

Principes de fonctionnement des systèmes d'exploitation

Hainaut Patrick 2014

But de cette présentation

- La plupart des systèmes informatiques utilisent un système d'exploitation
- Il est donc primordial de connaître le fonctionnement de ce logiciel

GÉNÉRALITÉS

Qu'est ce qu'un OS

- Un système d'exploitation (Operating System -> OS) est un logiciel permettant la liaison entre la couche matérielle et la couche utilisateur

Couche utilisateur

Logiciels d'application

Système d'exploitation

Noyau (Kernel), gestion de la mémoire, gestion des processus, gestion des entrées/sorties, ...

Couche matérielle

Processeur, mémoire, cartes d'entrée/sortie, ...

Qu'est ce qu'un OS ?

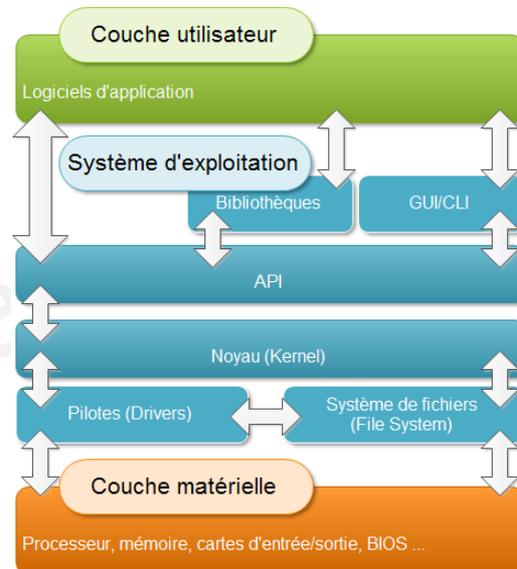
- C'est un programme indispensable de la machine:
 - Il permet une gestion des ressources de celle-ci
 - Il représente une base pour le développement et l'exécution des programmes d'applications et pour la virtualisation
- Remarque: Les premiers systèmes informatiques n'utilisaient pas d'OS. Les instructions du processeur devaient être reprises dans chaque programme

Terminologie

- Un programme (program) est une suite statique d'instructions, généralement stocké sur un support de masse (disque dur, ...)
- Un processeur (processor) est l'agent qui exécute les instructions d'un programme
- Un processus (process) est un programme en cours d'exécution, on passe donc d'un état statique à un état dynamique
- Un processus ou tâche (task) peut être composé de plusieurs tâches atomiques appelées thread

Architecture simplifiée d'un OS

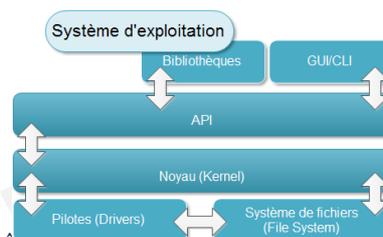
- Un OS peut être scindé en plusieurs éléments



©Hainaut P. 2014 - www.coursonline.be

Architecture simplifiée d'un OS

- *Pilote*: programme informatique permettant d'interagir avec un périphérique
Spécifique à l'OS employé
- *Système de fichiers*: les données doivent être conservées dans un périphérique de stockage. L'organisation des données sur le périphérique dépend du système de fichiers (FAT, NTFS, ...)
- *Noyau*: c'est le cœur de l'OS, il contient des fonctions fondamentales de gestion de la mémoire, des processus, des fichiers ...

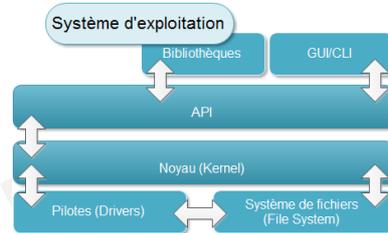


©Hainaut P. 2014 - www.coursonline.be

8

Architecture simplifiée d'un OS

- API (Application Programmable Interface): l'interface de programmation contient des fonctions permettant à l'utilisateur d'interagir avec le système



- Bibliothèques: ensembles de fonctions utilitaires pouvant être utilisées dans les applications utilisateurs

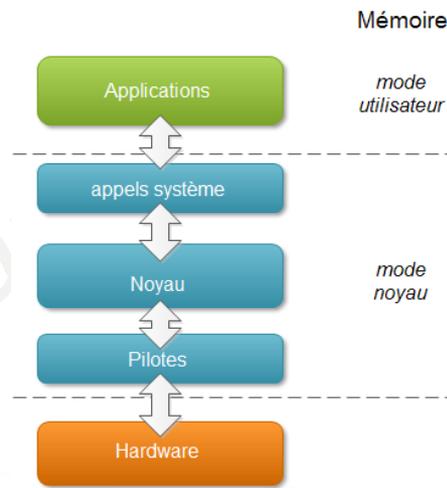
GUI/CLI (Graphical User Interface/Command Line Interface): environnement de travail graphique ou texte permettant d'exécuter des commandes

