

Ordinateurs et systèmes

Chapitre 4 : Généralités sur les systèmes d'exploitation

Elaboré par : DhiaEddineSaied

Plan

- I. Introduction aux systèmes d'exploitation
- II. Historique
- III. Les systèmes d'exploitation
- IV. Classes des systèmes d'exploitation
- V. Éléments de base des systèmes d'exploitation

I. Introduction

- Les ordinateurs permettent de collecter des données, de réaliser des calculs, de stocker des informations et de communiquer avec d'autres ordinateurs.
- Un ordinateur est formé d'une partie matérielle et d'une partie logicielle. Cette dernière comporte des logiciels qui sont classés en deux catégories : les programmes d'application des utilisateurs et les programmes système qui permettent le fonctionnement de l'ordinateur. Parmi ceux-ci, le système d'exploitation (SE).
- Le système d'exploitation est le logiciel qui prend en charge les fonctionnalités élémentaires du matériel et qui propose une plateforme plus efficace en vue de l'exécution des programmes. Il gère les ressources matérielles, offre des services pour accéder à ces ressources et crée des éléments abstraits de niveau supérieur, tels que des fichiers, des répertoires et des processus.

III. Le système d'exploitation

1. Rôle et définition

Un système d'exploitation (SE) est présent au cœur de l'ordinateur coordonnant les tâches essentielles à la bonne marche du matériel.

C'est du système d'exploitation que dépend la qualité de la gestion des ressources (processeur, mémoire, périphériques) et la convivialité de l'utilisation d'un ordinateur.

III. Le système d'exploitation

1. Rôle et définition

Un SE résout les problèmes relatifs à l'exploitation de l'ordinateur en garantissant :

- Une gestion efficace, fiable et économique des ressources physiques de l'ordinateur (notamment les ressources critiques telles que processeur, mémoire...) : il ordonne et contrôle l'allocation des processeurs, des mémoires, des icônes et fenêtres, des périphériques, des réseaux entre les programmes qui les utilisent. Il assiste les programmes utilisateurs. Il protège les utilisateurs dans le cas d'usage partagé.
- Il propose à l'utilisateur une abstraction plus simple et plus agréable que le matériel : une machine virtuelle permettant l'interaction avec les utilisateurs en leur présentant une machine plus simple à exploiter que la machine réelle

III. Le système d'exploitation

2. Classes de systèmes d'exploitation

- **Mono-tâche (DOS)** : A tout instant, un seul programme est exécuté; un autre programme ne démarrera, sauf conditions exceptionnelles, que lorsque le premier sera terminé
- **Multi-tâches (Windows, Unix, Linux, VMS)** : plusieurs processus (i. e. un «programme» en cours d'exécution) peuvent s'exécuter simultanément (systèmes multiprocesseurs) ou en quasi- parallélisme (systèmes à temps partagé)

III. Le système d'exploitation

2. Classes de systèmes d'exploitation

- Mono- session (Windows 98,2000) : au plus un utilisateur à la fois sur une machine.

Les systèmes réseaux permettent de différencier plusieurs utilisateurs, mais chacun d'eux utilise de manière exclusive la machine (multi- utilisateurs, mono- session)

- multi- sessions (Windows XP, Unix, Linux, VMS) : Plusieurs utilisateurs peuvent travailler simultanément sur la même machine.

