Administration Réseaux et Systèmes

Plan

1. Gestion des comptes systèmes et des permissions
2. Installation Système et gestion des disques
3. Script Shell
4. Gestion des adresses, routage et parfeu
5. Planification et automatisation des taches
6.

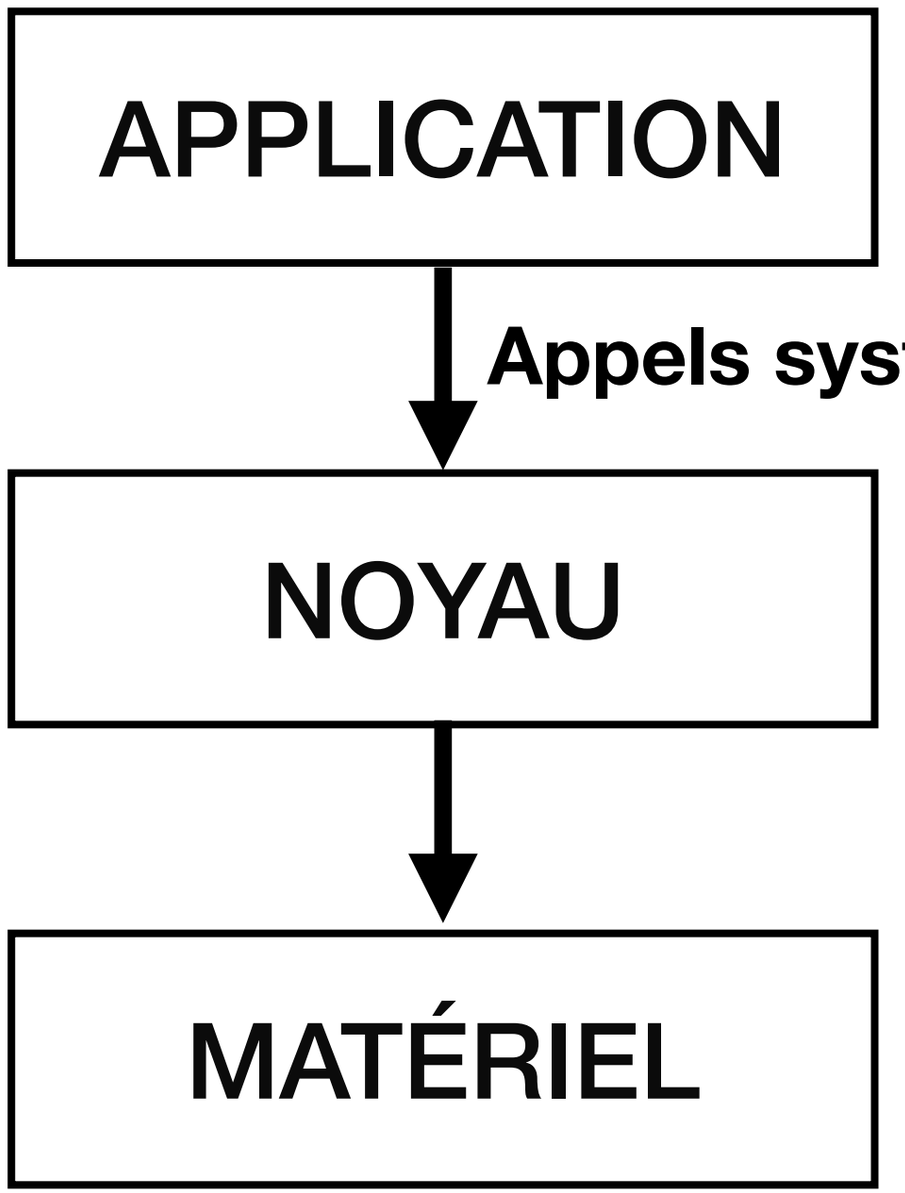
I. Gestion Comptes Systèmes

Sommaire

Sytème d’exploitation
Histoire

- Séparation machine / données
    - 1645 **Pascaline**;
    - 1943 **Colossus vs Enigma**;
- Séparation machine / programme
    - 1944 **MarkI**;
- Séparation programme utilisateur / système d’exploitation
    - 1972 **C / UNIX**;
    - 1979 **Intel 8086**.

Sytème d’exploitation
Schéma



cp, Firefox, gcc, ls,

apache, ftp, etc

Linux, Kernel32DLL, etc

AMD64, ARM, INTEL, etc

Sytème d’exploitation
Définition

Système d’exploitation = Applications + Noyau

Operating System = Userland + Kernel

Exemple : Distribution GNU/Linux (Debian, CentOS, etc.)