Modes privilégié et non-privilégié

- Pour pouvoir un accès non restrictif et privilégié aux ressources de la machine, l'OS s'exécute dans un mode privilégié, aussi appelé mode noyau (kernel)
- Les applications, devant passer par l'OS pour l'accès aux ressources (mémoire et matériel), s'exécutent dans un mode non-privilégié, aussi appelé mode utilisateur (user)
- Cela permet d'éviter bien des "plantages"
- Remarque: rien à voir avec les modes administrateur et utilisateur d'un OS

Modes privilégié et non-privilégié

- Une application ne peut jamais aller écrire dans la zone mémoire du noyau
- Ce n'était pas le cas en DOS, ce qui pouvait causer des "plantages" de la machine

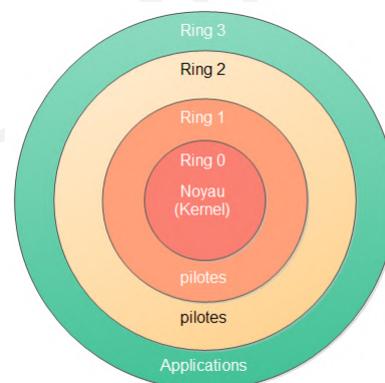


©Hainaut P. 2014 - www.coursonline.be

11

Modes privilégié et non-privilégié

- On peut aussi représenter le niveau de privilège par des anneaux, du plus privilégié (ring0) au moins privilégié (ring3)
- En pratique, le processeur et l'OS utilisent essentiellement le mode noyau et le mode utilisateur

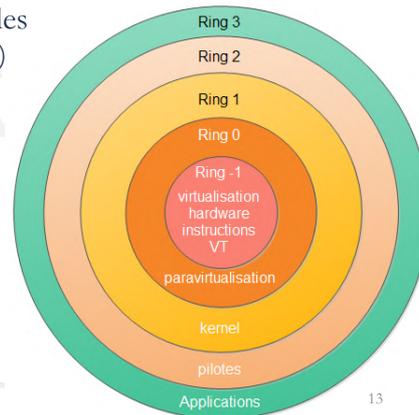


©Hainaut P. 2014 - www.coursonline.be

12

Modes privilégié et non-privilégié

- Cette notion d'anneau a évolué depuis qu'on utilise la paravirtualisation
- Les processeurs actuels intègrent des instructions de virtualisation (hard)
- La paravirtualisation optimise les performances de virtualisation mais exige un kernel modifié pour cela -> concerne surtout les OS libres
- La virtualisation sera abordée en détail dans une autre présentation



©Hainaut P. 2014 - www.coursonline.be

13

Modes privilégié et non-privilégié

- Remarque: dans le monde de l'embarqué, généralement, on ne s'encombre pas de mémoire virtuelle (vue plus loin) et de mode non privilégié sécuritaire, pour une question de place mémoire et d'efficacité dans l'accès aux ressources
- Dans ce cas, tout s'exécute dans le mode noyau (architecture MMU-less)
- Exemple d'OS utilisant ce mode: μ CLinux
- La programmation doit bien sur être impeccable car pas de protection de mémoire

©Hainaut P. 2014 - www.coursonline.be

14

Types d'OS

- OS monotâche: un seul processus est exécuté à la fois
Exemple: le DOS
- OS mono-utilisateur: un seul utilisateur à la fois peut se connecter
Exemple: le DOS
- OS multitâche: plusieurs processus peuvent s'exécuter en même temps
Exemple: Windows, Linux
- OS multi-utilisateurs: plusieurs utilisateurs peuvent se connecter en même temps
Exemple: Windows, Linux

Ordonnancement

- L'ordonnancement consiste à décider dans quel ordre effectuer les tâches de façon à utiliser au mieux la puissance du processeur
- Il existent plusieurs manières de prendre cette décision, gérée par des algorithmes (méthode destinée à résoudre un ensemble de problèmes, exemple: une recette de cuisine ...)
- On distingue, à ce propos, les OS temps réel des OS temps partagé

OS temps partagé (Time sharing OS)

- Présent sur la plupart des ordinateurs classiques
- Permet de partager le temps processeur entre plusieurs utilisateurs qui ont l'impression d'avoir la machine pour eux seuls
- Exemples: Windows, Linux
- Utilise un système de priorités adaptives

OS temps partagé (Shared Time OS)

- Remarque:
 - Le temps partagé se confond souvent avec le multitâche, bien que ce soit des notions légèrement différentes
 - Le temps partagé se rapporte aux utilisateurs tandis que le multitâche se rapporte aux processus
 - Mais les deux ont le même but, permettre au processeur de faire plusieurs choses "en même temps"
 - Dans le reste de notre propos, on regroupera les deux notions sous le terme "multitâche"

OS temps réel (Real Time OS -> RTOS)

