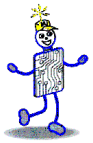


Système d'Exploitation

et

Outils de programmation
Partie 2

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

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

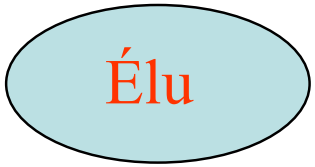
- Définitions
- États d'un processus
- Bloc de contrôle du processus
- Opérations sur les processus

Notion de Processus: Définitions

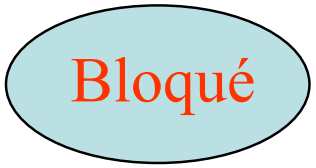
- Un programme en cours d'exécution auquel est associé un environnement processeur (PC, registres généraux, …) et un environnement mémoire (zone de code, de données et de pile) appelés contexte du processus
- Instance dynamique d'un programme et incarne le fil d'exécution de celui-ci dans un espace d'adressage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

Notion de Processus: États d'un processus

- Au fur et a 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



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.

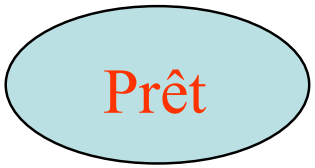
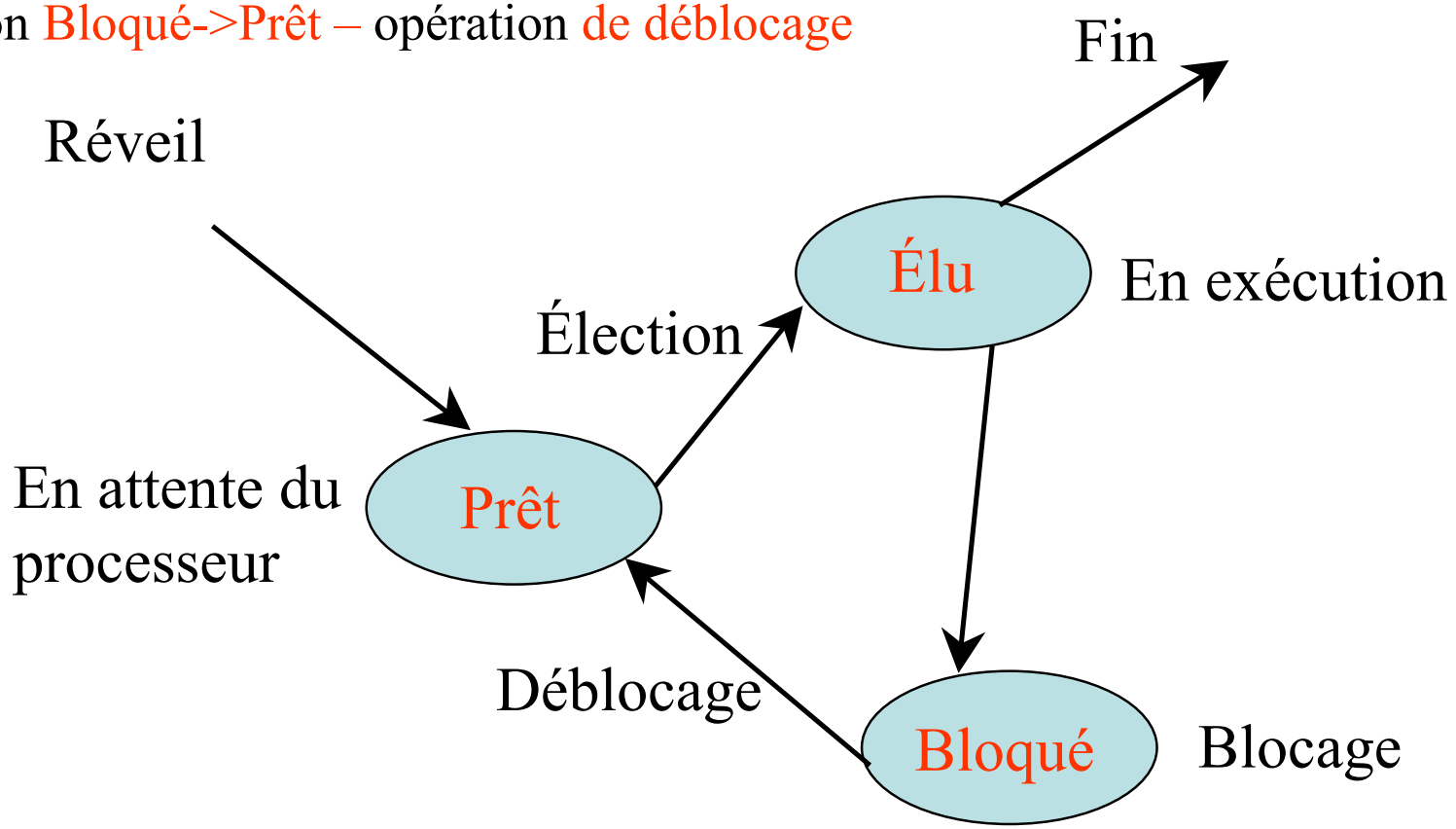