Ordinateurs et systèmes

Chapitre 5 : Gestion des processus

Plan

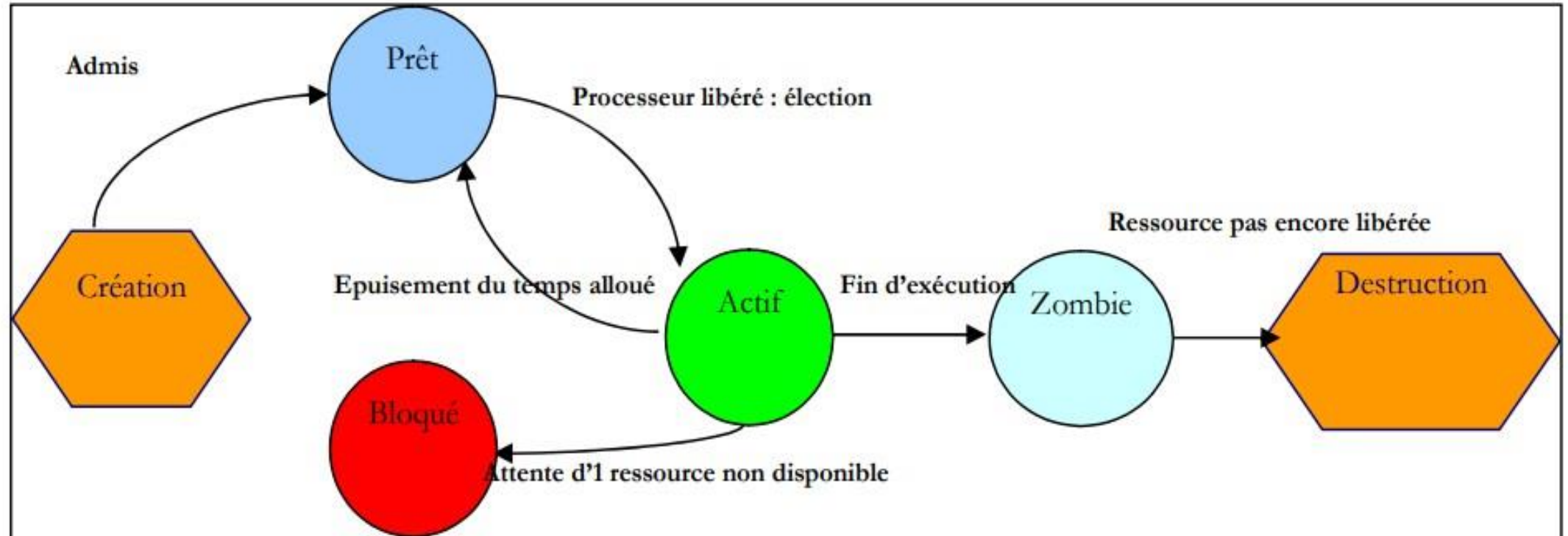
- I. Définition d'un processus
- II. . Caractéristique d'un processus
- III. Notion d'interruptions
- IV. Ressources et allocation de ressources
- V. Cycle de vie d'un processus

I. Définition d'un processus

Un processus est une entité dynamique correspondant à l'exécution d'une suite d'instructions : un programme qui s'exécute, ainsi que ses données, sa pile, son compteur ordinal, son pointeur de pile et les autres contenus de registres nécessaires à son exécution.

Attention : ne pas confondre un processus (aspect dynamique, exécution qui peut être suspendue, puis reprise), avec le texte d'un programme exécutable (aspect statique).

V. Cycle de vie d'un processus



Cycle de vie d'un processus

V. Cycle de vie d'un processus

Création : chargement des instructions, allocation de mémoires et des ressources (statiquement), il passe directement à l'état prêt.

Il existe 4 événements pour créer un processus :

- L'initialisation du système : au chargement du système il y'a création automatique du processus racine père de tous les processus utilisateurs (id=0)
- Un processus peut lancer un autre processus, il en devient le parent, l'autre dernier sera désigné comme processus fils. (Un processus père ne se termine que lorsque tous ses fils sont terminés. On a donc une structure arborescente de processus).
- Une requête de l'utilisateur
- Initiation d'un travail en traitement par lot

V. Cycle de vie d'un processus

La destruction d'un processus : Lors de la destruction le processus libère les ressources allouées. Il y a quatre causes possibles de la destruction d'un processus :

- Arrêt normal : volontaire, lorsque le processus termine sa tâche.
- Arrêt pour erreur : volontaire suite à une erreur pour une instruction illégale
- Arrêt pour erreur fatale : involontaire tel que les mauvais paramètres de l'exécution du processus
- Arrêt volontaire par un autre processus

L'état prêt : Le processus est prêt à être exécuté. Il est mis en attente jusqu'à ce qu'on lui libère le processeur (dispatch de l'Ordonnanceur), il passera alors à l'état Actif

V. Cycle de vie d'un processus

L'état actif ou élu :

- Le processus est en cours d'exécution par le processeur.
- Si le processus épuise le temps qui lui est alloué par le SE, il est remis en file d'attente des Prêts.
- Si il a besoin d'une ressource non disponible (opérations sur les périphériques), il est mis en attente prolongée (Interruption : état bloqué) jusqu'à la libération de la ressource nécessaire.
- Si le processus atteint son terme (se termine) il passe à l'état Zombie