- Les OS temps réel sont utilisés dans les systèmes embarqués (thermostats programmables, ...) , les robots, les automates industriels, ...
- Exemples: Windows CE, RTLinux
- Ces OS assurent qu'une tâche sera terminée dans un délai donné

Multitâche coopératif

- C'est la première forme de multitâche, la plus simple, qui n'est plus utilisée aujourd'hui (employée jusqu'à Windows 3.11 dans le monde Windows)
- Chaque tâche doit explicitement passer la main à la suivante
- Inconvénients:
 - Un processus peut être bloqué et le système entier se bloque
 - Le partage des ressources peut être inefficace (un périphérique mettant des données à disposition devra attendre que le processus pour les lire ait reçu la main)

Multitâche préemptif

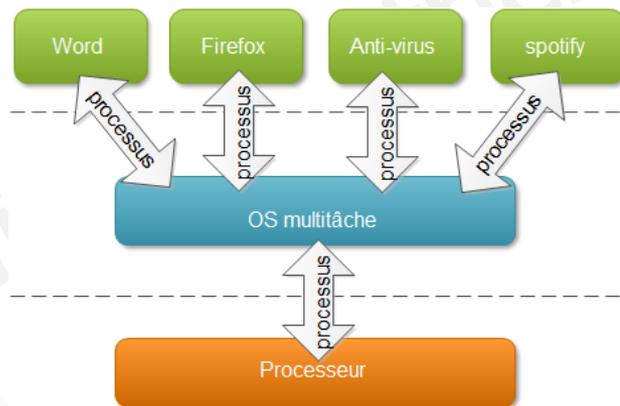
- Dans un système préemptif, un temps processeur défini est attribué à chaque processus
- Si le processus n'est pas terminé à la fin du temps accordé, l'ordonnanceur (scheduler) sauve son état (compteur ordinal, registres, ...) le replace dans la file d'attente (appelée pile) et exécute le processus suivant de cette file d'attente
- L'état d'un processus doit être rechargé dans le processeur pour que le code soit exécuté de nouveau : c'est la commutation de contexte

Multitâche préemptif

- Un système de priorités à plusieurs niveaux est mis en place
- Certaines tâches sont prioritaires par rapport à d'autres et doivent être exécutées avant
- De plus, un système d'interruptions permet lorsque l'interruption survient (lors d'un événement par exemple) d'interrompre la tâche en cours pour exécuter celle liée à l'interruption
- Contrairement à un système collaboratif, un processus bloqué ne bloque pas tout le système puisque ce n'est pas lui qui décide quand il rend la main

Multitâche et monocœur

- Quand une unité centrale (processeur) comprend un seul cœur, toutes les tâches sont exécutées par ce cœur

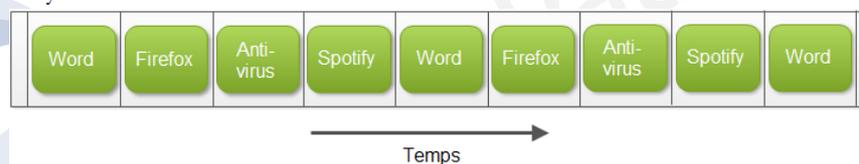


©Hainaut P. 2014 - www.coursonline.be

23

Multitâche et monocœur

- L'ordonnanceur permet au processeur d'exécuter chaque tâche pendant un temps défini avant de passer à la suivante et ainsi de suite avant de revenir à la première tâche et de recommencer le cycle



- Avec le matériel et les OS actuels, on a vraiment l'impression que toutes ces tâches (et les applications associées) sont exécutées simultanément (ce qui n'est pas le cas)
- C'est ce qu'on appelle le pseudo-parallélisme

©Hainaut P. 2014 - www.coursonline.be

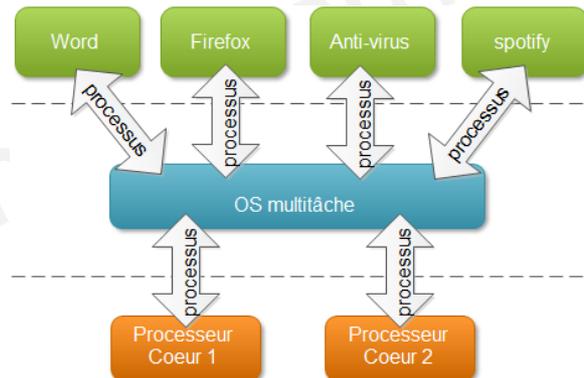
24

Multitâche et multicœur

- Un système multicœur diffère d'un système multiprocesseur par le fait que des ressources sont partagées (cache L2, chipsets, ...) et que par conséquent, le coût est moindre