Diagramme d'états d'un processus

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

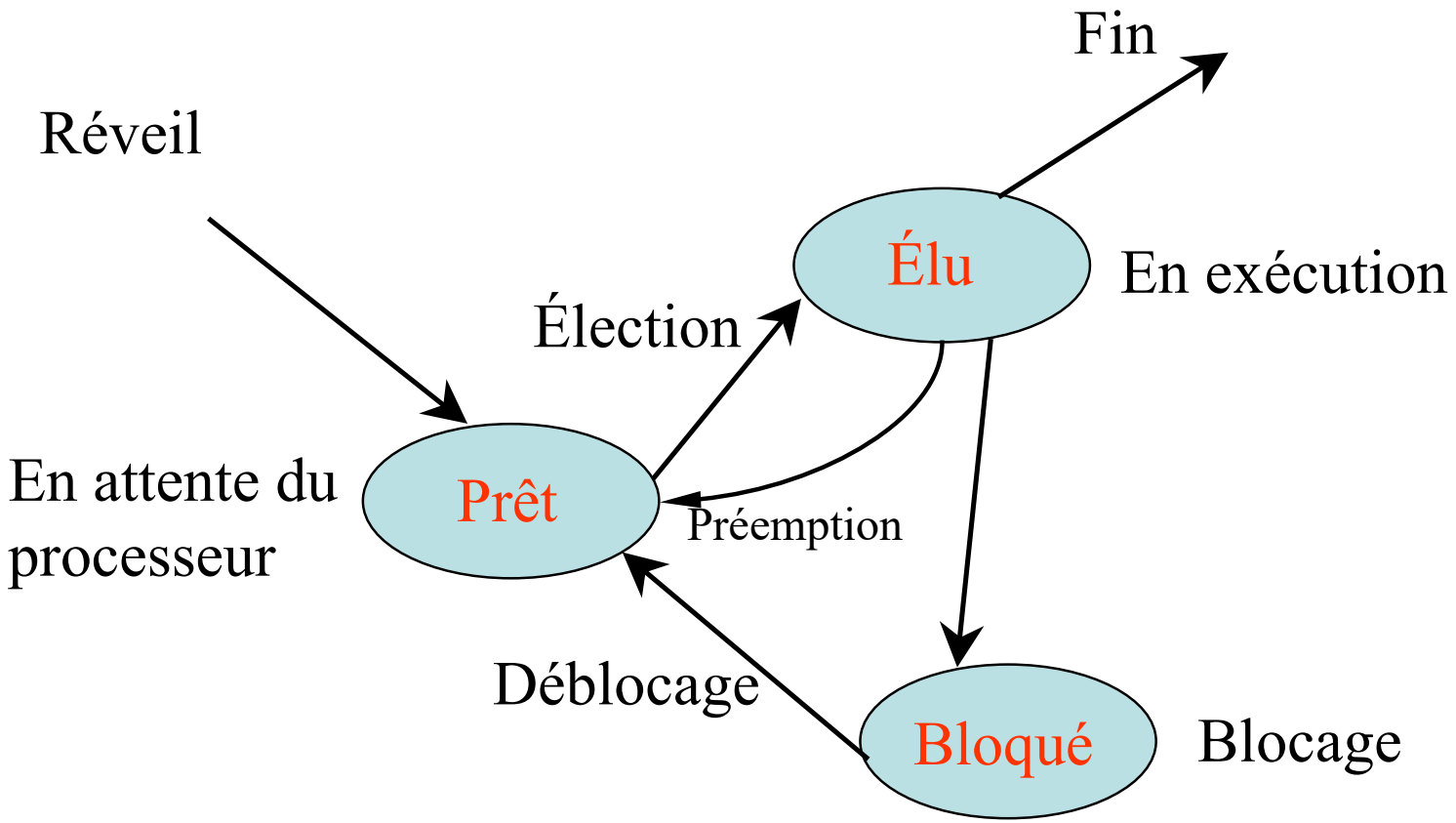
Ordonnancement sur l'unité centrale

- Ordonnancement préemptif et non préemptif
- Entités systèmes responsable 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

