

# Plan du cours

- 12. Gestion des processus
  - 12.1 Notions théoriques
  - 12.2 Lancement d'une commande
  - 12.3 Identifier et contrôler un processus
    - PS, KILL
  - 12.4 Exercices

# Plan du cours

- 12. Gestion des processus
  - 12.1 Notions théoriques
    - 12.1.1 Processus
    - 12.1.2 Processus père et processus fils
    - 12.1.3 Identification d'un processus
    - 12.1.4 Temps partagé
    - 12.1.5 Swapping (va et vient)
    - 12.1.6 Classification des processus

# Plan du cours

- 12. Gestion des processus
- 12.2 Notions théoriques
  - 12.2.1 Le mode interactif
  - 12.2.2 Le mode en arrière-plan
  - 12.2.3 Le mode différé
  - 12.2.4 Le mode batch
  - 12.2.5 Le mode cyclique

# • 12. Gestion des processus

## • 12.1 Notions théoriques

- **Le processus :**

- Un processus est un **programme en cours d'exécution**. Les attributs d'un processus appartiennent à ce que l'on appelle son **environnement :**

- **Le code, les données temporaires, les données permanentes, les fichiers associés, les variables.**

- 12. Gestion des processus

- 12.1 Notions théoriques

- L'environnement du processus contient aussi les données que lui attribue le système :
- **Les descripteurs, la mémoire allouée, la pile d'exécution du noyau.**

# • 12. Gestion des processus

## • 12.1 Notions théoriques

- **Le processus père et processus fils**

- Un processus fils est un processus qui a été créé par un autre processus (processus père).

- **Identification d'un processus**

- Chaque processus a un **PID** unique (**Process Identifier**) attribué par le système.

# • 12. Gestion des processus

## • 12.1 Notions théoriques

### • **Temps partagé et Multitâches**

- Le **temps partagé** est une approche permettant de simuler le partage par plusieurs utilisateurs de temps processeur.
- **Temps partagé** se réfère à l'utilisation d'un ordinateur par plusieurs utilisateurs, tandis que **multitâche** évoque plus largement le déroulement simultané de processus multiples sans accorder d'importance spéciale au nombre d'utilisateurs.

# • 12. Gestion des processus

## • 12.1 Notions théoriques

### • **Swapping (va-et-vient)**

- Le **swapping** consiste en la recopie sur disque d'un processus complet ou d'une partie d'un processus ayant perdu le contrôle de l'unité central et ne pouvant plus rester en mémoire centrale.
- La mémoire centrale ainsi libérée est affectée à un processus plus prioritaire.
- La qualité du swapping est très importante pour la performance d'un système.

- 12. Gestion des processus
- 12.1 Notions théoriques
  - **Classification des processus**
  - Processus système (daemons)
    - Les plus courants sont :
      - **Init** processus parent de l'interpréteur de commandes créés par la connexion d'un utilisateur (**PID = 1**).
      - **Cron** permet l'exécution de programmes en mode cyclique.
      - **Xinetd** super démon internet chargé de créer les processus serveurs réseau sur requête des clients.
- Processus utilisateurs

- 12. Gestion des processus
- 12. Gestion des processus
- 12.1 Notions théoriques
  - **Classification des processus**
  - Processus utilisateurs
    - Toute exécution d'un programme par l'utilisateur
    - Le premier est l'interpréteur de commandes shell à la connexion.

- 12. Gestion des processus

- 12.2 Exécution d'une commande

- **Les 5 modes d'exécution**

- Le mode interactif (le plus utilisé)

- En mode console

- **<ctrl-c>** pour arrêter le processus

- **<ctrl-z>** pour le suspendre

# • 12. Gestion des processus

## • 12.2 Exécution d'une commande

### ➔ **Les modes d'exécution**

#### ➤ Le mode en arrière-plan

- En mode console, permet de rendre immédiatement le contrôle à l'utilisateur

#### • **Exemple :**

- firefox &

- **Pbe:** si on quitte la console, les processus lancés en arrière plan sont arrêtés

#### • **Solution : nohup**

# • 12. Gestion des processus

## • 12.2 Exécution d'une commande

### ➔ **Les modes d'exécution**

#### ➤ Le mode différé

➤ On peut lancer une commande à une heure donnée à l'aide de la commande `at`

➤ **`at 20:05 20/01/2009 <commande`**

➤ Les fichiers de configuration : **`/etc/at.allow`**, **`/etc/at.deny`**

➤ Voir commandes : **`atq`**, **`atrm`**

# • 12. Gestion des processus

## • 12.2 Exécution d'une commande

### ➔ **Les modes d'exécution**

#### ➤ Le mode batch

- Le but est de placer une commande dans une file d'attente. Toutes les commandes lancées par **batch** sont exécutées séquentiellement.
- Commande apparente à **at**

