

Système d'exploitation

Processus

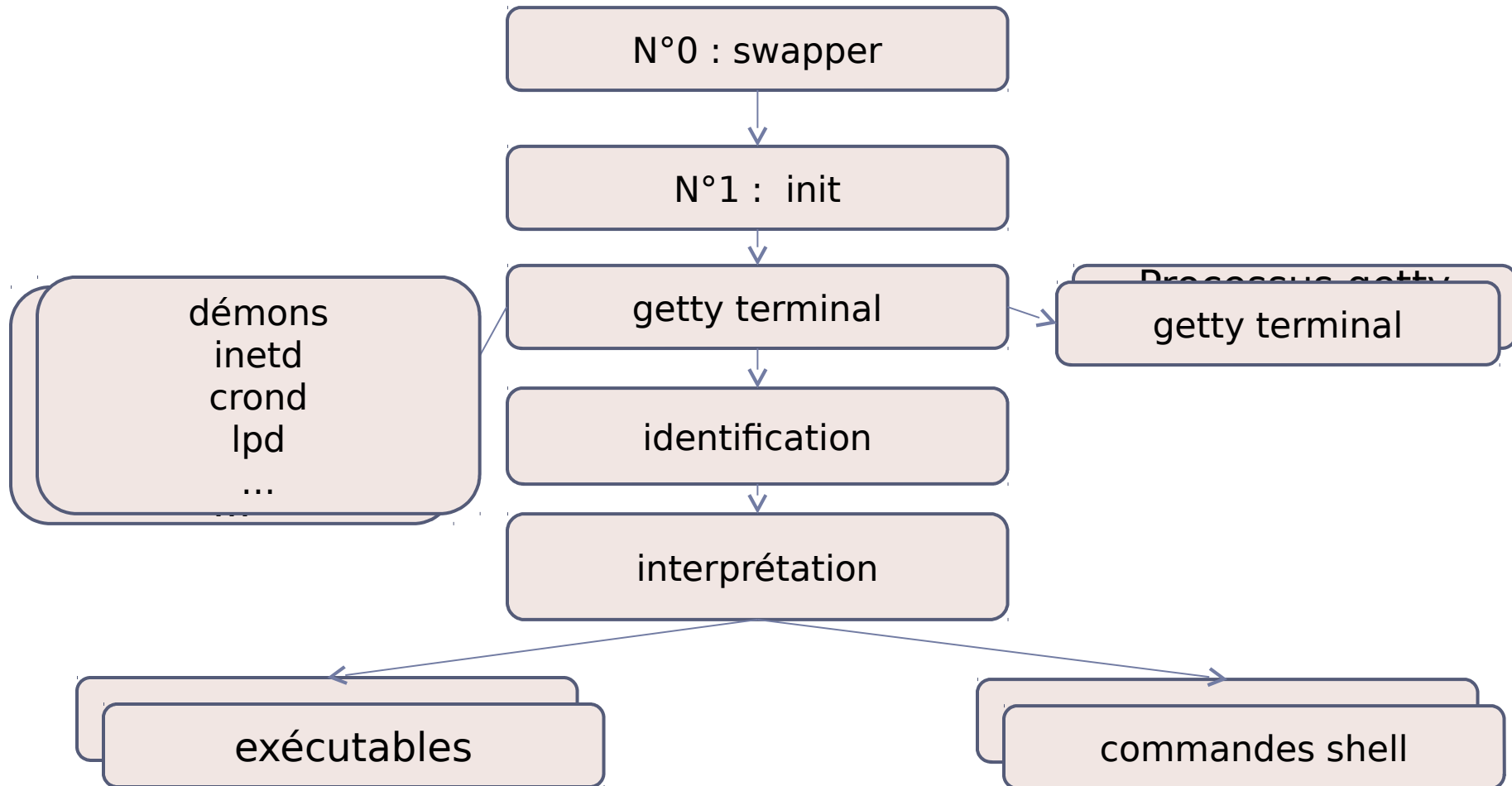
(9)

- Processus - Signaux - Jobs
- Exécutions différées
- Archives

Processus

- Instance de programme en cours d'exécution : **processus**
- Unix multi-tâches => exécution des processus en parallèle simulée
 - => image de l'environnement pour pouvoir suspendre puis reprendre l'exécution : *compteur ordinal, registres, zone de code, données, pile...*
 - **contexte** d'exécution est associé à chaque processus
- Une table des **PCB** (Process Control Block)
 - **PID** (*Process IDentifier*) : *entrée dans la table des processus*
 - Un processus peut créer un ou plusieurs autres processus : **PPID**

