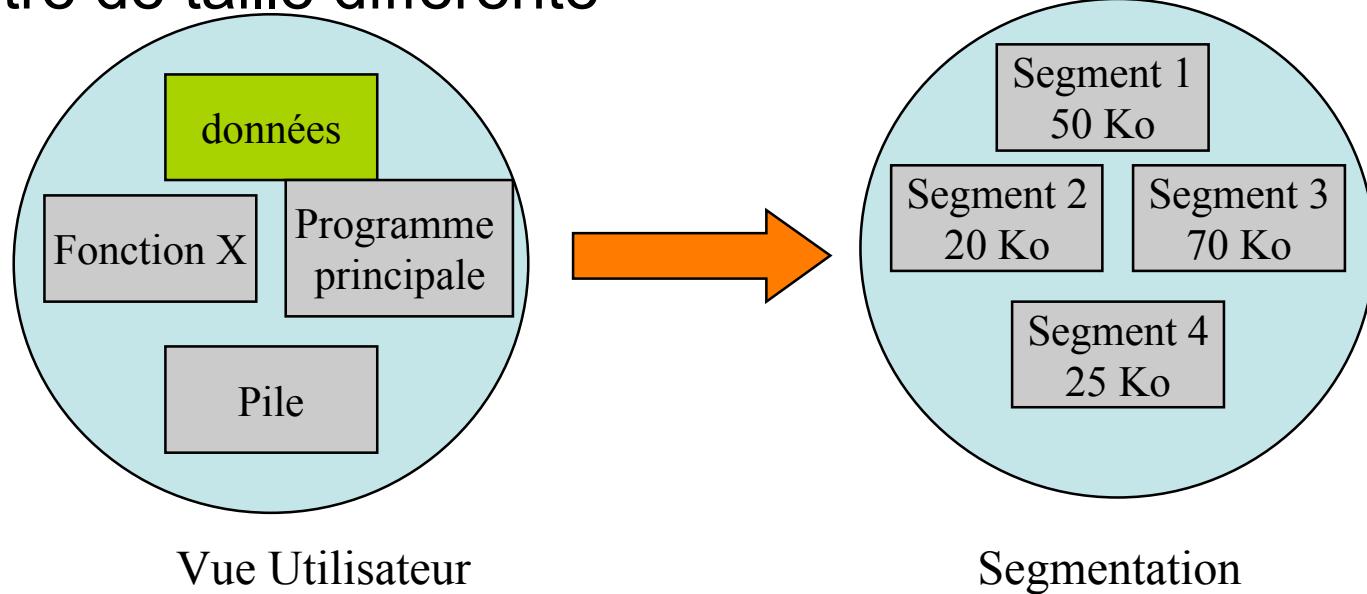
- Pagination mult;niveaux
- L'adresse paginée devient un triplet $\langle hp, p, d \rangle$
 - hp – l'entrée d'une hypertable des pages
 - Chaque entrée correspond à une page contenant une partie de la table des pages du processus
 - p – une entrée de cette partie de la table des pages
 - d – un déplacement dans la page p
- L'hypertable des pages est placée en mémoire centrale et son adresse d'implantation en mémoire centrale est repérée par un registre matériel de la MMU
 - 3 accès à la mémoire

