

Systeme d'exploitation

Gestion des processeurs et de la mémoire



Qu'est-ce qu'un processus ?

A quoi sert la mémoire RAM ?



Sommaire

De quoi s'agit-il ?

- 01** Notion de processus
- 02** Gestion de la mémoire et ordonnancement
- 03** L'approche GNU/Linux
- 04** Gestion des processus avec PowerShell
- 05** Démarrage et services



Notion de processus



Programme (rappel)



La recette

Séquence d'instructions machines (binaire)

Stocké dans une mémoire (vive ou stockage)

Statique

Destiné à être exécuté par un processeur comprenant ces instructions



Fonctionnement de l'ordinateur



Retour en bas

Un CPU peut exécuter 1 code machine (programme) à la fois

- Exécute une opération unitaire
- Passe à la suivante (en boucle)

Comment partager le (ou les) processeurs entre les différents programmes (et notamment le noyau) ?



Une première idée

Chacun son tour

Au démarrage, on lance le système d'exploitation (le noyau)

Puis le noyau lance un programme utilisateur

Puis le programme utilisateur relance le noyau...

Inconvénients :

- Et si le programme ne rend pas la main ?
- Efficacité ?
- Réactivité ?



Optimiser l'utilisation du CPU



Différentes
échelles de temps

Accès à une information sur le disque (ou autre)

- Lent (par rapport au CPU)
- Le programme doit attendre
- Gaspillage de temps CPU ?



Le problème de réactivité



Réagir aux actions
de l'utilisateur

Cas particulier des applications interactives

- Utilisateur déclenche une action
- Attendre la fin de l'action pour pouvoir faire autre chose ?
- Même en cas de lancement d'un calcul long ?



Systeme à temps partagé

Simuler des
exécutions
simultanées

Systeme d'exploitation orchestre l'usage des processeurs

- Lancement d'un programme (à la demande de l'utilisateur)
- Interrompre un programme et récupérer la main
- Relancer un programme en attente

Gestion de processus - Ordonnancement.

Egalement appelé “**Multitâche préemptif**”.



La notion de processus

Un programme en
cours d'exécution

Conceptuellement :

- Programme en cours d'exécution

Techniquement :

- Structure de donnée du système d'exploitation
- Informations permettant la gestion



Mode processeur et interruptions

Soutien matériel

Processeur dispose d'un mode protégé (restriction d'accès à certaines instructions sensibles) :

- Certaines instructions interdites
- Force l'appel au système pour accéder au matériel

Mécanisme d'interruptions (matérielles) :

- Lancement automatique de routine de traitement
- Lors de certains événements matériels
- Horloge
- Commutation de contexte



Et les données alors ?

Partager une
mémoire

Mémoire (rappel) :

- Suite de mots mémoire (*byte*)
- Adresse

Processus stocke ses données en mémoire

- Accès à toute la mémoire ?
- Comment garantir qu'un processus n'accède (et ne modifie) pas les données d'un autre processus (ou du système) ?



Gestion de la mémoire et ordonnancement



Protection de la mémoire

Abstraire la
mémoire

Mémoire virtuelle

- Adresses virtuelles \Leftrightarrow physiques (MMU - [Memory Management Unit](#))
- Segmentation et/ou Pagination
- Allocation par page - Table des pages
 - Associée à des droits d'accès
 - Peuvent être déportées sur du stockage (swap)
 - Plus de mémoire disponible qu'en réalité



Allocation d'un processus

Pas si anodin

Exécutable stocké sur disque => processus s'exécutant en mémoire

- Allocation de mémoire (code, données statiques, pile, tas)
- Copie du code en mémoire



Allocation d'un processus (suite)

Pas si anodin

- Création d'un PCB (*Process Control Block*)
 - Identifiant de processus
 - État - priorité
 - Adresses zones mémoires
 - Sauvegarde du contexte
 - Autres informations (métriques)
- Insertion dans table des processus



Ordonnanceur

À qui le tour ?

À chaque interruption d'un processus le système doit décider quel processus en attente il affecte sur le processeur

Politique d'ordonnancement :

- En fonction de l'état des processus
- Gestion des priorités
- Objectifs : réactivité, optimisation, équité...

Ex : FIFO (First In First Out), Tourniquet...

CFS (Completely Fair Scheduler) : Ordonnanceur principal linux



Gestion du swap

Swapper oui, mais comment ?

Quand décharger les pages mémoires en swap ?