Arborescence des processus



Suivi des processus

- Commande **ps**: Process Status 3 séries d'options
 - sans '-' : BSD avec '-' : Unix avec '--' : GNU longues
- **a** : (*autres*) présente également les processus des autres utilisateurs
- **u** : (*utilisateur*) présente le nom de l'utilisateur et l'heure de lancement
- **x** : affiche les processus qui n'ont pas de terminal de contrôle
- **l** : affichage long
- **f** : (*forêt*) affiche les arbres généalogiques des processus
- **e** : (*environnement*) présente l'environnement à la suite de la ligne de commande exécutée
- ...
- Autres commandes : **pstree** **top** **Mate System Monitor**

Arrêt d'un processus

- Un processus réagit à des **signaux** qui lui sont adressés :
 - *Arrêter le processus en avant plan*
 - signal d'arrêt **SIGKILL** émis p.ex. lors d'un **Ctrl+C**
 - *Suspendre le processus (reprise possible)*
 - signal de suspension **SIGSTOP** émis p.ex. lors d'un **Ctrl+Z**
- Commandes pour envoyer des signaux aux processus :
 - **kill** n°processus, **killall** nom_processus
 - **kill -9 825** # 9 = signal KILL
 - **kill -l** : liste des signaux supportés
- Vérifier les fichiers ouverts
 - **lsof** : liste des fichiers ouverts
 - **fuser** : liste des processus utilisant un fichier

Capture des signaux

- **trap** : intercepte les signaux pour lancer une commande (*interpréteur courant*)
 - **trap 'commande' n** événement n intercepté et *commande* exécutée
 - **trap n** le signal n est rétabli
 - **trap -l** liste les signaux (*man 7 signal*)
 - **trap " n** le signal n est ignoré

programme que l'on ne peut pas interrompre par Ctrl+C

trap 'echo "Arrêt par Ctrl-C interdit"' 2

for ((i=0;i<10;i++)) ; do echo -n "." ; sleep 1 ; done

trap 2 # NPO : rétablir l'interruption (*si source xxx.sh*)

- **nohup** exécute une commande qui ne sera pas interrompue par la déconnexion

Les processus en arrière-plan

- Un processus s'exécute soit :
 - en **avant-plan** (mode normal) : processus
 - en **arrière-plan** (jobs/démons) : terminer la ligne de commandes par un **&**
- Gestion via des commandes built-in :
 - **jobs** : lister les jobs avec leur N° et leur état
 - gestion spécifique : N° de job avec numérotation par interpréteur
 - **fg** *%N°job* : ramener un job en avant-plan
 - **bg** *%N°job* : relancer un job suspendu (*ctrl+Z*), en arrière-plan
 - **kill** *%N°job* : tuer un job

Exécutions différées

- **at** : Exécute des commandes à une heure donnée
- **batch** : Exécute des commandes "dès que possible"
i.e. lorsque la charge du processeur le permet
- Les commandes **at** et **batch** lisent, depuis l'entrée standard ou un fichier, des commandes qui seront exécutées ultérieurement :
 - par un démon **atd** ultérieurement,
 - en utilisant un interpréteur **/bin/sh**
 - **ne sont pas associées à un terminal**
(Si la commande génère un affichage celui-ci doit être redirigé vers un fichier pour pouvoir être exploité)

La commande **at**

- **at** : Lance à une heure donnée les commandes spécifiées
 - la commande **at heure** attend la saisie des commandes :
 - invite **at>**
 - ... # saisie de la suite de commandes
 - terminer par un **Ctrl+D** en début de ligne
 - Spécification de l'heure d'exécution :
 - HHMM ou HH:MM ou midnight , noon , ou teatime ...
 - nom-du-mois (en anglais) jour et éventuellement année
 - date du type MMJJAA ou MM/JJ/AA ou JJ.MM.AA
 - now + *nb* minutes, hours, days, ou weeks
 - suffixes today , tomorrow ...
- **at -f script heure** : lit la(les) commande(s) depuis le fichier spécifié

Commandes en attente

- Les commandes en attente sont associées à un fichier dans le répertoire `/var/spool/cron/atjobs`
 - le nom du fichier est un nombre attribué dynamiquement
 - il appartient à l'utilisateur qui a lancé la commande (*et au groupe daemon*)
- **atq** ou **at -l**
 - affiche la liste des commandes en attente d'exécution de l'utilisateur
 - *sauf si l'appelant est root : affiche la liste pour tous les utilisateurs*
- **atrm** ou **at -d**
 - efface les travaux en attente