- **GNU** Applications;
- **Linux** Noyau

Sytème d’exploitation
Évolution des OS

1. Interrupteurs / rouages : pas de SE ;
2. SE sur cartes perforées / bandes magnétiques :
   - ➡ On charge le SE puis le programme ;
   - ➡ Puis Job Control Language (JCL) permet d’enchaîner les jobs.
3. Mono-processus, mono-utilisateur
   - ➡ La machine ne fait qu’une chose à la fois ;
   - ➡ Elle ne fait pas de différence entre les utilisateurs.
4. Apparition des terminaux (écran / clavier)
   - ➡ Multi-job ;
   - ➡ Puis multi-utilisateurs / multi-processus : UNIX !

Sytème d’exploitation
Évolution des OS

Système multi-processus

- ➡ Plusieurs processus (tâches, programmes...) s’exécutent « simultanément » sur la machine.

Système multi-utilisateurs

- ➡ Chaque utilisateur est identifié de façon unique (notion de login, d’authentification) ;
- ➡ Il dispose de la possibilité de protéger ses données (notion de droits sur les fichiers et les processus).

Sytème d’exploitation
Utilisation d’un OS

Interface texte « shell » :

- ➡ Commandes de base ;
- ➡ Plus simple et rapide à développer que des outils graphiques ;
- ➡ Moins d’erreurs (bugs) ;
- ➡ Plus de stabilité ;
- ➡ Plus automatisable (donc plus facile à tester).

Commande UNIX (de base) :

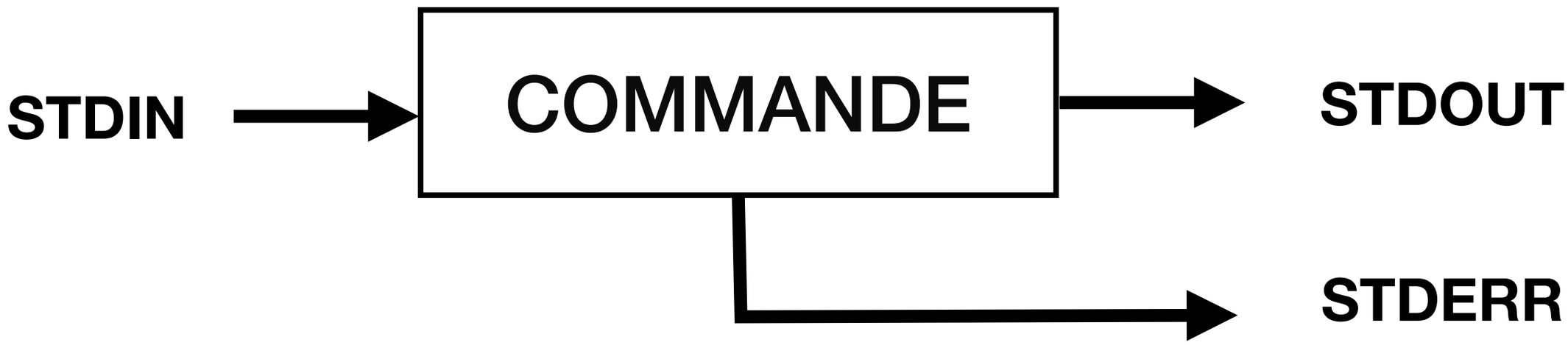
- ➡ Exécutée par un shell (souvent bash(1))

  Syntaxe :

  % **commande** [-options] [arguments]

- ➡ Peut utiliser l’entrée standard (**stdin**), la sortie standard (**stdout**) et la sortie d’erreur (**stderr**) du shell pour interagir avec l’utilisateur.

Sytème d’exploitation
Utilisation OS



Sytème d'exploitation
Utilisation OS

% **commande** [-options] [arguments]

Commande peut être :

- ➡ Interne au shell ;
- ➡ Programme externe;
- ➡ Alias;

Options :

- ➡ Commencent par un tiret;
- ➡ Modifient le comportement de la commande;
- ➡ Peuvent généralement être groupées

```
% ls -r -t /var/log
% ls -rt /var/log
```

Arguments :

- ➡ Ce sur quoi agit la commande;
- ➡ Exemple

```
% ls -rt  /var/log
```

Sytème d’exploitation
Système de fichiers

```
/                                   # /dev/sda2
|--  bin/
|--  boot/                          # /dev/sda1
|        |-- initrd.img
|        |-- ulmage
|        |-- ulnitrd
|--  dev/
|— apt/
|--  etc/
|--  home/                  #   marvin:/nfs/home
|        |-- john/
|        |-- romain/
|--  lib/
|--  mnt/
|        |— cdrom/
|--  var/
```