Gestion de la mémoire

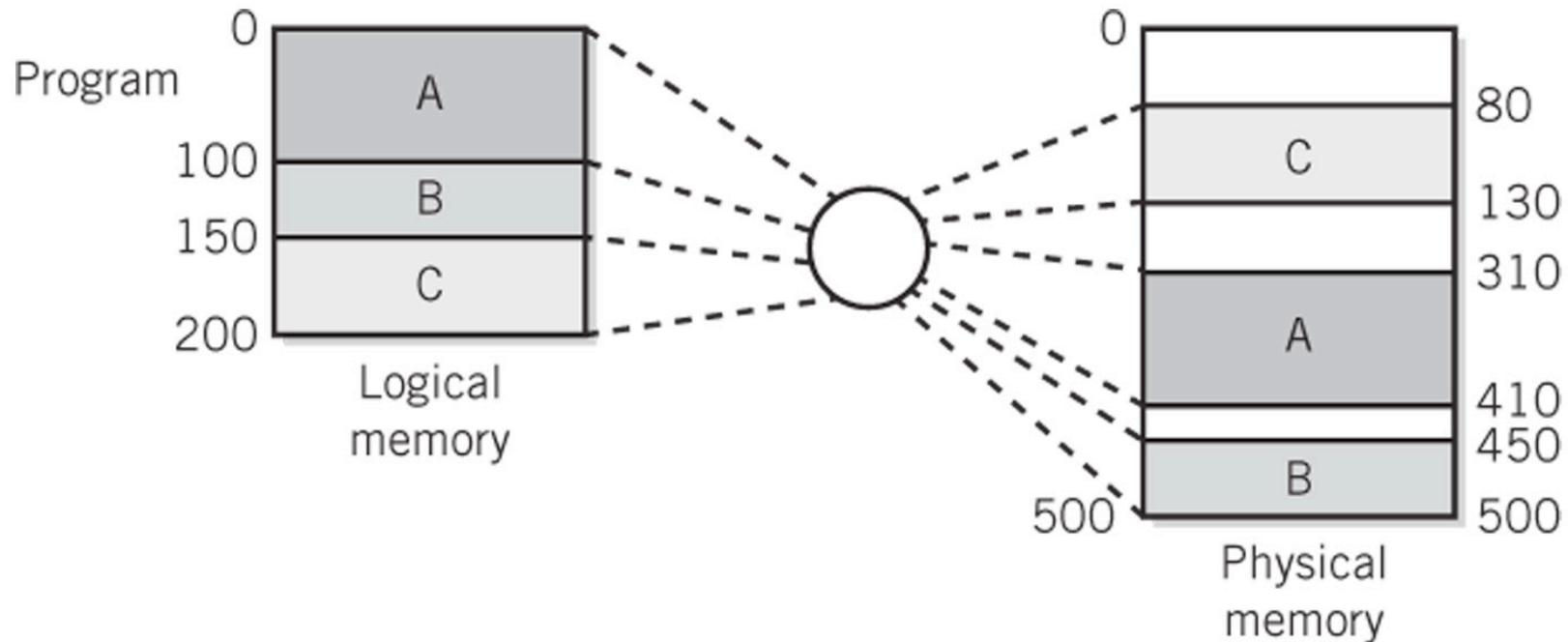
- Segmentation
 - La pagination constitue un découpage de l'espace d'adressage du processus qui ne correspond pas à l'image que le programmeur a de son programme
 - Données
 - Programme principal
 - Procédures séparées
 - Pile d'exécution
 - La segmentation est un découpage de l'espace d'adressage qui cherche à conserver la vue du programmeur

Gestion de la mémoire

- Segmentation
 - Lors de la compilation, le compilateur associe un segment à chaque morceau du programme compilé
- Segment est un ensemble d'emplacement mémoire consécutifs non sécable
 - Les segments d'un même espace d'adressage peuvent être de taille différente



Segmentation



Gestion de la mémoire

- Principe de la mémoire virtuelle
 - à un instant donné, seules les parties de code et données utiles à l'exécution sont chargées en mémoire centrale
 - Répartition de la mémoire centrale entre les processus
 - Allocation équitable ou fixe
 - Même nombre de cases quelle que soit la taille de l'espace d'adressage
 - Allocation proportionnelle
 - Proportionnellement à la taille des processus

Gestion de la mémoire

- Le principe de la mémoire virtuelle est couramment implémenté avec la pagination à la demande
 - Les pages des processus ne sont chargées en mémoire centrale que lorsque le processeur demande à y accéder

Mémoire Virtuelle vs. Mémoire Cache

- Cache améliore le temps d'accès à la mémoire
- Mémoire Virtuelle augmente la quantité de la mémoire considérée
 - Indépendante de la configuration et de la capacité de la mémoire physique