L'état bloqué : Le processus est en attente d'une ressource pour terminer. Dès sa libération il repasse à l'état Prêt

L'état zombie : Le processus a terminé son exécution et il ne peut plus évoluer mais les ressources qu'il a allouées ne sont pas encore libérées

L'ordonnancement :

On appelle ordonnancement la stratégie d'attribution des ressources aux processus qui en font la demande. Différents critères peuvent être pris en compte :

- temps moyen d'exécution minimal
- temps de réponse borné pour les systèmes interactifs
- taux d'utilisation élevé de l'UC
- respect de la date d'exécution au plus tard, pour le temps réel, etc...

Ordinateurs et systèmes

Chapitre 6 :

L'ORDONNANCEMENT DES PROCESSUS

Plan

- I. Multitâche et ordonnancement des processus
- II. Critères et types d'ordonnancement
- III. Algorithmes d'ordonnancement

Introduction

Un ordinateur possède forcément plusieurs processus en concurrence pour l'obtention du temps processeur, cette situation se produit lorsque 2 ou plusieurs processus sont en état prêt simultanément.

L'Ordonnanceur (planificateur, scheduler) est la partie (un programme) du système d'exploitation responsable de régler les états des processus (Prêt, Actif,...etc.) et de gérer les transitions entre ces états ; c'est l'allocateur du processeur aux différent processus, il alloue le processeur au processus en tête de file des Prêts.

Objectifs d'un Ordonnanceur

Les objectifs d'un Ordonnanceur sont :

- Maximiser l'utilisation du processeur
- Présenter un temps de réponse acceptable
- Respecter l'équité entre les processus selon le critère d'ordonnancement utilisé.

Types d'ordonnancement

Il existe 2 types d'ordonnancement :

- a) **Ordonnancement préemptif** : Avec réquisition où l'Ordonnanceur peut interrompre un processus en cours d'exécution si un nouveau processus de priorité plus élevée est inséré dans la file des Prêts.
- b) **Ordonnancement coopératif** : Ordonnancement jusqu'à achèvement : le processus élu garde le contrôle jusqu'à épuisement du temps qui lui a été alloué même si des processus plus prioritaires ont atteint la liste des Prêts.

Les algorithmes d'ordonnancement

- L'algorithme FIFO (First In First Out)

L'ordonnancement est fait dans l'ordre d'arrivée en gérant une file unique des processus sans priorité ni réquisition : chaque processus s'exécute jusqu'à son terme ; le processus élu est celui qui est en tête de liste des Prêts : le premier arrivé. Cet algorithme est facile à implanter, mais il est loin d'optimiser le temps de traitement moyen

Les algorithmes d'ordonnancement

- L'algorithme SJF (Shortest Job First)

L'ordonnancement par ordre inverse du temps d'exécution (supposé connu à l'avance) : lorsque plusieurs travaux d'égale importance se trouvent dans une file, l'Ordonnanceur élit le plus court d'abord (les travaux les plus courts étant en tête de la file des prêts).

Cet algorithme possède l'inconvénient de la nécessité de connaissance du temps de service à priori et le risque de privation des tâches les plus longues. Afin de résoudre ces problèmes on pourra attribuer aux travaux une priorité croissante avec leur temps de séjour dans la file (temps d'attente). A temps d'attente égale, le travail le plus court est prioritaire. Toutefois, cette solution nécessite le calcul des priorités périodiquement et un réarrangement de la FA.

Performance des algorithmes d'ordonnancement

- Les files d'attente Rétroactives

Les performances d'un algorithme pour un ensemble de processus donné peut être analysée si les informations appropriées relatives aux processus sont fournies. Par exemple, des données sur l'arrivée du processus et sur l'heure d'exécution de ce processus sont nécessaires pour évaluer l'algorithme SRT.

Temps de rotation = Temps fin d'exécution - Temps d'arrivée

Temps d'attente = Temps de rotation – Durée d'exécution

$$\text{Temps moyen d'attente} = \frac{\sum \text{Temps attente}}{\text{nbre de processus}}$$

$$\text{Rendement} = \frac{\sum \text{Temps d'exécution}}{\text{nbre de processus}}$$