Répertoire	Contenu
bin	Binaires (exécutables) des commandes essentielles ((ex: `cd`, `cat`, `ls`…))
boot	Fichiers statiques pour le programme d'amorçage
dev	Fichiers des pilotes de périphériques
etc	Configuration système propre à la machine
home	Répertoires personnels des utilisateurs
lib	Bibliothèques partagées et modules noyaux essentiels
media	Points de montage pour les supports amovibles
mnt	Point de montage pour les montages temporaires
proc	Répertoire virtuel pour les informations système
root	Répertoire personnel de l'utilisateur root
run	Données variables d'exécution
sbin	Exécutables système essentiels
sys	Répertoire virtuel pour les informations système
tmp	Fichiers temporaires
usr	Programmes, bibliothèques des utilisateurs
var	Données variables (caches, sites web, base de donnée,etc.)
srv	Données pour les services fournis par le système
opt	Répertoire pour d'autres logiciels

Sytème d’exploitation
Système de fichiers

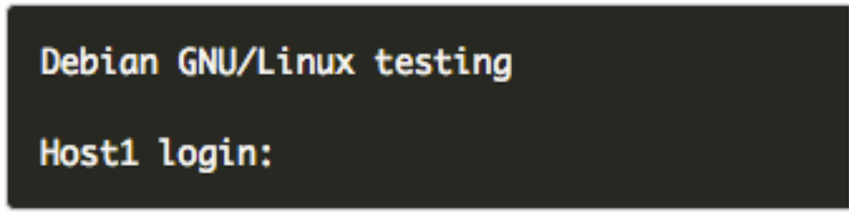
Structure arborescente.

- ➡ **/** est la racine du système de fichiers ;
  Chaque nœud est un fichier (un répertoire est un fichier spécial) ;
- ➡ Un répertoire est un fichier qui peut contenir d’autres fichiers ;
- ➡ Un chemin est un parcours dans l’arbre pour identifier un fichier.

Un fichier ordinaire (regular file en anglais) est le nom donné à un fichier qui n’est pas spécial.

Ouverture et fermeture d’une session

- Deux Types d’installation possible : mode graphique ou mode texte



mode texte

mode graphique

- Pour se connecter, il faut d’abord créer des comptes utilisateurs caractérisés par un login et un mot de passe

Ouverture et fermeture d'une session

- **Mode Texte**

Ouverture de session

```
Debian GNU/Linux testing

Host1 login:
```

```
Debian GNU/Linux testing

Host1 login: lucas
Password:
```

Une fois connecté, le prompt apparait

```
lucas@host1:~$
```

~ représente le répertoire d'utilisateur /home/lucas

Fermeture de session

- Pour se déconnecter ou fermer la session. Il vous suffit au prompt de taper **logout**. Vous vous retrouvez alors avec le prompt de login, un autre utilisateur pourra alors utiliser la machine.

```
lucas@host1:~$ logout
```

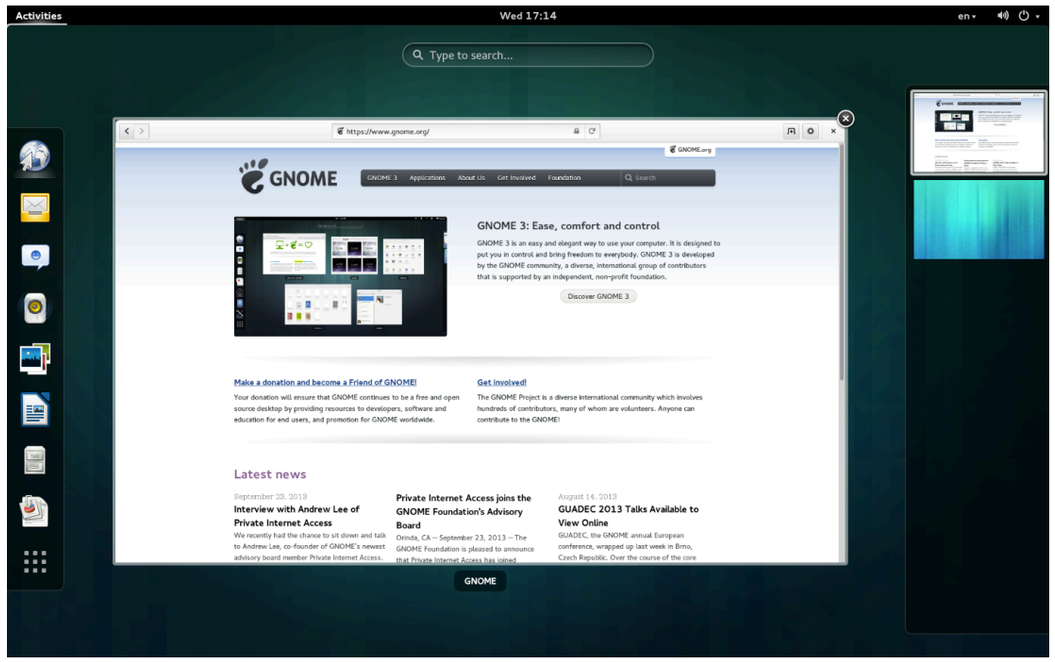
Ouverture et fermeture d’une session

- Mode graphique

Écran de connexion

- Pour se déconnecter, il faut cliquer sur le bouton d’arrêt ou le nom de l’utilisateur du coin supérieur droit