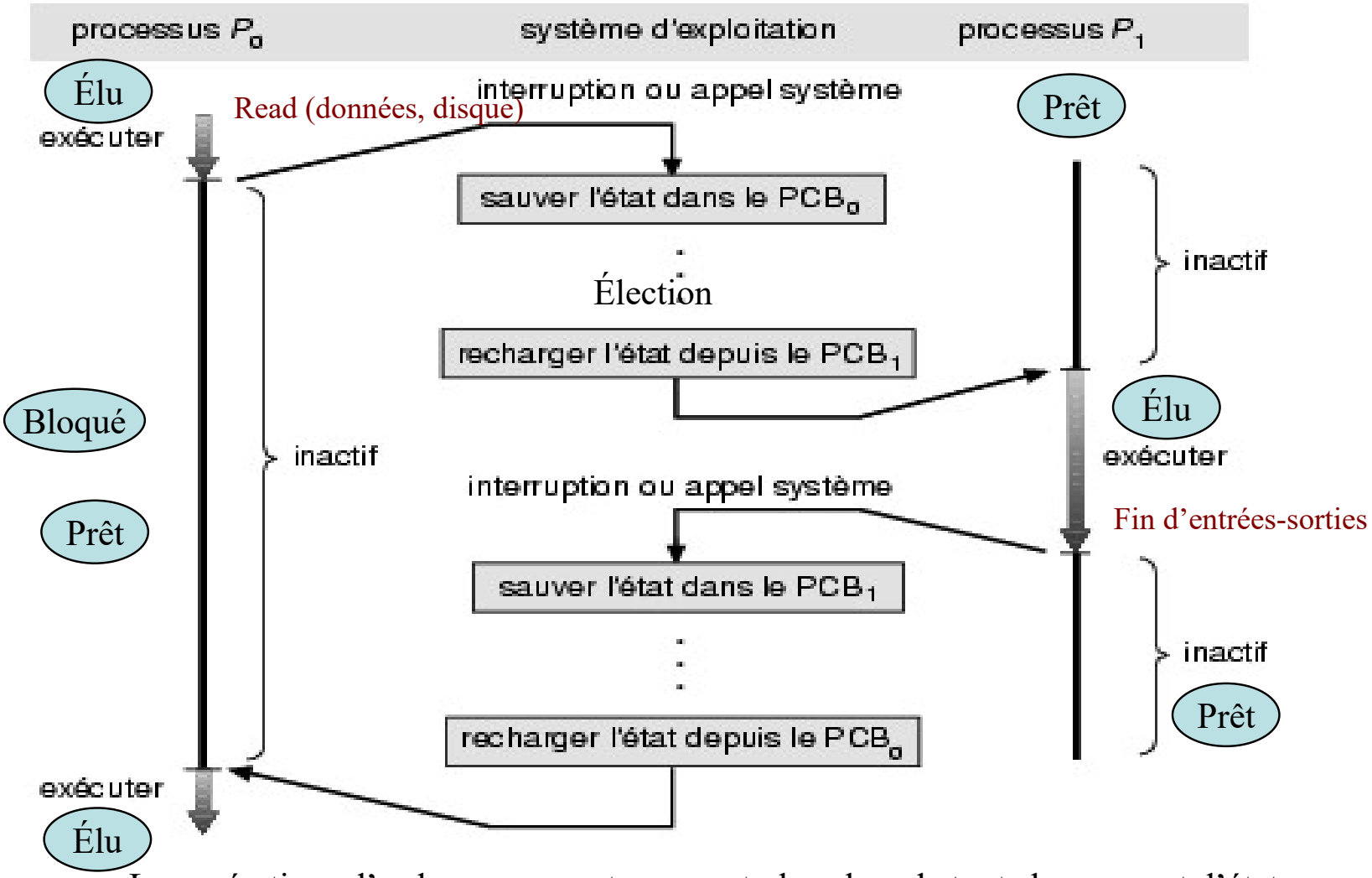


Transition élu->prêt – réquisition du processeur, préemption

Ordonnancement préemptif et non préemptif

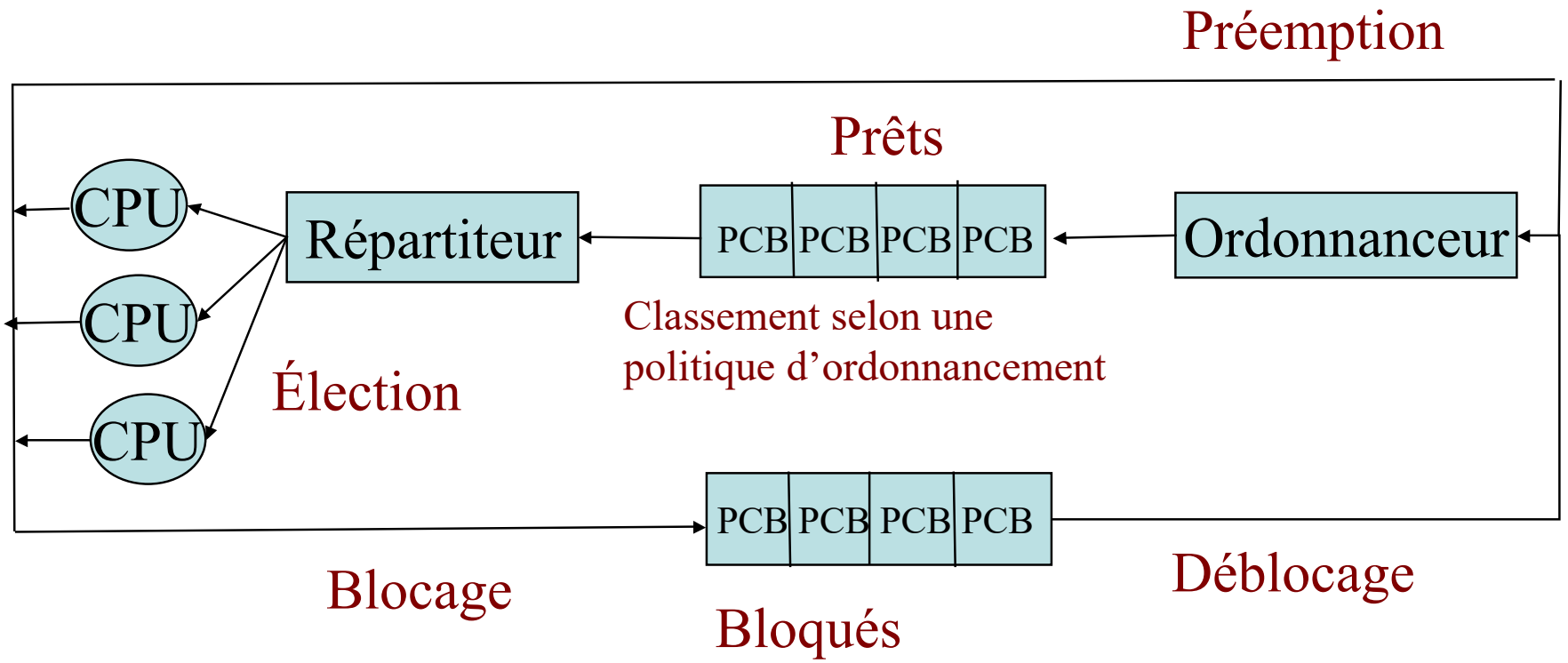
- Selon si l'opération de réquisition est autorisée ou non, l'ordonnancement est qualifié d'ordonnancement préemptif ou non préemptif
  - Ordonnancement non préemptif
    - Transition de l'état élu vers l'état prêt est interdite: un processus quitte le processeur s'il a terminé son exécution ou s'il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

d’ordonnancement



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

Entités systèmes 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 Fi qu'il existe de niveaux de priorité
  - Chaque file est gérée en tourniquet avec un quantum Qi
  - 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

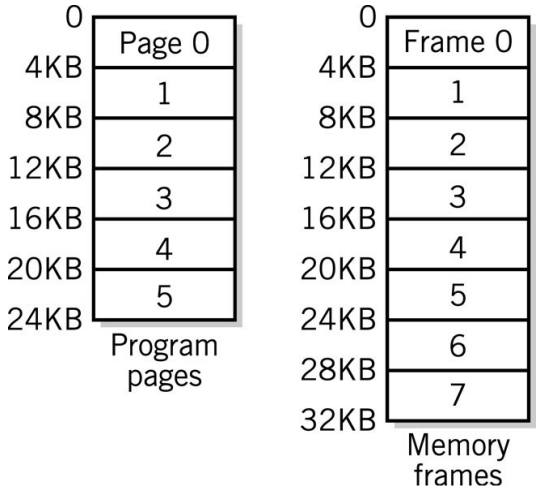
- Pagination
  - L'espace d'adressage du programme est découpé en morceaux linéaires de même taille appelés pages
  - L'espace de la mémoire physique est lui-même découpé en morceaux linéaires de même taille appelés case
    - Taille page = case – définie par le matériel (selon SE entre 512 octets et 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)