Machines virtuelles: le problème et la solution

- Comment permettre de rouler différents SE sur une seule machine physique?
- Pas évident, car chaque SE demande accès direct au matériel
- SOLUTION: Un programme qui crée une couche qui met à disposition plusieurs machines physiques *virtuelles*
- Sur chacune, nous pouvons rouler un SE différent

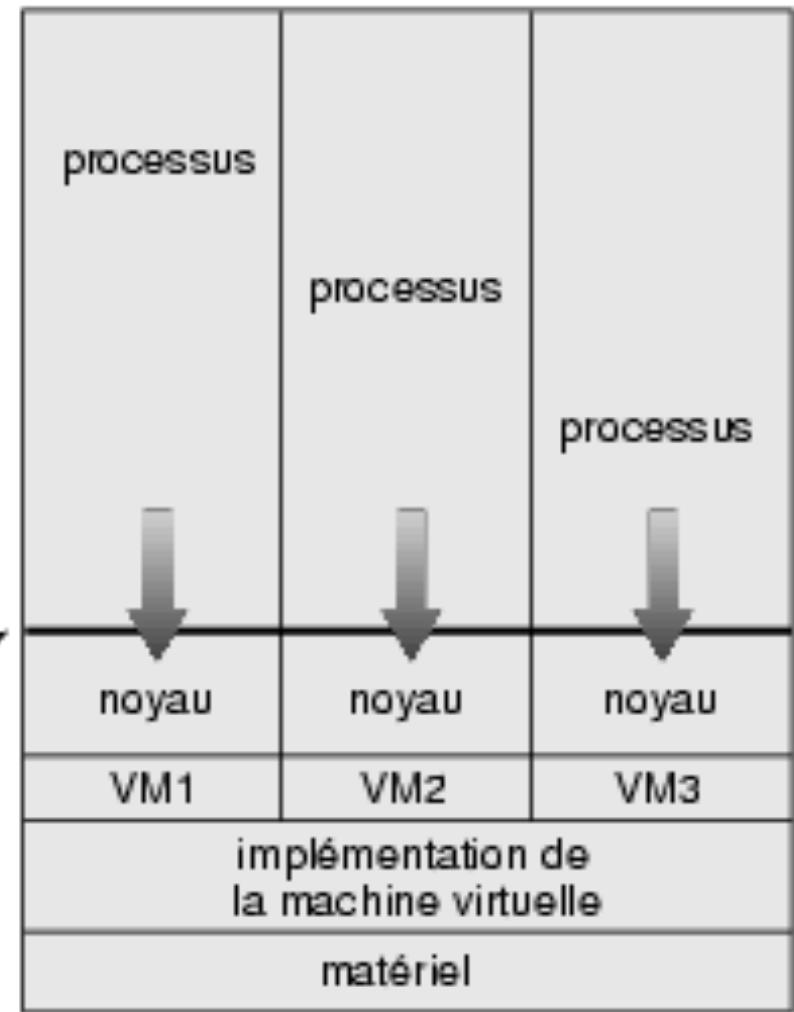
Machines Virtuelles

- *Virtuel* en informatique dénote quelque chose qui n'est pas *réel*, n'est pas du matériel: il est construit par le logiciel sur la base des ressources fournies par le matériel
- Une machine virtuelle est une machine créée par des couches de logiciel
- Elle peut avoir des caractéristiques identiques à la machine physique du système:
 - mêmes instructions, etc.
- Ou elle peut 'simuler' une autre machine physique
 - p.ex. pour exécuter Microsoft sur Apple
- Plusieurs machines virtuelles peuvent être créées sur une machine physique donnée!