Gnome : Après connexion



Sytème d’exploitation
Montage système de fichiers

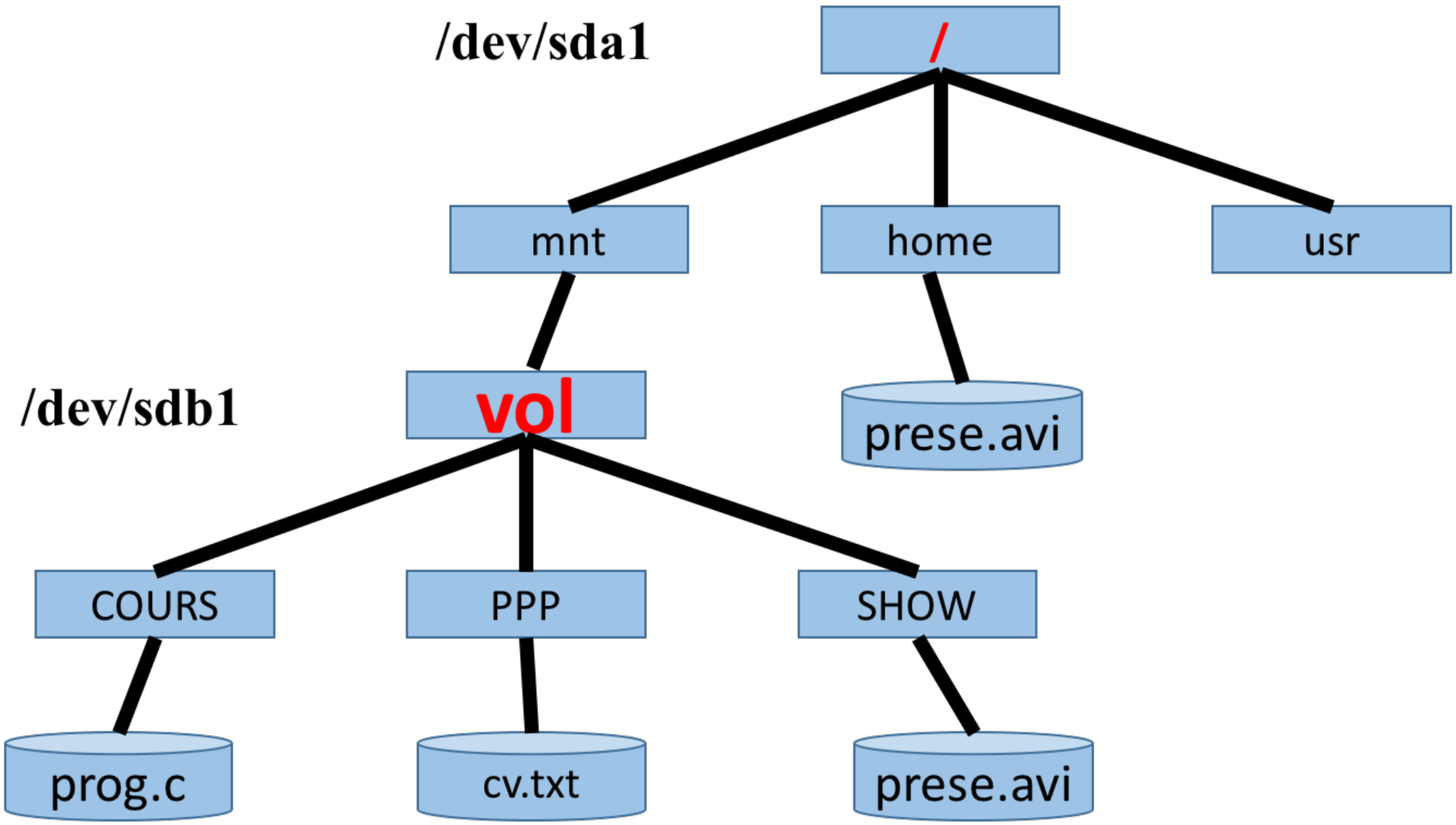
Sous UNIX, les unités physiques sont masquées.

Pour accéder à des données d’un périphérique (clé USB, partition de disque dur), le système de fichier doit préalablement être monté dans un répertoire.

Les fichiers du volume sont dès lors accessibles via le dossier de montage dans le système de fichier (on parle de **point de montage**).

Les systèmes d’exploitation modernent montent généralement automatiquement les périphériques amovibles lorsqu’ils sont connectés.