Autorisations

- Fichiers d'autorisation (*un nom par ligne + mot réservé **all** pour tous*)
 - **/etc/at.allow** : utilisateurs autorisés
 - **/etc/at.deny** : utilisateurs non autorisés

- Règle d'autorisation :
 - Si le fichier **/etc/at.allow existe** : seuls les utilisateurs dont les noms sont listés peuvent utiliser at
 - Si **/etc/at.allow n'existe pas** et que **/etc/at.deny existe** : les utilisateurs non listés dans ce fichier peuvent utiliser at
 - Si aucun des deux fichiers n'existe : seul le Super Utilisateur a le droit d'utiliser at

- Sous Debian configuration par défaut : un unique fichier **/etc/at.deny**

Processus périodiques

- Le démon **cron** permet d'exécuter des commandes périodiquement : lancer la(les) commande(s) à telle(s) heure(s), tel(s) jour(s)'
- Les commandes se trouvent dans un fichier **crontab** et sont exécutées
 - avec les droits du propriétaire de la table
 - depuis son répertoire de connexion
- Chaque utilisateur dispose de sa propre table unique à son nom dans **/var/spool/cron/crontabs**
- Des dossiers sont aussi prévus pour root pour y placer des scripts :
 - **/etc/cron.daily**, **/etc/cron.hourly** , **/etc/cron.monthly** , **/etc/cron.weekly**

Editeur de table

- Edition de la table : **crontab -e**
- Format d'une ligne de la crontable :
 - **5 champs** fixant la date et l'heure, suivis des commandes à exécuter:
 - **minute** (valeur entre 0 et 59),
 - **heure** (0-23),
 - **jour du mois** (0-31),
 - **mois** (0-12) (ou nom anglais (*3 premières lettres*)),
 - **jour de semaine** (0-7 ou nom) (*où 0 et 7 sont Dimanche*)
- Les lignes blanches, espaces et tabulations en tête de lignes sont ignorées
- Les lignes dont le 1^{er} caractère non-blanc est un # sont des commentaires
 - **les commentaires ne peuvent pas se trouver sur la même ligne** qu'une commande cron, ils seraient considérés comme des arguments

Syntaxe crontable

- * : correspond à l'intervalle premier-dernier
 - 0 14 1 * * => à 14h00 le 1^{er} de chaque mois
 - * * * * * date>>quand.txt =W> toutes les minutes
- Les intervalles sont permis :
 - 0-5 => 0 1 2 3 4 5
 - des valeurs de "pas" peuvent être associées aux intervalles
0-30/10 => 0 10 20 30
- Les listes sont permises :
 - 1,2,5,9
 - 0-4, 8-12
- Il est aussi possible de spécifier la périodicité via des mots clés :
 - *daily, weekly, monthly...*

Démon cron

- **cron** examine ses données une fois par minute
 - Les commandes sont exécutées par *cron* lorsque
 - 1) les champs minute, heure, et mois correspondent à la date
 - 2) **et** quand **au moins l'un des deux autres champs**
(jour du mois, jour de la semaine) correspond aussi
 - Exemple : 30 4 **1,15** * **5** *commande*
 - exécute la commande à 4:30 du matin les 1^{er} **et** 15 *(1,15)*, de chaque mois(*), **et** chaque Vendredi *(5) même si l'on n'est pas le 1 ou le 15*
- **cron n'est pas associé à un terminal**
- Autorisations : **/etc/cron.allow** **/etc/cron.deny**
- **crontab fichier** : *remplace le contenu de la table par celui du fichier*
- **crontab -l** : *liste les commandes de la table*
- **crontab -r** : *supprime la table*

Archivage

- Une archive rassemble des fichiers et répertoires en un seul fichier
- Commande d'archivage : **tar** (**t**ape **a**rchiver) sans compression
- Options
 - **v** verbose (*fréquemment spécifiée*)
 - **f** *nom-fichier* spécifier le nom du fichier d'archive
 - **c** créer un fichier archive
 - **x** extraire des fichiers de l'archive
 - **t** lister les fichiers de l'archive
 - **z** compresser au format
 - **j** compresser au format bz2
- La restauration des fichiers **recrée l'arborescence nécessaire** à partir du **point de restauration**
 - regrouper les fichiers dans un répertoire puis l'archiver

Usages

- Combinaisons utilisées :
 - Sauvegarder : `tar -cvf sauve.tar repertoires fichiers ...`
 - Lister : `tar -tf sauve.tar`
 - Restaurer : `tar -xvf sauve.tar`
- Ne restaurer que certains fichiers :
 - spécifier le chemin tel que dans l'archive (*supporte leglobbing*)
`tar -xvf sauve.tar repertoire/images/*.jpg`