- (a) Une seule mach. réelle et un seul noyau
(b) Plus. mach. virtuelles et plus. noyaux
-



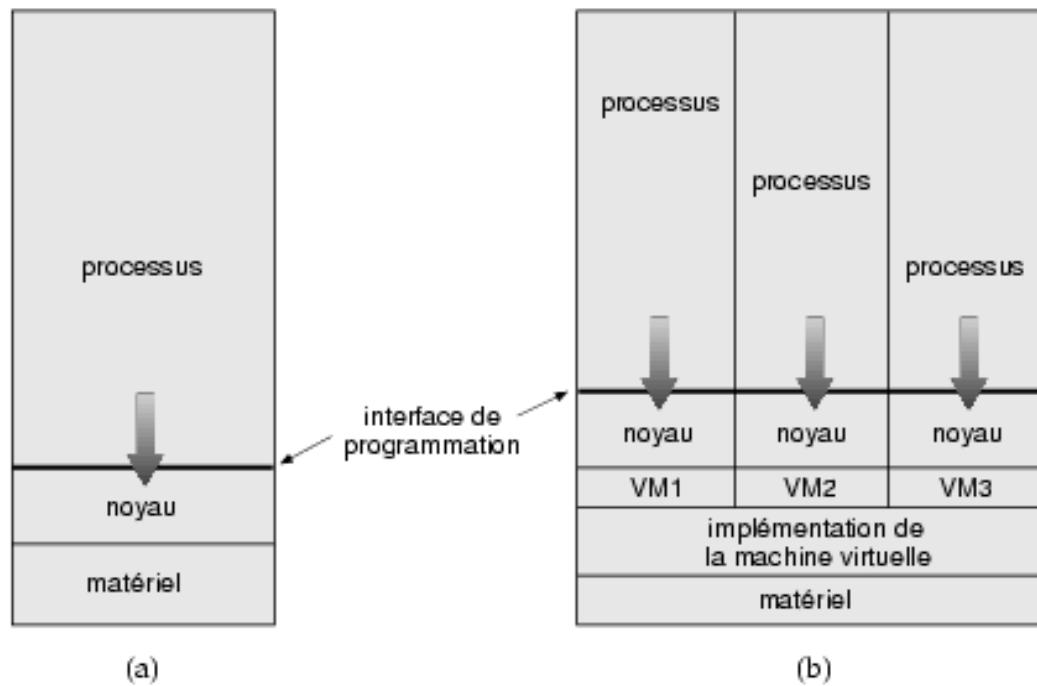
(a)



(b)

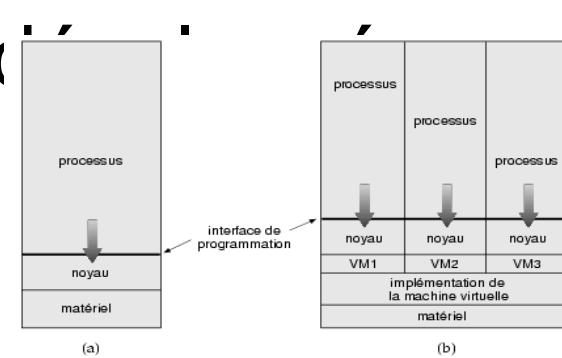
Fonctionnement typique

- Le système VM laisse exécuter normalement les instructions non privilégiées
- Les appels au système sont exécutés par le système VM et les résultats sont passés à la machine virtuelle sur laquelle le processus exécute



Avantages

- Chaque machine virtuelle peut utiliser un SE différent!
- En théorie, on peut bâtir des machines virtuelles sur des machines virtuelles!
- Protection complète, car les machines virtuelles sont complètement isolées les unes des autres
- Un nouveau SE peut être installé sur une machine virtuelle
- sans déranger les autres