- Les OS multitâches peuvent vraiment exécuter plusieurs tâches simultanément

- C'est le parallélisme

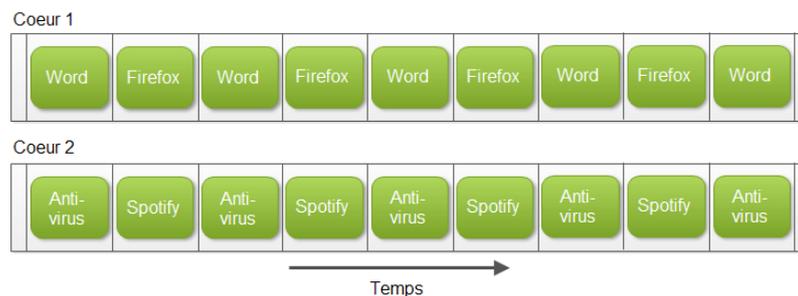


©Hainaut P. 2014 - www.coursonline.be

25

Multitâche et multicoeur

- Au lieu d'avoir une seule file d'attente, il y en a une par cœur



- On combine parallélisme et pseudo-parallélisme et on augmente ainsi les performances globales des applications

©Hainaut P. 2014 - www.coursonline.be

26

Multitâche

- Dans l'illustration des files d'attentes précédentes, il est indiqué le nom des applications par soucis de simplification mais ce sont en réalité des threads faisant partie de processus eux-mêmes reliés aux applications qui sont réellement exécutés par les processeurs
-> multithread
- Avant, lorsqu'il n'y avait que des processeurs monocœur, le processeur exécutait des processus et pas des threads
-> multitâche
- Pour les systèmes monoprocesseur monocœur, un thread = un processus

©Hainaut P. 2014 - www.coursonline.be

27

Multithread

- Le multitâche est mis en œuvre au sein de l'OS
- Le multithread lui est mis en œuvre au sein des applications
- On peut sous-diviser des opérations spécifiques au sein d'une même application en threads individuels
- Chaque thread peut fonctionner en pseudo-parallèle ou parallèle suivant le nombre de processeurs
- Remarque: actuellement, au niveau de l'OS on passe aussi en multithread (voir dia précédente)

©Hainaut P. 2014 - www.coursonline.be

28

Multithread

- Dans une application monothread, si on doit accéder à une ressource, l'application est bloquée pendant le temps nécessaire à cet accès
- Dans une application multithread, pendant que l'accès à la ressource s'exécute sur un thread, d'autres parties de l'application ne dépendant pas de cet appel peuvent s'exécuter sur d'autres threads
- On optimise encore plus ainsi l'utilisation du processeur

©Hainaut P. 2014 - www.coursonline.be

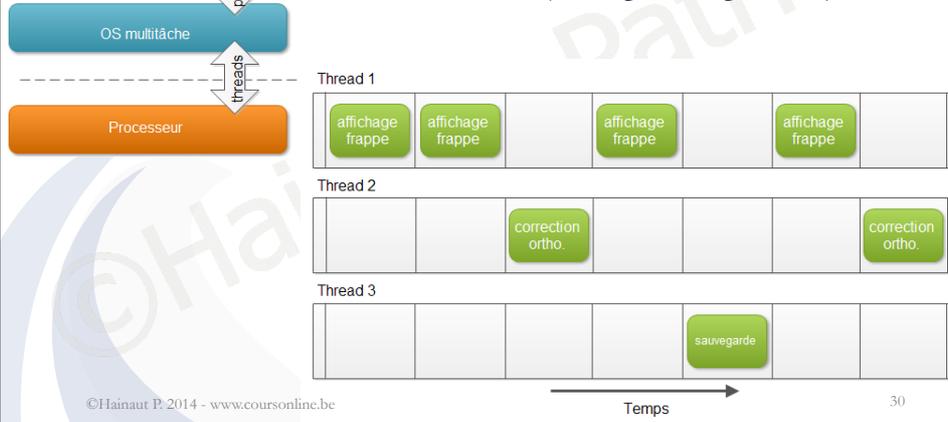
29

Multithread - exemple

Word



- L'application Word peut être divisée en plusieurs threads (ici en pseudo-parallèle)



©Hainaut P. 2014 - www.coursonline.be

30

Multithread - exemple

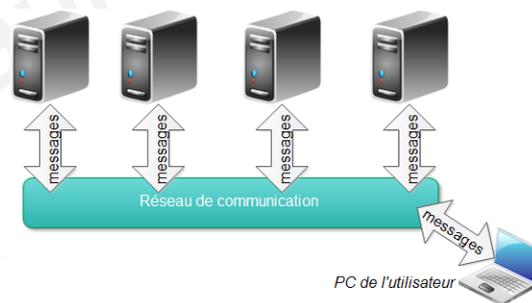
- On prend ici une seule application et un processeur monocœur
- On peut évidemment optimiser en prenant un processeur multicœur où les threads peuvent alors s'exécuter simultanément
- De même, on peut combiner le multithreading avec le multitâche
- C'est ce qu'on retrouve dans les systèmes actuels