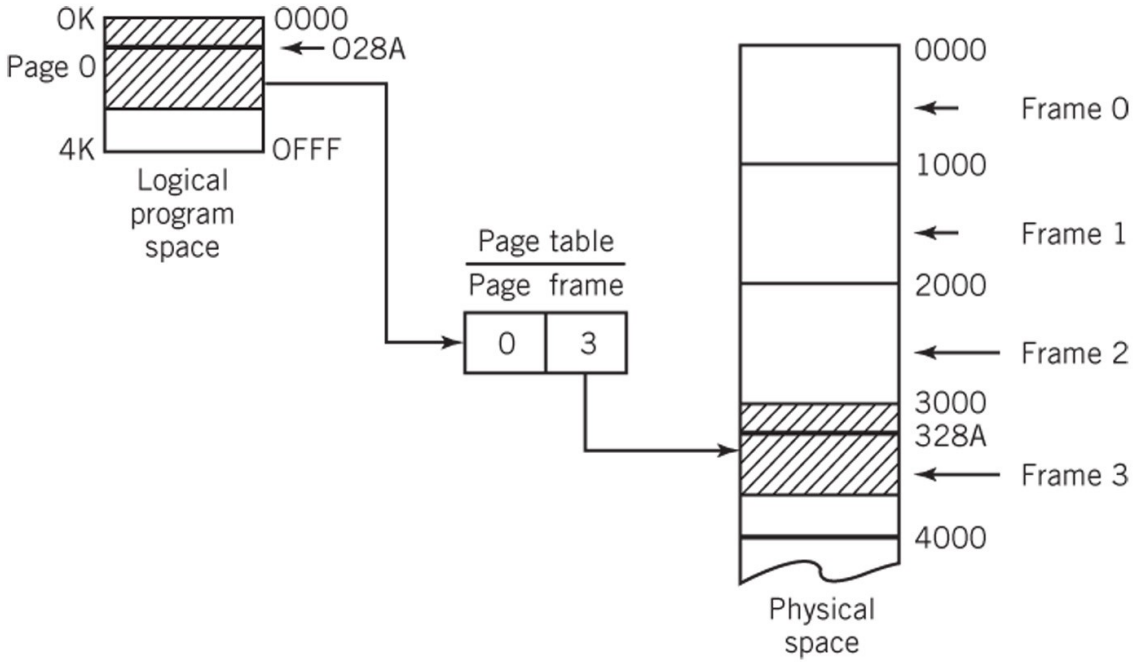
	Programme	Mémoire
Unité	Page	Case
Adresse	Logique	Physique
Taille de l'unité	2 to 4KB	2 to 4KB
Quantité	# de bits dans l'instruction	Mémoire installée

Pages et Cases (2)

Pages and Frames



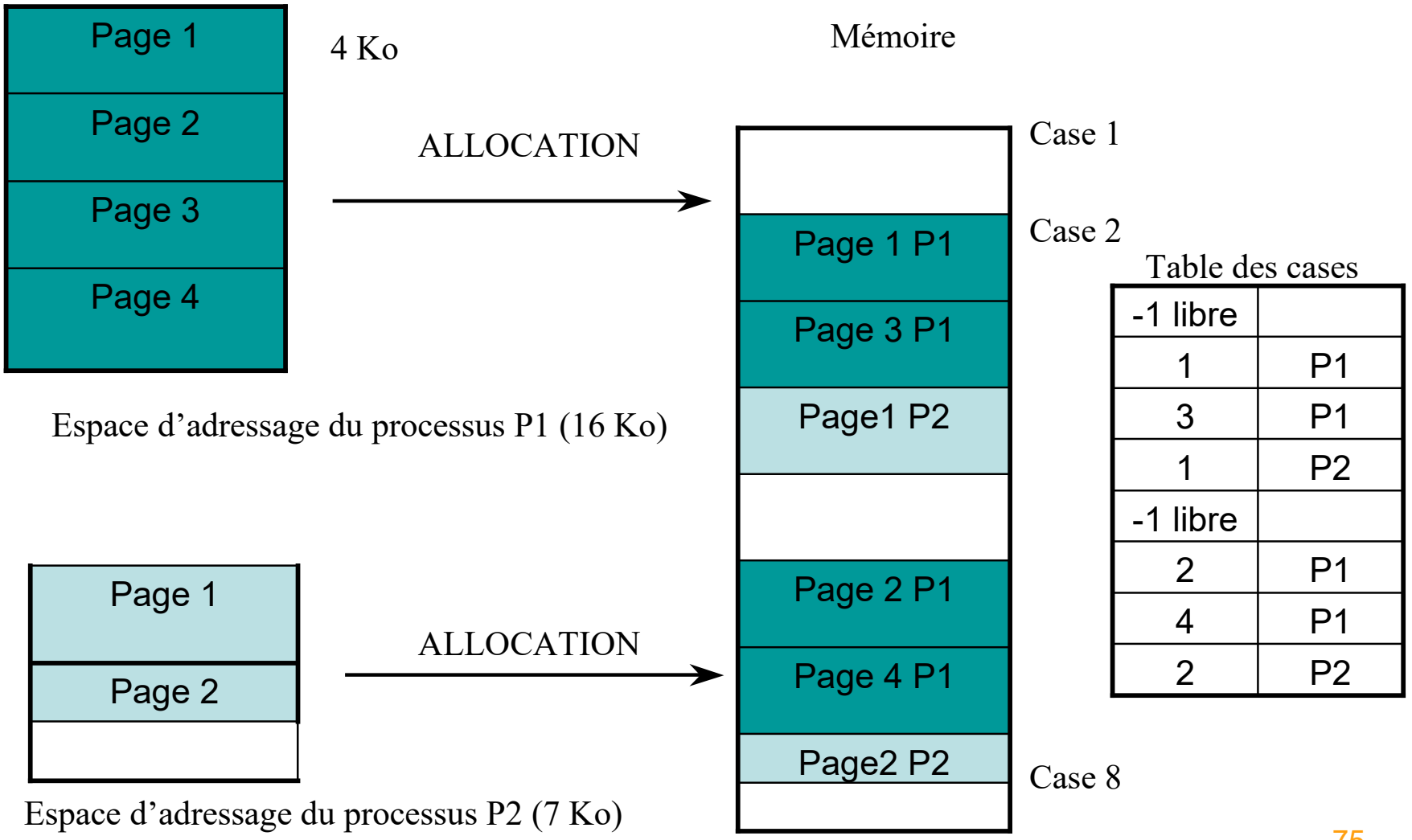
A Simple Page Table Translation



Each program has its collection of pages.