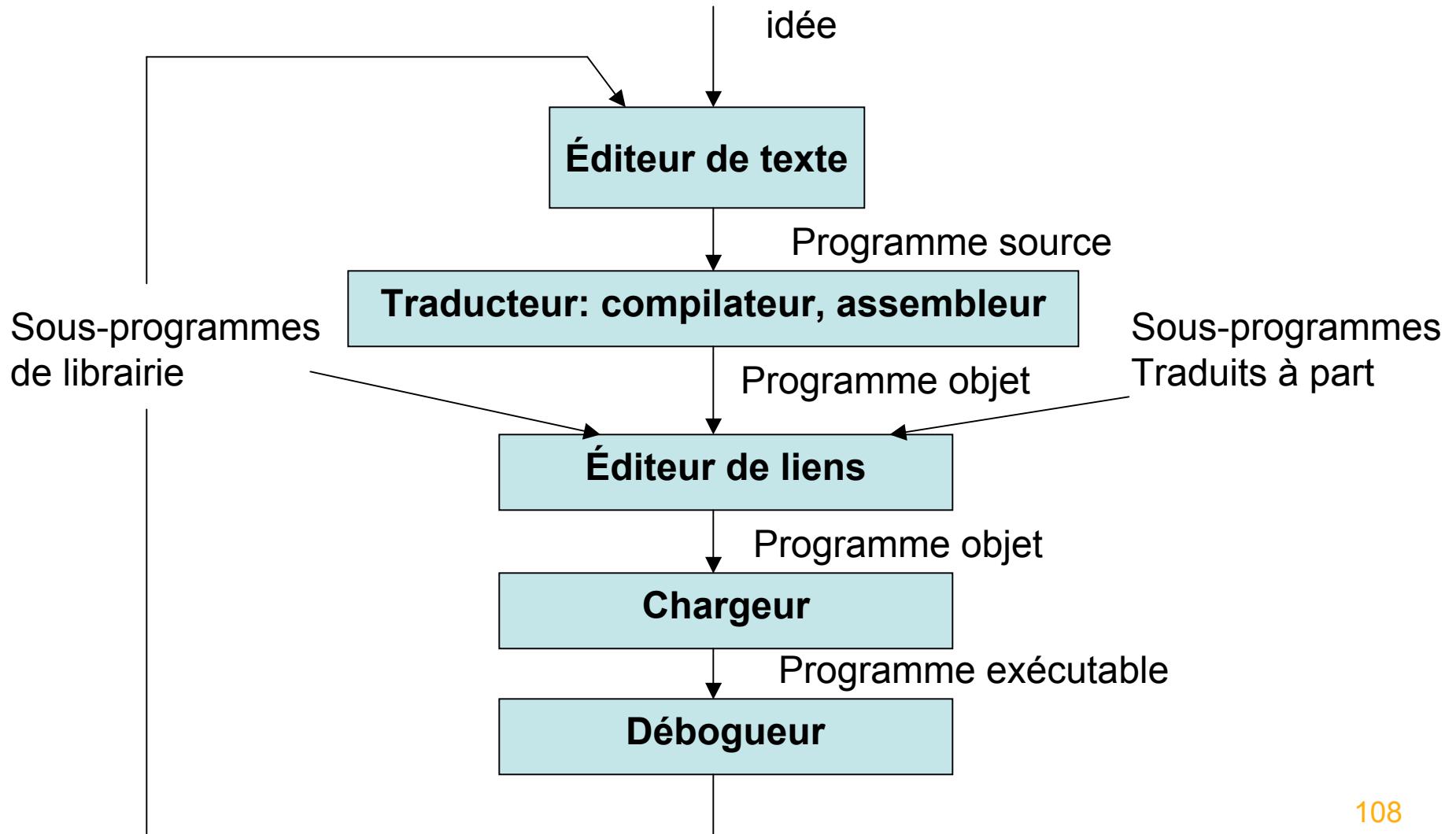
Implémentations

- Le concept de VM est très utilisé pour permettre de rouler un SE sur un autre
- P.ex. SUN, Apple, Linux permettent de rouler Windows sur leur plateforme,
- Ils doivent fournir à Windows un environnement que Windows reconnaît comme son environnement Intel usuel

Outils de programmation

- Les outils classiques utilisés dans le développement d'un programme sont
 - Éditeur de texte
 - Traducteur
 - Compilateur
 - Assembleur
 - Éditeur de liens
 - Débogueur

Outils de programmation



Compilateur

- Le rôle du compilateur est de traduire un fichier programme source en langage de haut niveau (avec lequel le programmeur s'est libéré des détails des instructions machines) en instructions/langage machine