©Hainaut P. 2014 - www.coursonline.be

31

Systemes distribués

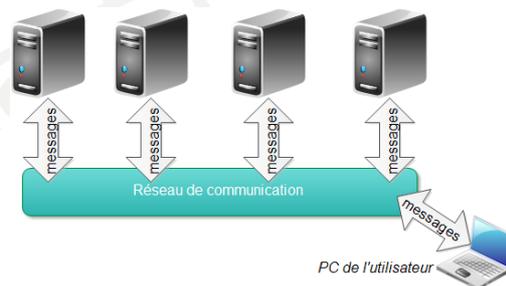
- Un système distribué est généralement un ensemble de machines autonomes connectées les une aux autres
- Chaque machine effectue des opérations contribuant à un projet global
- Pour l'utilisateur, cet ensemble de machines apparaît comme une machine unique



©Hainaut P. 2014 - www.coursonline.be

Systemes distribués

- Un logiciel spécifique appelé middleware est chargé de coordonner l'ensemble: découpage d'une tâche complexe en un ensemble de tâches simples qu'on répartit équitablement sur les machines et ensuite regroupement des résultats



©Hainaut P. 2014 - www.coursonline.be

33

Systemes distribués

- Internet est le plus grand système distribué et grâce à lui, il est facile de créer un système de calcul distribué
- Plusieurs projets ont vu le jour dont SETI@home de l'université de Berkeley (Californie, USA) et son évolution libre BOINC (Berkeley Open Infrastructure for Network Computing)
- Le projet de départ vise la détection d'une activité extra-terrestre SETI -> Search for Extra-Terrestrial Intelligence

©Hainaut P. 2014 - www.coursonline.be

34

Systemes distribués

- Le principe est d'utiliser la puissance de calcul non utilisé des ordinateurs connectés, et ce de façon transparente pour l'utilisateur de l'ordinateur
- On a pu ainsi décrypter une part importante du génome humain et d'autres projets sont en cours comme l'analyse des protéines et la recherche de solutions contre le sida
- Les réseaux peer to peer sont aussi des systèmes distribués



©Hainaut P. 2014 - www.coursonline.be

35

Clustering

- Comme alternative aux systèmes multiprocesseurs couteux, on peut constituer des grappes de serveurs (cluster)
- Ces serveurs indépendants, appelés nœuds (node) sont reliés entre-eux:
 - Soit pour apparaître comme un seul serveur avec plus de puissance processeur, plus de mémoire, ... pour former un serveur haute disponibilité
 - Soit pour former un système distribué

©Hainaut P. 2014 - www.coursonline.be

36

GESTION DES PROCESSUS

©Hainaut P. 2014 - www.coursonline.be

37

Introduction

- Comme dit plus haut, un processus est un programme en cours d'exécution par un ordinateur
- Il a besoin de temps processeur, d'espace mémoire et éventuellement de ressources
- Son exécution prend un certain temps avec un début et une fin
- Son exécution peut être déclenché par un utilisateur, par un évènement ou par un autre processus
- Les applications utilisateurs sont des ensembles de processus

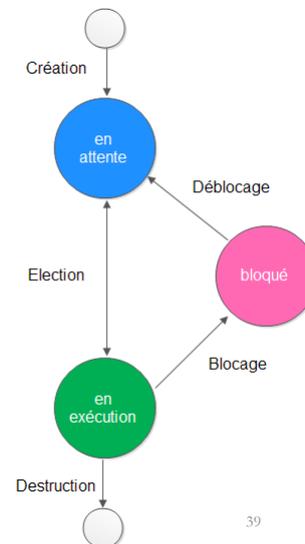
©Hainaut P. 2014 - www.coursonline.be

38

Etats simplifiés d'un processus

- Un processus est d'abord créé, il passe d'un état passif (programme) à un état actif (processus) en étant chargé en mémoire vive
- Il est ensuite placé dans la file d'attente gérée par l'ordonnanceur et attend son tour (et une commutation de contexte) pour être exécuté
- Quand vient son tour, il est exécuté (élu) par un processeur

©Hainaut P. 2014 - www.coursonline.be

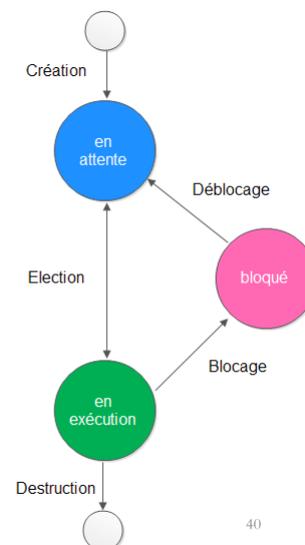


39

Etats simplifiés d'un processus

- Il peut être interrompu (événement, attente d'une ressource, ...) et passe dans un état bloqué (endormi)
- A la libération du blocage, il retourne dans la file d'attente
- Après le temps alloué à son exécution, s'il n'est pas terminé (pas encore de résultats), il retourne dans la file d'attente