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

Gestion de la mémoire



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 correspond au numéro de page p et les n bits restants au déplacement d dans la page

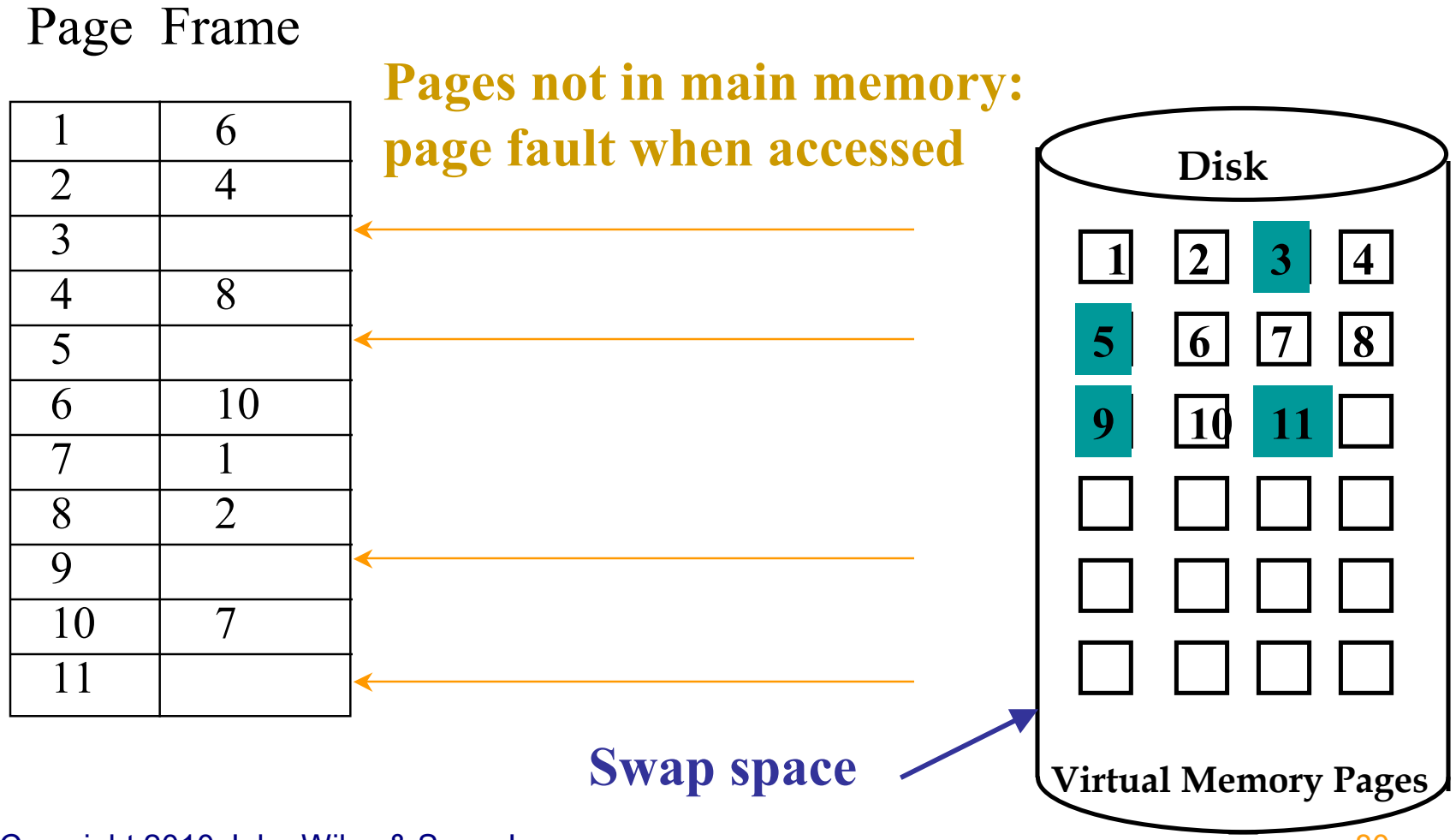
Gestion de la mémoire