- Attendre d'avoir besoin de place
 - Retrouve un problème de choix ([FIFO, LRU, autre...](#))
- Décharger la mémoire avant
 - En fonction de l'état des processus

Tentative d'accès à une page sur disque => Défaut de page

- Mise en attente du processus
- Rechargement de la page en mémoire



Les threads

Programmation
parallèle

Constats :

- Création d'un processus coûteux
- Communication entre processus délicate

Les processus légers (*Threads*) :

- Plusieurs fils d'exécutions concurrents (parallèles)
- Au sein d'un même processus
- Zones mémoires partagées



L'approche GNU/Linux



Les métadonnées

Caractéristiques

PID : identifiant du processus

PPID : identifiant du processus parent

CMD : commande de lancement

UID : identifiant utilisateur associé

GID : identifiant groupe associé

TTY : terminal d'entrée/sortie

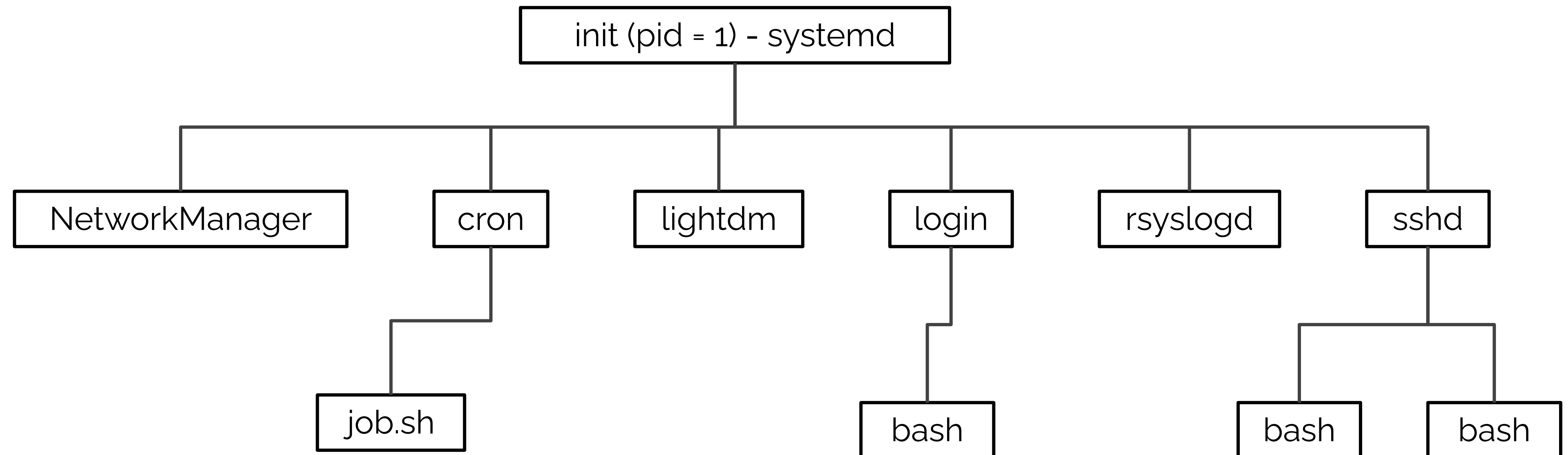
État, répertoire de travail, priorité, métriques

Correspond fichiers dans /proc



Les processus linux

Je suis ton père !





Messages standards

Communication
inter-processus

Un processus peut envoyer un signal à un autre processus

Par exemple :

- SIGINT : demande d'interruption (ctrl + c dans un terminal)
- SIGTERM : demande d'arrêt
- SIGKILL : destruction du processus par le noyau
- SIGTSTP : mise en pause (ctrl + z dans un terminal, reprise avec fg)
- SIGQUIT : arrêt et core dump (ctrl + \ dans un terminal)
- SIGSEGV : erreur de segmentation (accès mémoire interdit)

La commande **kill** permet d'envoyer des signaux



Quelques commandes

Voir et agir

ps : liste les processus

pstree : arborescence de processus

top : liste les processus par consommation CPU

htop : liste les processus interactive

kill / **killall** : envoi de signaux aux processus (arrêt)



Quelques commandes (suite)

Voir et agir

fg : passe un processus en premier plan (avec reprise si nécessaire)

bg : reprise d'un processus en pause avec maintien en arrière plan

<command> & : lancement d'une commande en arrière plan (rappel)

nohup (pour *no hang up*): lancement d'un processus détaché de la session utilisateur



Exemples d'utilisation de nohup

Voir et agir

Exécution standard d'un script avec nohup :

nohup ./monScript.sh &

Exécution d'un script avec nohup et redirection de la sortie standard (stdout) et la sortie d'erreur (stderr) vers un fichier nommé "monLog.log" :

nohup ./monScript.sh > monLog.log 2>&1 &



Exemples d'utilisation de nohup (suite)