Compilateur

- Le travail du compilateur se divise en plusieurs phases:
 - L'analyse lexicale
 - Reconnaissance des mots du langage
 - L'analyse syntaxique
 - Vérification de la syntaxe
 - L'analyse sémantique
 - L'optimisation et la génération du code objet

Compilateur

- Analyse lexicale
 - Consiste à lire le programme source et à produire une séquence d'éléments syntaxiques (nombres, variables, identificateurs, opérateurs, etc.)
 - Permet d'identifier quelques erreurs (éléments \neq langage, nombres illégaux, etc.)

Compilateur

- Analyse syntaxique
 - Analyse la suite de symboles issus de l'analyseur lexical et vérifie si cette suite de symboles est conforme à la syntaxe du langage (règles de la grammaire)
 - Analyseur syntaxique essaye de construire l'arbre syntaxique correspondant au programme. Dans cet arbre les feuilles correspondent aux symboles issus de l'analyse lexicale et les nœuds intermédiaires correspondent aux objets grammaticaux

Compilateur

- Analyse sémantique
 - Associer un sens aux différentes phrases du programme source
 - Reconnaître les objets manipulés et analyser leurs propriétés
 - Type de l'objet, sa durée de vie, sa taille et son adresse
 - Contrôler que l'utilisation de ces objets se fait de manière cohérente
 - Recherche les erreurs de typage, les déclarations multiples, absentes ou inutiles, les expressions incohérentes

Compilateur

- L'optimisation et la génération du code objet
 - Étape ultime de la compilation
 - Consiste à produire dans un fichier objet le code machine équivalent au code du langage de haut niveau, 3 étapes
 - Génération d'un code intermédiaire
 - Optimisation de ce code intermédiaire
 - Génération du code final

Compilateur

- Génération d'un code intermédiaire
 - Consiste à remplacer les phrases reconnues par des macros plus aisément manipulables qui ne font pas de référence aux registres de la machine cible
- Optimisation du code intermédiaire
 - Vise à produire un code machine plus performant
 - L'exécution plus rapide
 - Un code plus compact dont l'encombrement mémoire est moindre

Compilateur

- Optimisation du code intermédiaire
 - Diverses améliorations, exemples:
 - La réduction des expressions constantes
 - Simplification des boucles et prés-évaluation des expressions constantes
- Génération du code final
 - Les macros sont remplacées par les instructions machine correspondantes avec utilisations des registres et les objets sont remplacés par leur adresse
 - Le code obtenu à ce niveau est appelé code relogable
 - Toutes les adresses des objets sont calculées en considérant que l'adresse du premier octet du code est égale à 0

Éditeur de lien

- Un éditeur de lien est un logiciel qui permet de combiner plusieurs programmes objet en un seul
- Pour pouvoir développer de gros programmes on structure ceux-ci en modules que l'on traduit indépendamment

Chargeur

- Le programme objet après édition de lien doit être chargé en mémoire centrale pour être exécuter
- Fichier exécutable – fichier relogable
 - Lorsque le chargeur copie le code exécutable depuis le disque vers la mémoire centrale, il implante le code dans un espace libre de la mémoire centrale avec une adresse quelconque appelée adresse d'implantation mémoire

Chargeur

- Fichier exécutable – fichier relogable
 - Toutes les adresses calculées dans le programme exécutable doivent être modifiées
 - Opération de translation des adresses
 - Ajouter à chaque adresse la valeur de l'adresse d'implantation mémoire
 - Deux types de chargement
 - Statique
 - L'opération de translation est effectuée au moment du chargement pour toutes adresses
 - Dynamique
 - L'opération de translation est effectuée au moment d'utilisation d'une adresse relogable au cours de l'exécution par le processeur

Débogueur

- Le débogueur est un logiciel qui facilite la mise au point détection des erreurs dans un programme. Il permet de suivre pas à pas l'exécution d'un programme en examinant le contenu de la mémoire et des registres