- 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



Allocation de la mémoire pour 3 processus

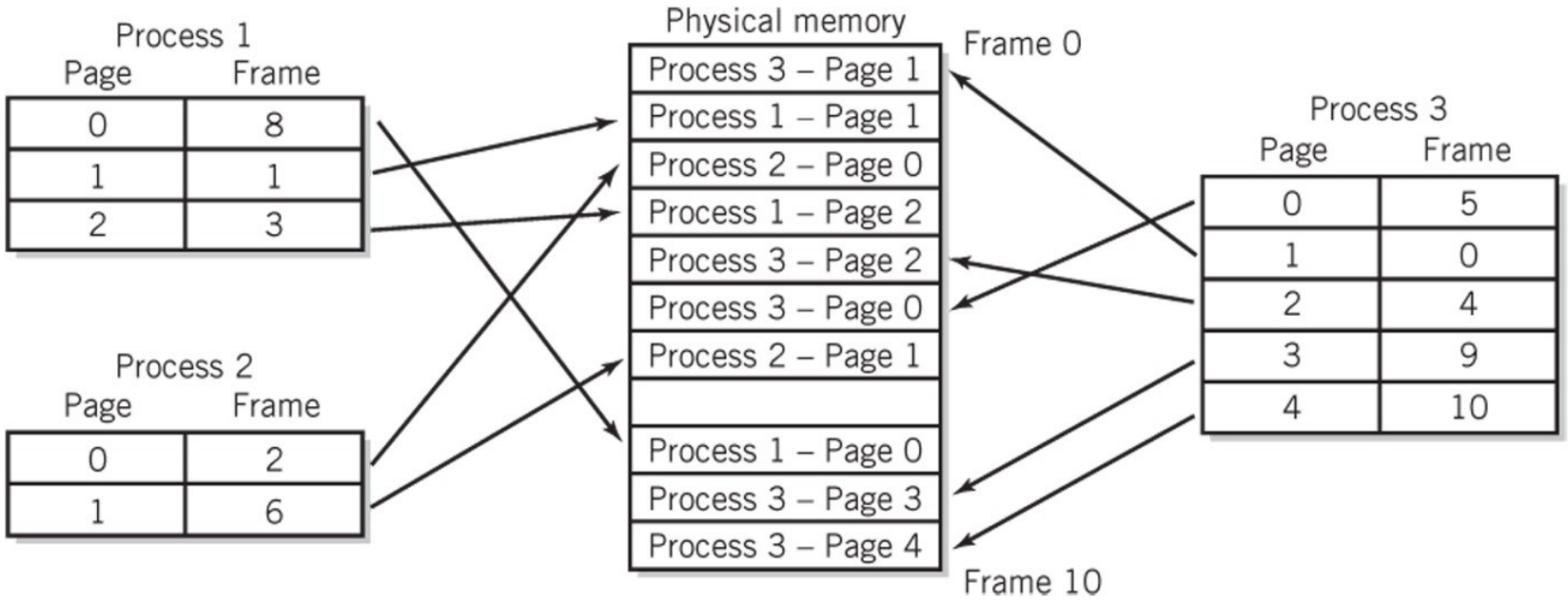
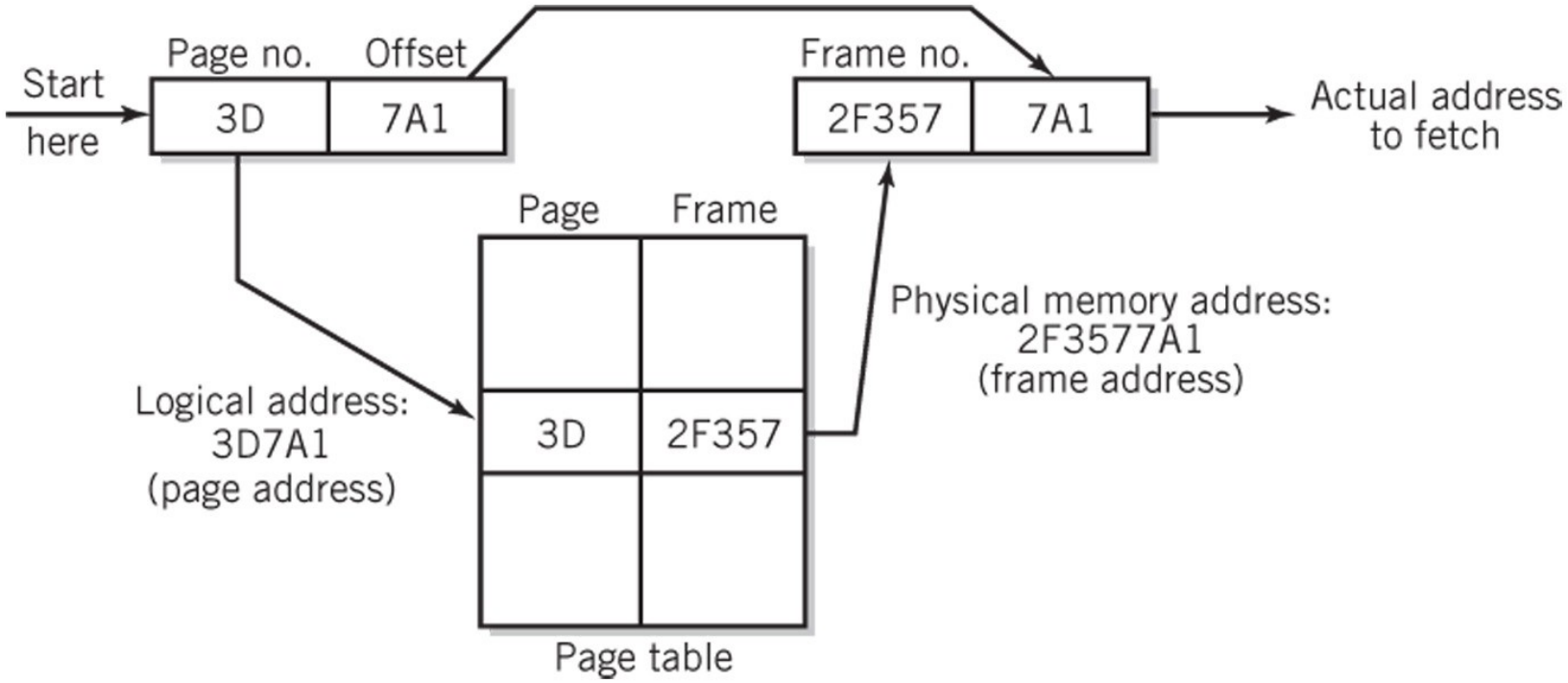


Table de cases

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			Free page frame
8	1	0	
9	3	3	
10	3	4	