Sytème d'exploitation
Montage système de fichiers



Sytème d’exploitation
Type de fichiers

La commande **ls**(1) avec l’option **-l** affiche les caractéristiques des fichiers du répertoire courant :

```
% ls -l
-rwxr-xr-x 1 toto etud 1168 juin 8 19:43 rapport.txt
```

Morceau	Description
-rwxr-xr-x	Type et permissions
1	Nombre de liens physiques
toto	Propriétaire
etud	Groupe
1168	Taille (en octets)
juin 8 19:43	Date de dernière modification
rapport.txt	Nom du fichier

Sytème d’exploitation
Type de fichiers

Le premier caractère de la ligne indique le type de fichier

% ls -l

-rwxr-xr-x 1 toto etud 1168 juin 8 19:43 rapport.txt

Type de fichiers	
-	Fichier ordinaire
d	Répertoire
l	Lien symbolique
p	Tube nommé (fifo)
s	Socket
c	Périphérique en mode caractère
b	Périphérique en mode bloc

Sytème d’exploitation
Quelques répertoires particuliers

Répertoire personnel - Home directory- $HOME

- ➡ Propre à chaque utilisateur (défini à la création d compte)
- ➡ Stocke les fichiers de l’utilisateur

Répertoire courant - Current work directory

- ➡ Répertoire où le Shell fera ses opérations sur le fichier
- ➡ Vaut initialement $HOME pour un shell de connexion
- ➡ Peut être changer via la commande **cd** (**c**hange **d**irectory)
- ➡ Pour afficher le répertoire courant on utilise la commande **pwd** (**p**rint **w**ork **d**irectory)

Sytème d'exploitation

Quelques répertoires particuliers

. Répertoire courant

.. Répertoire parent

~ Le shell substitue un tilde par le chemin du répertoire $HOME

~toto le shell le substitue par le chemin du répertoire $HOME de **toto**

/tmp Accessible à tous pour placer temporairement des fichiers (vidé au démarrage de la machine)

Sytème d’exploitation
Chemins d’un fichier

Le chemin d’un fichier indique sa localisation dans l’arborescence

Deux types de chemin :

Absolu

Liste des répertoires traversés pour aller de la racine (/) au fichier.

Les différents répertoires sont séparés par le séparateur / .
Exemples : */etc/passwd* , */home/etud/1a/toto/tp1*

Relatif

Liste des répertoires traversés pour aller du répertoire courant au fichier.

Les différents répertoires sont séparés par le séparateur / .

Exemples : d1/fic1 , ../d2/fic

Sytème d’exploitation

Commandes de manipulation des fichiers

% *cd* [chemin répertoire]

Change le répertoire courant (change directory) vers le répertoire [répertoire].

% *pwd*

Affiche le chemin absolu du répertoire de travail (print working directory).

% *ls* [options] [fichier...]

Affiche le contenu d’un répertoire (list).

% *cp* [options] source... répertoire_destination

Copie des fichiers

% *mkdir* [options] répertoire...

Créer des répertoires (make directory).

Sytème d'exploitation
Droits des fichiers

3 groupes de 3 droits pour le propriétaire (**u**ser), le groupe (**g**roup) et les autres (**o**ther)

% ls -l

-rwxr-xr-x 1 toto etud 1168 juin 8 19:43 upside-down-ternet

uuugggooo

Type de fichiers

uuu permission pour le propriétaire

ggg permission pour le groupe

ooo permission pour les autres

r = read. ; w = write ; x = execute

Une lettre indique que la permission est donnée, un tiret que la permission n'est pas accordée

Droits sur les fichiers
Notion d'utilisateur

Sous unix, un utilisateur est décrit par une ligne dans le fichier **/etc/passwd**.

Exemple

```
paul:*:1000:521:Paul Bismuth:/home/paul:/bin/sh
```

Morceau	Description
paul	Identifiant (login)
*	Mot de passe haché (dans /etc/shadow)
1000	User ID (uid)
521	Group ID (gid) du groupe primaire (dans /etc/group)
Paul Bismuth	Champs GECOS (Nom réel, bureau, téléphone, …)
/home/paul	Chemin du répertoire ($HOME)
/bin/sh	Shell de connexion

Droits sur les fichiers
Création d'utilisateur

% **useradd** [-options] LOGIN : binaire basique de création d'utilisateurs en mode console

% **adduser** [-options] LOGIN : script interactif de création d'utilisateurs

% **usermod** [-options] LOGIN : modifie un compte utilisateur

% **userdel** [-options] LOGIN : suppression basique d'un compte utilisateur

% **deluser** [-options] LOGIN : suppression d'un compte utilisateur avec option de sauvegarde

% **groupadd** [-options] groupe : création d'un nouveau groupe

% **groupdel** [-options] groupe : suppression d'un groupe

http://www.man-linux-magique.net/

Droits sur les fichiers
Notion de groupe

Sous unix, un groupe est décrit par une ligne dans le fichier **/etc/group.**

Chaque utilisateur appartient au moins à un groupe

Exemple

```
etud:*:521:paul
```

Morceau	Description
etud	Nom du groupe (login)
*	Mot de passe haché (peu fréquent)
521	Group ID (gid)
paul	utilisateurs membres du groupe

Un utilisateur a un seul groupe primaire mais peut appartenir à plusieurs groupes.
À sa création, un fichier a pour propriétaire son créateur et pour groupe celui de son créateur.