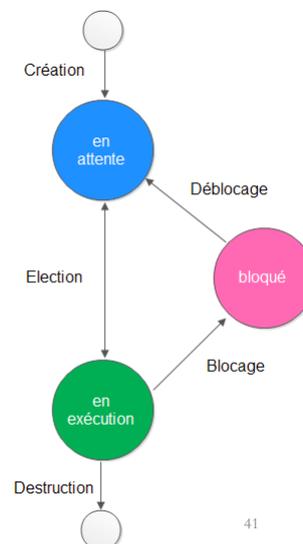
©Hainaut P. 2014 - www.coursonline.be



40

Etats simplifiés d'un processus

- En fin d'exécution, soit le résultat est produit, soit le processus est forcé de s'arrêter (via un signal)
- il est détruit avec tout ce qu'il mobilisait comme ressources (notamment espace mémoire) et retourne à l'état passif (programme)



©Hainaut P. 2014 - www.coursonline.be

41

PID

- Pour chaque processus créé, l'OS attribue un PID (Process Identifier) de manière à pouvoir identifier sans faille un processus (qui peut porter le même nom qu'un autre)
- Sous Windows, on retrouve ces PID dans le gestionnaire de tâches (task manager)
- (Il faut sélectionner dans Affichage, la colonne PID)

Gestionnaire des tâches de Windows - Affichage des processus :

Nom de l'image	PID	Nom d'...	Processeur	Mémoire...	Description
chrome.exe *32	5576	Patrick	02	98.680 K	Google Chrome
cmd.exe	1520	Patrick	00	948 K	Interpréteur de com
conhost.exe	1116	Patrick	00	1.472 K	Hôte de la fenêtre d
csrss.exe	424	Système	00	1.704 K	Processus d'exécuto
Dropbox.exe *32	2728	Patrick	00	62.700 K	Dropbox
dwm.exe	2908	Patrick	00	1.404 K	Gestionnaire de fenê
Edraw.exe *32	2780	Patrick	00	129.336 K	Edraw.exe
explorer.exe	2936	Patrick	00	40.172 K	Explorateur Window
InputPersonaliz...	2316	Patrick	00	2.104 K	Serveur de personni
jusched.exe *32	2644	Patrick	00	1.124 K	Java Update Schedu
notepad++.exe...	3100	Patrick	00	15.824 K	Notepad++ - a free
POWERPNT.EX...	588	Patrick	00	48.752 K	Microsoft PowerPoi
Screenpresso.e...	1300	Patrick	00	32.740 K	Screenpresso
splwow64.exe	3540	Patrick	00	2.216 K	Print driver host for
TabTip.exe	2680	Patrick	00	3.880 K	Tablet PC Input Pan
TabTip32.exe *32	2768	Patrick	00	620 K	Tablet PC Input Pan

©Hainaut P. 2014 - www.coursonline.be

Processus père et processus fils

- Au niveau de l'OS, certains processus dépendent d'un processus qui les a engendré, ce sont des processus fils et celui qui les a engendré est le processus père
- Si le processus père se termine avant le fils, le processus fils prend sa place

Communications inter-processus

- Nous verrons deux types de communications inter-processus:
 - Les pipes
 - Les sémaphores

Pipe

- Le pipe est une file de type FIFO (First In First Out) où le premier processus a accès en écriture et le deuxième en lecture
- Ça permet d'enchaîner des commandes
- Exemple:
En Linux la commande **ps aux** permet de voir les processus actifs et la commande **grep** permet de filtrer les résultats
L'enchaînement `ps aux | grep dhcpd` ne montre que les processus actifs liés au serveur DHCP

Sémaphores

- Dans le cas de ressources partagées, si une de ces ressources est monopolisée par un processus, il faut l'indiquer aux autres processus de manière à ce qu'ils attendent avant d'utiliser la ressource, qu'elle soit libérée
- C'est le problème du dîner des philosophes posé par Edsger Dijkstra dont une solution possible est l'utilisation de sémaphores (proposée également par Dijkstra)