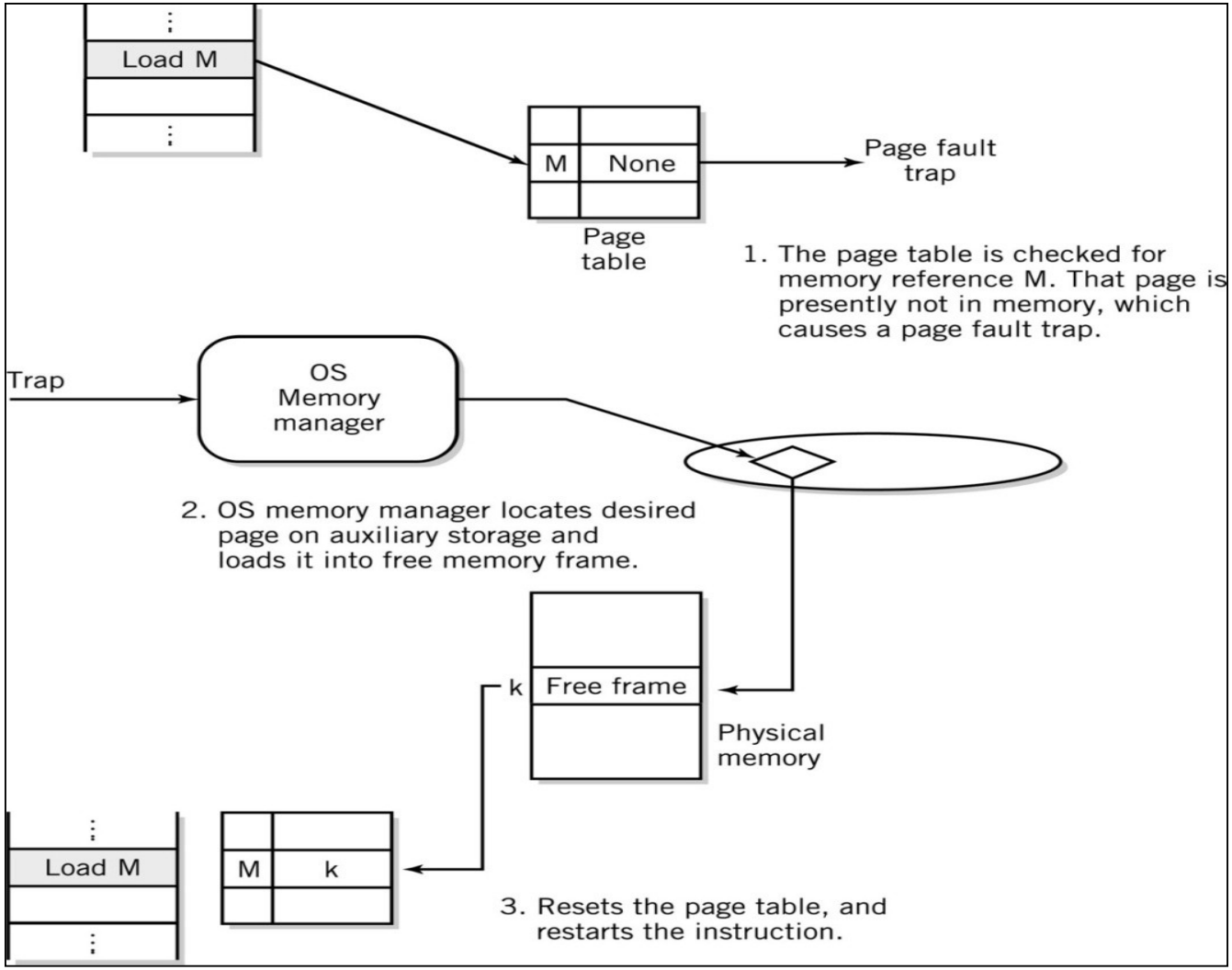
Translation d’adresse



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 tables 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

Étapes lors de défaut de page



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

Gestion de la mémoire

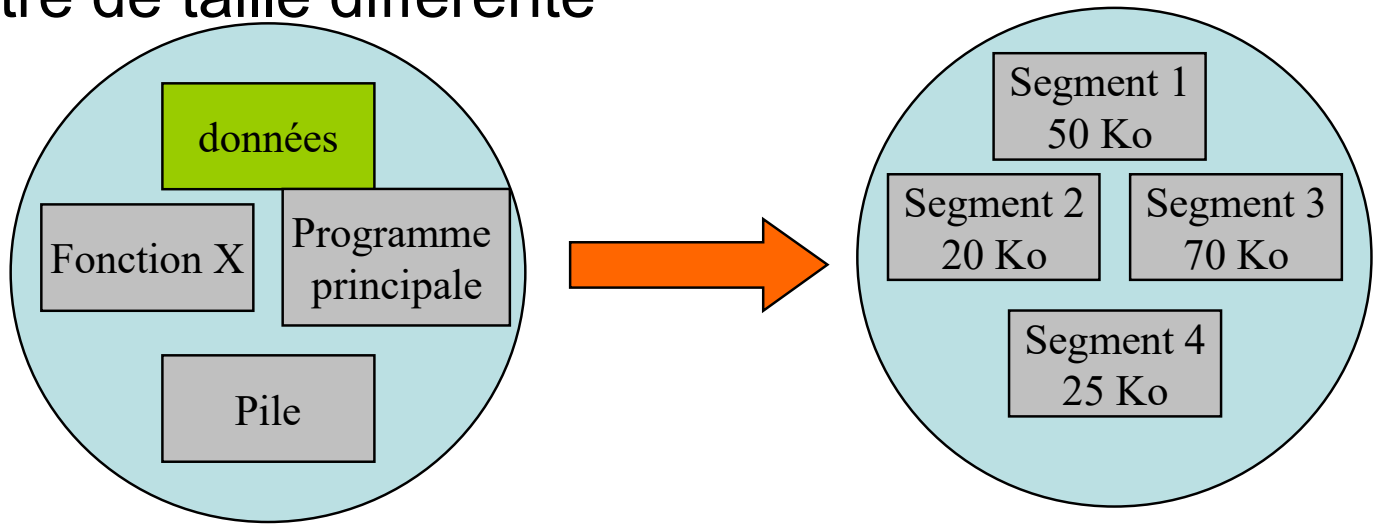
- Protection de l'espace d'adressage des processus
  - Lors d'un accès à une page, la cohérence du type d'accès avec les droits associés à la page est vérifiée et une trappe est levée par le SE si le type d'accès réalisé est interdit
  - Chaque processus ne peut avoir accès qu'à sa propre table des pages =>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