Droits sur les fichiers
Propriétaire et groupe

Un utilisateur peut changer le groupe de son fichier avec **chgrp**(1) ou **chown**(1) .

% chgrp [options] groupe fichier...
Modifie le groupe d'un fichier (change file group).

% chown [options] propriétaire[:groupe] fichier...
% chown [options] :groupe fichier...
Modifie le propriétaire et le groupe d'un fichier (change file owner).

Un utilisateur standard ne peut pas changer le propriétaire d'un fichier.
L'administrateur peut changer le propriétaire et le groupe de tous les fichiers.

Droits sur les fichiers
Propriétaire et groupe

La commande ls(1) avec l'option -l permet de visualiser les droits actifs :

% ls -l

-rwxr-xr-x 1 toto etud 1168 juin 8 19:43 upside-down-ternet

Le propriétaire d'un fichier peut fixer des droits sur ce fichier, via la commande chmod(1) , en :

- ➡ lecture (**r**ead) = 4;
- ➡ écriture (**w**rite) = 2;
- ➡ exécution (**e**xecute) = 1.

% chmod [options] mode fichier

Modifie les permissions d'un fichier (change file mode).

Droits sur les fichiers
Propriétaire et groupe

L'utilisateur peut définir les permissions pour les 3 catégories d'utilisateurs suivantes :

- ➡ lui-même (user) ;
- ➡ ceux de son groupe (group) ;
- ➡ tous les autres (other).

L'administrateur (root) a tous les droits et peut donc définir les permissions qu'il désire.

Droits sur les fichiers
Définir les options en symbolique

% chmod g+x fichier

Ajoute le droit d'exécution à ceux déjà présents pour le groupe de fichier.

% chmod a=rw fichier

Donne les droits de lecture et d'écriture pour tout le monde (all), et enlève le droit d'exécution de fichier.

% chmod o-rwx fichier

Enlève tous les droits pour les autres (ni propriétaire, ni dans le groupe de fichier).

Droits sur les fichiers
Définir les options en octal

% chmod 755 fichier

Tous les droits pour le propriétaire, mais pas de droit d'écriture pour les autres (y compris ceux du groupe de fichier).

% chmod 600 fichier

Permissions restreintes. Donnez un cas réel.

% chmod 705 fichier

Permissions restreintes. Donnez un cas réel.

Droits sur les fichiers
Cas des répertoires

La signification des permissions sur un répertoire est sensiblement différente que sur un fichier ordinaire :

- ➡ **lecture** Permet d'obtenir la liste de ses fichiers ;
- ➡ **écriture** Permet de créer / supprimer des fichiers ;
- ➡ **exécution** Permet d'entrer dans le répertoire.

Que permet ce répertoire ?

drwx-wx--- 1 prof etud 11 mai 16 14:23 public/

Pattern matching (jockers)

Il est possible d'utiliser des jockers dans un nom de fichier pour désigner une liste de fichiers. Le shell les substituera avant d'exécuter la commande.

Jocker	Description
*	Suite de caractères quelconques (0 caractère ou plus) ;
?	Un caractère unique ;
[…]	Un caractère unique parmi l'ensemble ;
[!…]	Un caractère unique hors de l'ensemble.

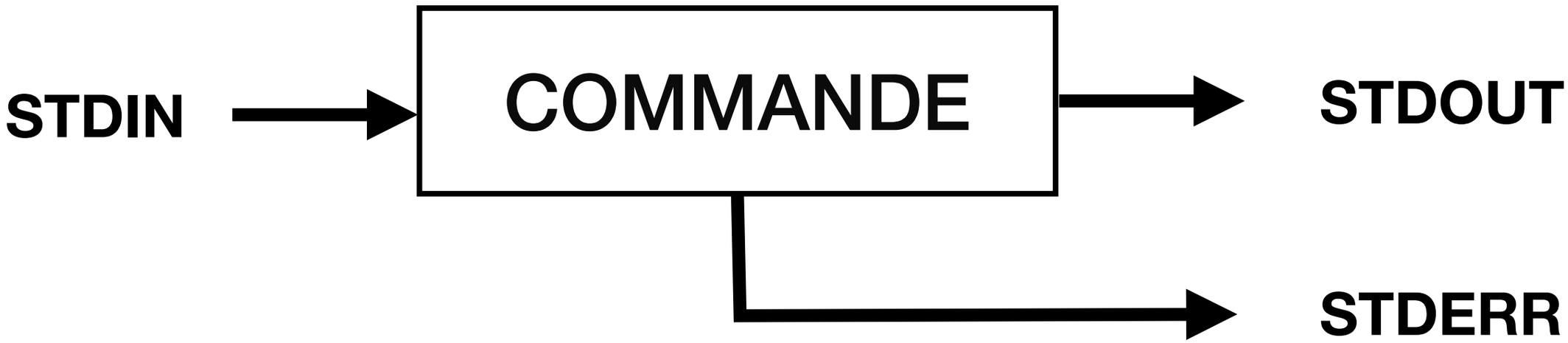
Avant d'exécuter la commande, le shell cherche les fichiers qui correspondent au motif et substitue le résultat de la recherche à l'expression qui a permis de les trouver.