# • 12. Gestion des processus

## • 12.2 Exécution d'une commande

### ➔ **Les modes d'exécution**

#### ➤ Le mode cyclique

- Très utilisé par l'administrateur système
- Commandes : **crontab**, **cron** (daemon)
- Fichiers de configuration : **/etc/cron.allow**, **/etc/cron.deny...**
- **crontab [options] file**
- Zone spool du crontab : **/var/spool/cron**

# • 12. Gestion des processus

## • 12.2 Exécution d'une commande

### ➔ **Les modes d'exécution**

#### ➤ Le mode cyclique

➤ Editer votre crontab

➤ Crontab -e

➤ Ligne(s) de votre crontab :

➤ Minute(0-59) Heure(0-23) Jour(1-31) Mois(1-12)  
JourSemaine(0-6, 0:dimanche) Tache\_a\_exécuter

➤ **Exemple:**

➤ 59 0 \* \* 1-6 sauvegarde\_journaliere\_du\_lundi\_au\_samedi

# • 12. Gestion des processus

## • 12.3 La commande PS

• `ps [options]` : Affiche l'état des processus en cours

### ➤ **Les options**

➤ **-u** nom : affiche les processus de l'utilisateur nom

➤ **-e** : renseignements de tous les processus en cours

➤ **-C** nom : affiche les renseignements du processus nom

➤ **-f** : génère pour chaque processus

• `ps -eo "%p %y %x %c"`

# • 12. Gestion des processus

## • 12.3 La commande PS

- `ps [options]` : Affiche l'état des processus en cours
  - Les options
  - **-f** : génère pour chaque processus
    - No du processus (**PID**)
    - No processus père (**PPID**)
    - Heure de lancement du processus (**STIME**)
    - Le nom du terminal (**TTY**)
    - Temps d'exécution du processus (**TIME**)

# • 12. Gestion des processus

- 12.3 La commande PS
- **ps [options]** : Affiche l'état des processus en cours
  - Exemples
  - ps -ef
  - ps -eo "%u %p %P %t %y %x %c"

# • 12. Gestion des processus

- 12.3 La commande KILL
- **kill -signal PID** : permet d'arrêter les processus en arrière-plan
- L'option -signal correspond à un signal envoyé au processus.
- Ce mécanisme permet la communication interprocessus, notamment entre le système d'exploitation et le processus.

# • 12. Gestion des processus

## • 12.3 La commande KILL

• Les principaux signaux sont :

- **SIGHUP** SIGnal Hang Up : fin du shell
- **SIGINT** SIGnal INTerrupt : interruption du programme
- **SIGKILL** SiGnal KILL : tuer le processus
- **SIGTERM** SIGnal TERMinate : terminaison douce
- **SIGQUIT** SIGnal QUIT : terminaison brutale
- **SIGSTOP** SIGnal STOP : stopper le processus

# • 12. Gestion des processus

## • 12.3 La commande KILL

- D'autres signaux existent et sont décrits dans signal.
- Kill sans option -signal envoie le signal **SIGTERM**
- Le signal **SIGQUIT** est plus brutal
- Le signal **SIGKILL** ne peut être ignoré, il arrête systématiquement un processus.
- A effectuer dans cette ordre :
  - **Kill pid**
  - **Kill -QUIT pid**
  - **Kill -KILL pid**

- 12. Gestion des processus

- 12.3 La commande KILL

- **Sortie de session** : le bash envoie à chaque processus attachés à la session le signal **SIGHUP**.
  - On veut qu'un processus en arrière-plan ne soit pas interrompu
  - **nohup commande &**

# • 12. Gestion des processus

## • 12.4 Exercices

### • 12.4.1

- Ecrire un script shell qui cherche dans votre arborescence tous les fichiers de nom core, a.out, \*.swp qui n'ont pas été accédés depuis plus de 3 jours, les supprime et vous envoie la liste par mail.  
Ce script sera exécuté tous les 3 jours à 3h du matin, sauf les samedi et dimanche.

# • 12. Gestion des processus

## • 12.4 Exercices correction

- 12.4.1 On place le script nommé `menage` dans un répertoire `script` de son `home`

- `#!/bin/bash`

- `TMP=$HOME/script/liste_du_menage`

- `cd $HOME`

- `find . -type f -name core -atime + 2 -print > $TMP`

- `find . -type f -name core -atime + 2 -print >> $TMP`

- `find . -type f -name core -atime + 2 -print >> $TMP`

- `if [ -s $TMP ]; then`

- `xargs rm <$TMP; mail -s "Fichiers effaces" olivier <$TMP`

- `fi`

- `rm -f $TMP`

# • 12. Gestion des processus

## • 12.4 Exercices correction

- 12.4.1 Ce script est lancé en *cron* par la commande *crontab -e* permettent d'écrire dans la *crontab* l'entrée suivante :
- # mn h jourdu mois mois jourdelasemaine commande
- 0 3 \* \* 1,2,3,4,5 /home/olive/scripts/menage > /dev/null
- 
- La sortie d'erreur n'a pas été redirigée. Les messages d'erreur, s'il y en a, seront envoyés à l'utilisateur par **mail**.