Voir et agir

Exécution d'un script avec nohup et redirection de la sortie standard (stdout) et la sortie d'erreur (stderr) vers /dev/null. Cela signifie que toutes les sorties du script seront supprimées et ne seront pas écrites.

```
nohup ./votre_script.sh > /dev/null 2>&1 &
```



Des commandes de planification de tâches

Quand exécuter

cron : exécution de processus récurrents (démon)

crontab : commande de configuration de cron

at / atq / atrm / batch : lancement de tâches planifiées



Fichiers et partitions d'échange

Mémoire virtuelle

Le swap est constitué :

- Partitions dédiées (classique)
- Fichiers dédiés

/etc/fstab contient les partitions de swap à monter au démarrage

free : utilisation mémoire et swap

swapon / swapoff : activation des espaces de swap

mkswap : initialise un·e fichier/partition de swap



Gestion des processus avec PowerShell



Les métadonnées

Caractéristiques

Id : identifiant du processus (équivalent de PID)

ParentId : identifiant du processus parent (équivalent de PPID)

UserName : nom de l'utilisateur associé (équivalent de UID)

Group : groupe associé (équivalent de GID)

Handles : Nombre de handles ouverts par le processus (peut être assimilé à l'équivalent de fichiers ouverts)

State : état actuel du processus (en cours d'exécution, etc.)

Path : chemin d'accès du répertoire de travail du processus



Quelques Cmdlets

Avec PowerShell

Get-Process : récupération d'info sur les processus

Stop-Process : arrêter un processus

Wait-Process : attendre la fin d'un processus

Start-Process : lancement d'un processus local

Invoke-Command : lancement d'une commande local ou distante

Get-ComputerInfo : récupération d'informations matérielles et système

Get-CimInstance : récupération d'objets WMI (*Windows Management Instrumentation*)



Démarrage et services



Le boot à l'ancienne

MBR ou UEFI
legacy

BIOS/UEFI démarre

- Choix du périphérique de stockage de démarrage (configuration)

MBR/GPT

- Le secteur de démarrage contient le bootloader (emplacement conventionnel)

Le bootloader

- Connaît le·s système·s de fichiers
- Décide quel noyau (OS) démarrer



Le boot UEFI

UEFI

UEFI démarre

- Choix du périphérique de stockage de démarrage (configuration)
- Ce périphérique contient une ESP (EFI System Partition)
- Cette partition contient :
 - Firmware UEFI
 - Bootloader

Le bootloader

- Décide quel noyau (OS) démarrer



Secure boot



UEFI secure boot

Idée générale : Interdire de démarrer des noyaux *compromis*.

Mis en place par Microsoft

- UEFI avec secure boot ne peut démarrer que des loader signés
- Mécanisme de *signature numérique*



Amorçage

Au début

L'amorçage GNU/Linux : **GNU grub**

- grub-legacy (sans esp)
- grub2
- Permet de démarrer n'importe quel système

L'amorçage Windows : **Bootmgr.exe**

- Permet uniquement de démarrer Windows

Plus (beaucoup) d'info sur l'amorçage sur le [site de Rod Smith](#).



Amorçage du système

Et après le noyau ?

Démarrage d'un premier processus

- En charge de démarrer tous les autres

Services (démons ou *daemons*) :

- Processus démarrés lors de l'amorçage de la machine
- Toujours chargés



Le démarrage GNU/Linux : Systemd

The new init

Nouveau système de démarrage GNU/Linux

- Gestion des dépendances de services
- Démarrage parallèle

Principale commande d'administration : **systemctl**



Le démarrage Windows : Boot manager

Le Windows boot

Le fichier bootmgr.exe

- Peut se trouver à la racine du c:\
- Peut être dans une partition NTFS système (boot partition)

Au boot:

- Il lit les données du BCD (Boot Configuration Data)
- Execute winload.exe → charge les pilotes du système et l'OS
- Lancement du SCM (*Service Control Manager*) qui lance les services de démarrage en base de registre



Le démarrage Windows : Boot manager

Avec Powershell

Get-Service : liste et état des services

Stop-Service : arrêter un service

Start-Service : démarrer un service

Suspend-Service : mettre un service en pause

Restart-Service : redémarrage d'un service

Set-Service : modifier un service

Plus d'info : [Gérer les services](#)



En résumé

A retenir

Gestion de processus et mémoire
L'amorçage (boot)
Notion de service/démon



MERCI

pour votre participation.

C'est à vous maintenant.

Des questions ?

Des remarques ?