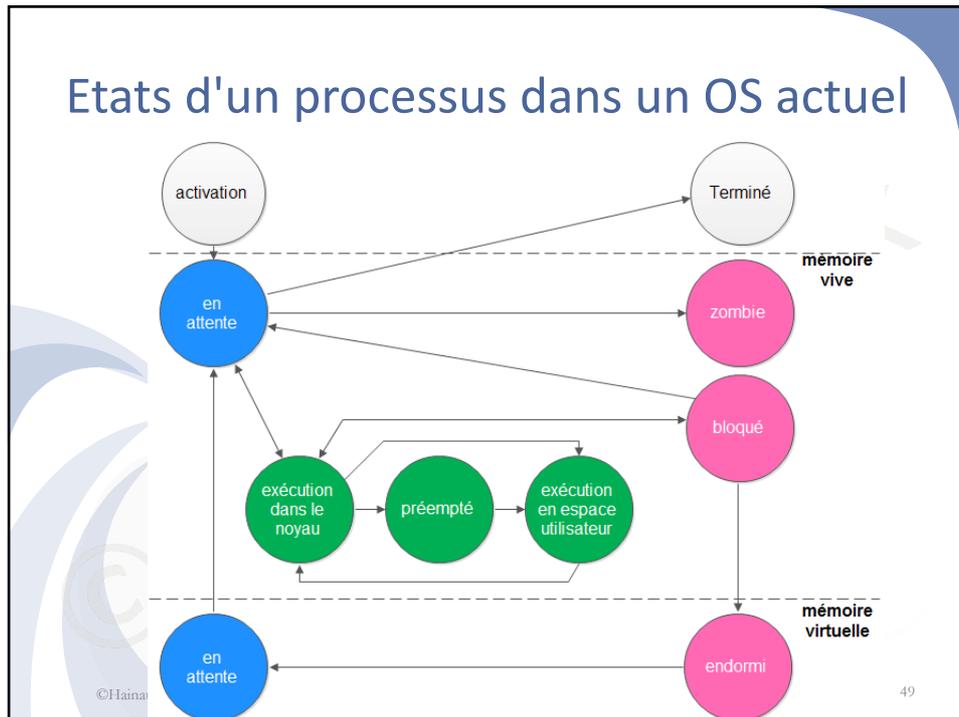
Sémaphores

- Un sémaphore est une variable qui accepte deux opérations P et V
- P permet de tester le sémaphore, si sa valeur est > 0 , elle est décrétementée de 1, sinon le processus est mis en attente car la ressource est déjà utilisée
- V permet d'incrémenter la valeur du sémaphore, rendant une ressource disponible
- Ces opérations sont exclusives, aucune autre opération ne peut avoir lieu en même temps sur le sémaphore (pour éviter des exécutions concurrentes)

Rôle du système d'exploitation

- L'OS est chargé d'allouer les ressources matérielles (mémoire, temps processeur, entrées/sorties) nécessaires aux processus et faire en sorte qu'un processus n'interfère pas avec un autre (isolation)
- Il doit en outre permettre la communication entre processus
- Si le système dispose de plusieurs processeurs ou d'un processeur multicœur, il doit répartir équitablement les processus entre ces éléments
- Il doit gérer l'ordonnancement des processus (multiplexage)

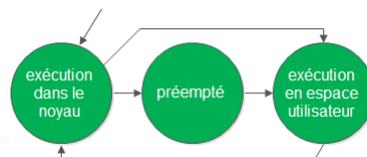
Etats d'un processus dans un OS actuel



Etats d'un processus dans un OS actuel

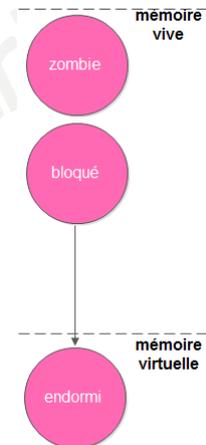
- Etats supplémentaires:

- Exécution dans le noyau: le processus est exécuté dans un espace protégé et réservé à l'OS et donc accès à l'entièreté de l'espace mémoire
- Préempté (swappé): le processus est suspendu par l'ordonnanceur et copié en swap (pour libérer la mémoire vive)
- Exécution en espace utilisateur: c'est le cas de tous les processus provenant d'applications, l'espace mémoire accessible est limité à celui réservé pour l'application pour éviter les plantages d'autres applications ou du système



Etats d'un processus dans un OS actuel

- Etats supplémentaires:
 - Zombie: si un processus terminé ne peut être déchargé de la mémoire (par exemple parce qu'un processus fils est toujours en cours d'exécution), il passe à l'état de zombie



©Hainaut P. 2014 - www.coursonline.be

51

Mémoire vive et swap

- Les OS utilisent deux types de mémoire:
 - La mémoire vive qui est la mémoire de travail dans laquelle doit être stocké un processus lorsqu'il est en cours d'exécution
Elle est d'un accès rapide
 - La mémoire swap qui est constitué par un espace particulier sur le disque dur (généralement une partition distincte)
Son accès est plus lent et elle ne contient que des processus endormi ou dans la file d'attente
Si un processus est appelé alors qu'il est en swap, il sera bloqué par l'ordonnanceur le temps de son transfert en mémoire vive
 - L'ensemble de ces deux mémoires constitue la mémoire virtuelle

©Hainaut P. 2014 - www.coursonline.be

52

Mémoire vive et swap

- La mémoire virtuelle permet d'exécuter simultanément plus de processus que ce que la mémoire vive peut contenir
- Chaque processus n'ayant besoin que d'une partie des informations qu'il manipule en mémoire vive, le reste est stocké en swap
- Quand le processus a besoin d'informations résidant en mémoire virtuelle, ces informations sont transférées en mémoire vive avant traitement