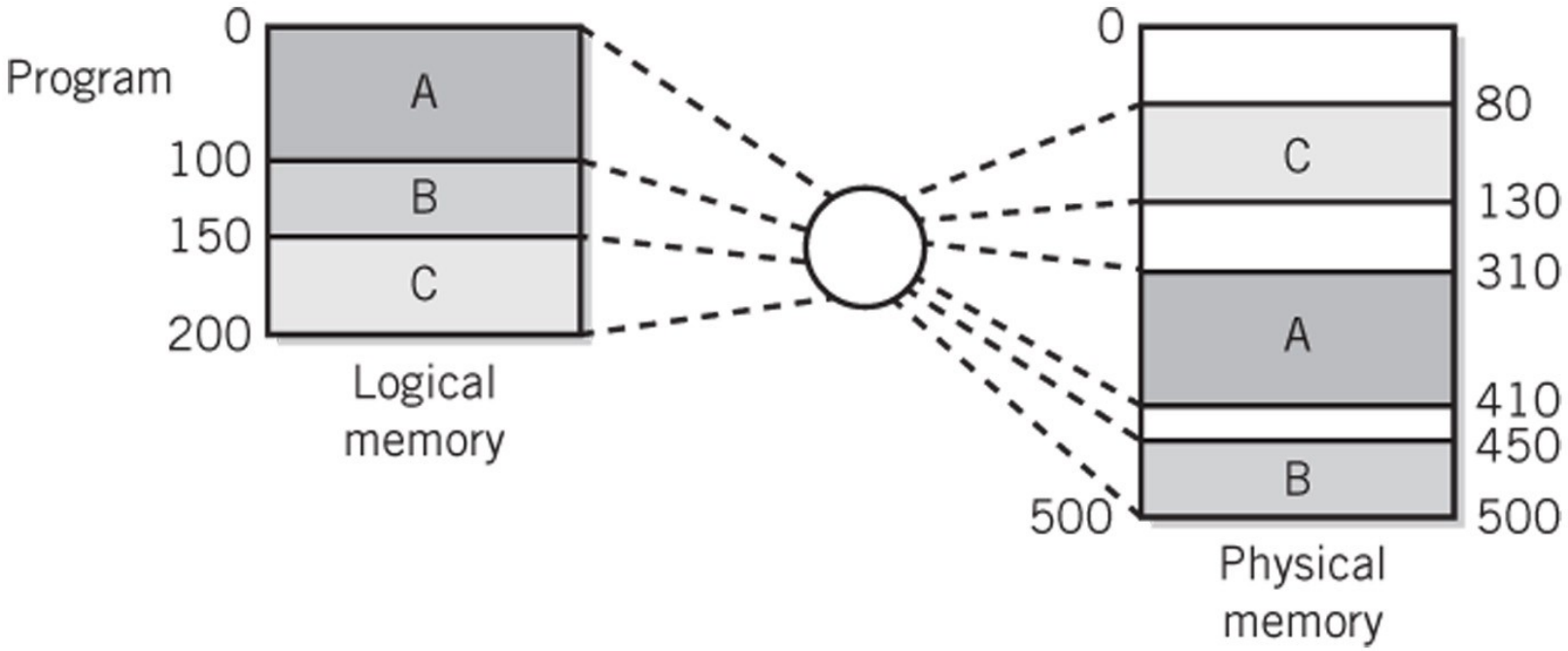
- Segmentation
  - La pagination constitue un découpage de l'espace d'adressage du processus qui ne correspond pas à l'image que le programmeur a de son programme
    - Données
    - Programme principal
    - Procédures séparées
    - Pile d'exécution
  - La segmentation est un découpage de l'espace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



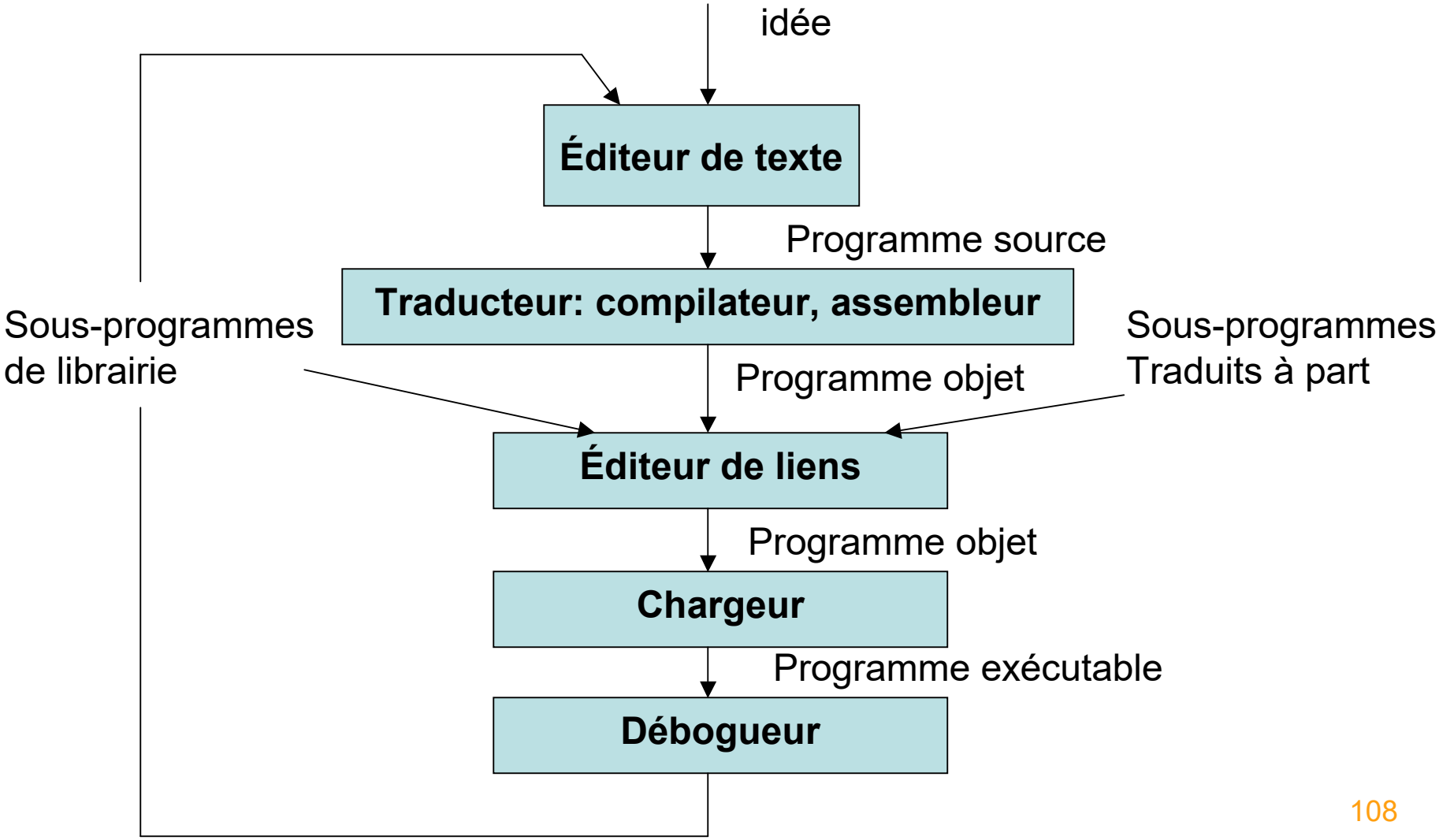
Segmentation



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 ≠ 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 relogeable
    - 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 relogeable
  - 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 relogeable
  - 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 relogeable 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