Organisation de la mémoire

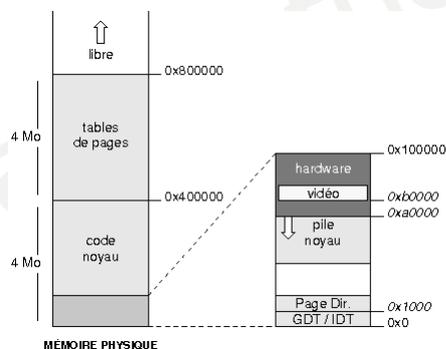
- La mémoire virtuelle est découpée en zones, appelées pages, de même taille (4 ko en Linux)
- Une adresse virtuelle est donc composée d'un numéro de page et d'un emplacement dans cette page
- Un ordinateur travaillant en binaire (deux valeurs: 0 et 1), la taille des pages est une puissance de 2
- La mémoire vive est découpée en zones de même taille, appelées cadres

Organisation de la mémoire

- Chaque cadre peut donc accueillir une page
- Il peut bien sur avoir plus de page que de cadres (c'est ce qu'on veut)
- Un mécanisme de translation assure la correspondance entre les adresses virtuelles et les adresses physiques
- L'adresse physique obtenue est donc composée d'un numéro de cadre et d'un emplacement dans ce cadre

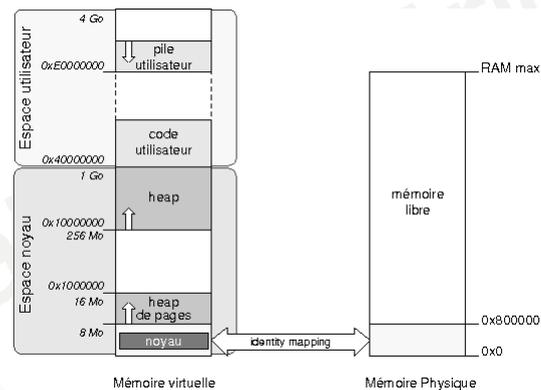
Espace d'adressage des processus

- Exemple d'utilisation de la mémoire physique sous Linux



Espace d'adressage des processus

- Exemple d'organisation de la mémoire virtuelle sous Linux



©Hainaut P. 2014 - www.coursonline.be

57

Méthodes d'ordonnancement des processus (algorithmes préemptifs)

- Il existe plusieurs méthodes d'ordonnancement des processus
- Nous en verrons brièvement 3:

le round-robin
le SJF
le CFS

©Hainaut P. 2014 - www.coursonline.be

58

Round-Robin ou méthode du tourniquet

- Le Round-Robin est un algorithme qui attribue les mêmes tranches de temps à chaque processus, sans priorité spécifique
- C'est comme si les processus étaient sur un tourniquet et passaient à tour de rôle devant le processeur, pendant un temps fini

Round-Robin ou méthode du tourniquet

- Règles:
 - un nouveau processus est ajouté en fin de liste
 - l'utilisation du processeur par un processus ne peut pas dépasser un certain laps de temps
 - l'attente maximum est donnée par la multiplication du nombre de processus en cours multiplié par le laps de temps accordé à chaque processus
 - un processus qui vient de finir d'utiliser le processeur (laps de temps écoulé) est préempté et placé en fin de liste
 - un processus qui a terminé son travail est sorti de la liste, par conséquent le temps d'attente pour les autres processus diminue

Round-Robin ou méthode du tourniquet

- Chaque fois qu'on change de processus, le processeur doit procéder à une commutation de contexte, ce qui prend du temps
- Il faut donc que le laps de temps accordé à chaque processus ne soit pas trop petit sinon on perd trop de temps en commutation de contexte
- Il ne faut pas non plus qu'il soit trop long sinon certains processus risquent de ne jamais être exécutés -> situation de famine (référence aux diners des philosophes)

SJF (Shortest Job First)

- Cet algorithme va exécuter d'abord le processus le plus court de la file d'attente
- Si un processus entre dans la file d'attente avec un temps d'exécution est plus court que le reste du temps d'exécution du processus en cours d'exécution, il y a alors commutation de contexte, le processus entrant prend la place de l'autre qui est replacé dans la file d'attente
- Cet algorithme est réservé aux environnements spécialisés car il faut évaluer précisément le temps d'exécution des processus

CFS (Completely Fair Scheduler)

- C'est, depuis 2007, l'ordonnanceur du noyau Linux
- L'algorithme trie les processus selon une valeur représentative du manque de ces processus en temps d'allocation du processeur, par rapport au temps qu'aurait alloué un processeur dit multitâche idéal, sur lequel tous les processus s'exécuteraient en même temps et à la même vitesse
- L'ordonnanceur exécute alors le processus le plus en manque

Conclusion

- Avec cette présentation, certes assez théorique, vous êtes à même de percevoir un peu mieux le fonctionnement des systèmes d'exploitation
- Il reste maintenant à passer à la partie pratique ...
- Merci de votre attention