Pattern matching (jockers)
Remarques

- ➡ Dans un ensemble, il est possible de préciser un intervalle à l'aide du tiret([a-zA-Z] pour les lettres). Il est également possible d'utiliser une classe de caractères [[:alpha:]] ;
- ➡ Pour protéger un jocker (empêcher le shell de le substituer), il est nécessaire de le faire précéder d'un **** ou de le placer à l'intérieur de **'** ou de **"** . Exemples : rm *.c ou rm '*.c' ou enfin rm "*.c " permettent de supprimer un fichier nommé « *.c »;
- ➡ Par défaut, le shell bash(1) ne lève pas d'erreur lorsqu'une substitution échoue mais passe le motif comme argument de la commande !

Redirections

➡ Un processus UNIX utilise par défaut trois entrées / sorties standards héritées de son père :



Entrée standard associée au descripteur 0 (flux stdin).

Par défaut dans un shell interactif, le clavier ;

Sortie standard associée au descripteur 1 (flux stdout).

Par défaut dans un shell interactif, l'écran ;

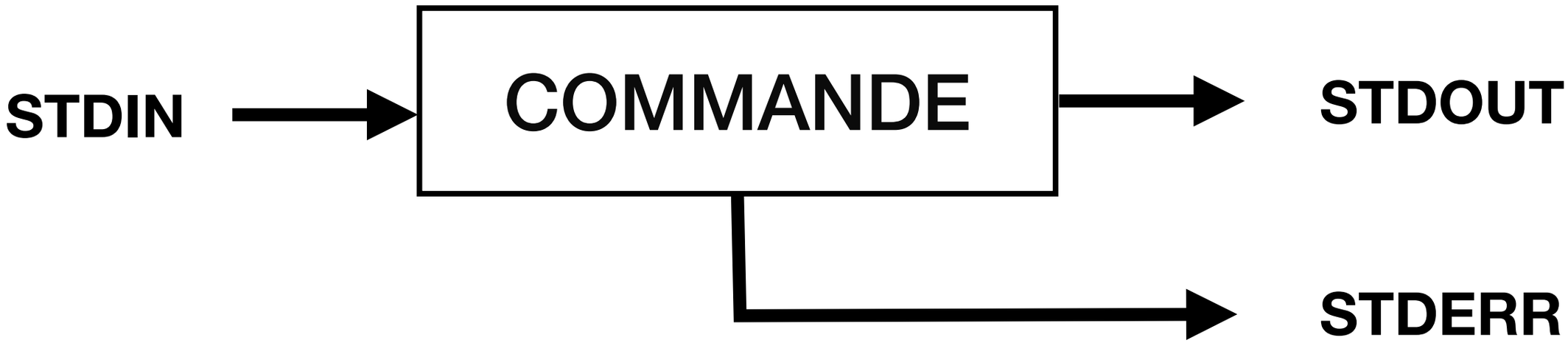
Sortie d'erreur associée au descripteur 2 (flux stderr).

Par défaut dans un shell interactif, l'écran ;

Redirections

Opérateurs de redirection

➡ Un processus UNIX utilise par défaut trois entrées / sorties standards héritées de son père :



Entrée standard associée au descripteur 0 (flux stdin).
Par défaut dans un shell interactif, le clavier ;

Sortie standard associée au descripteur 1 (flux stdout).
Par défaut dans un shell interactif, l'écran ;

Sortie d'erreur associée au descripteur 2 (flux stderr).
Par défaut dans un shell interactif, l'écran ;

Redirections

Opérateurs de redirection

Les opérateurs suivants permettent de rediriger ces entrées / sorties:

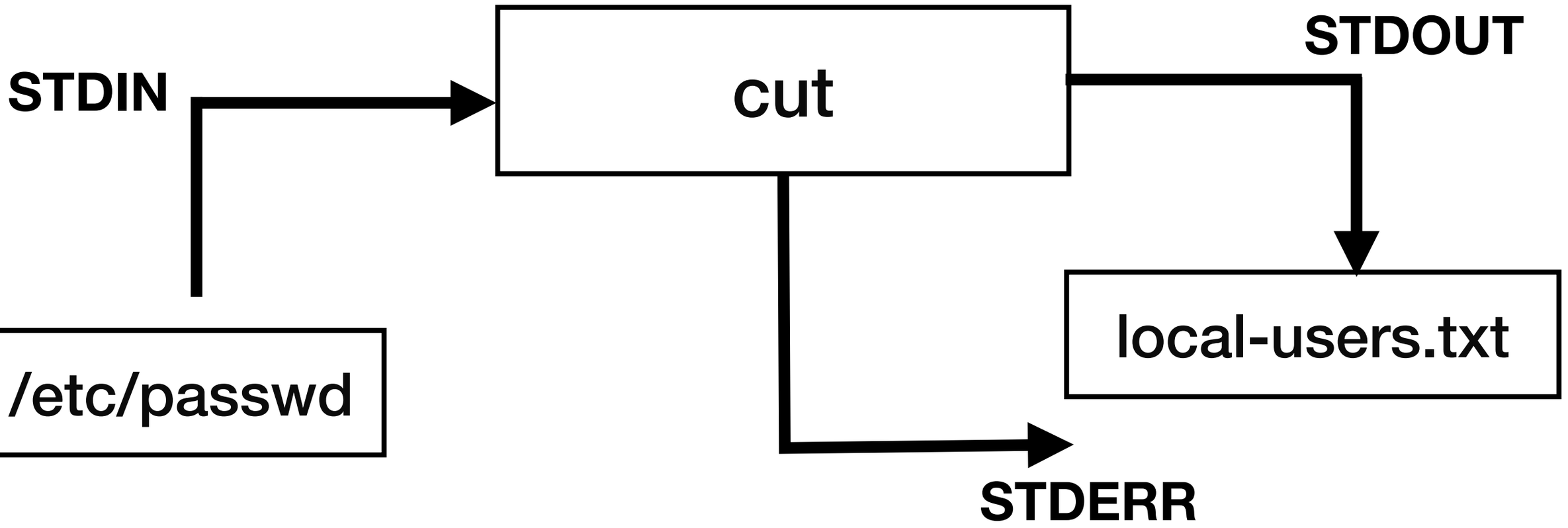
Opérateurs	Description
> fichier	Redirige la sortie standard vers fichier (avec écrasement du fichier);
>> fichier	Redirige la sortie standard vers fichier (sans écrasement du fichier mais ajout en fin) ;
< fichier	Redirige fichier vers l'entrée standard ;
2> fichier	Redirige la sortie d'erreur vers fichier ;
>& fichier	Redirige la sortie standard et la sortie d'erreur vers fichier ;
>&2	Redirige la sortie standard vers la sortie d'erreur.

Les redirections sont indiquées après la commande à exécuter.

Redirections

Exemples

% cut -d: -f 1 < /etc/passwd > local-users.txt



LES TUBES (PIPES)

➡ Un tube est une zone mémoire gérée comme une file (FIFO) permettant de *connecter la sortie standard d'un processus vers l'entrée standard d'un autre processus*. Ils permettent ainsi d'enchaîner plusieurs commandes en se passant de fichiers intermédiaires.

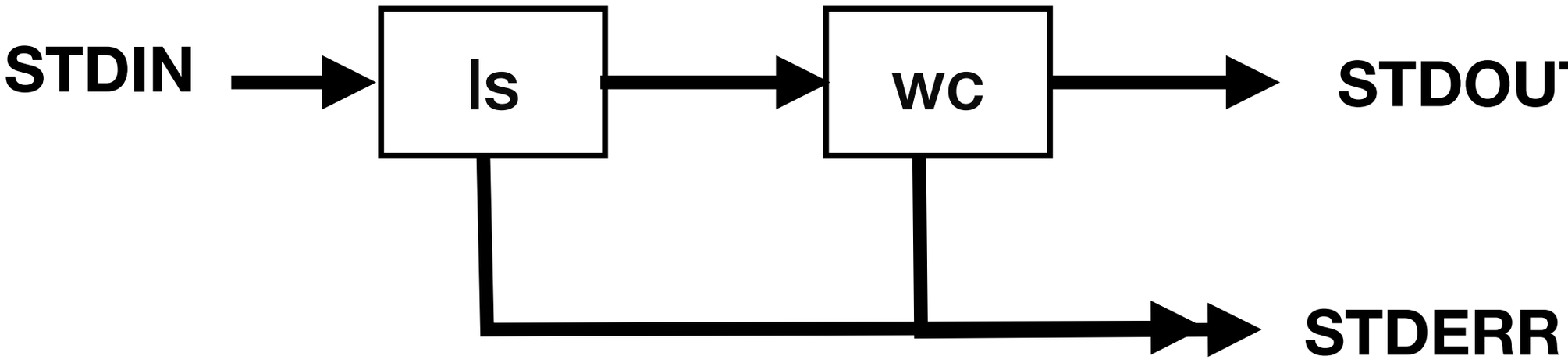
La syntaxe d'un tel enchaînement est la suivante :

% command 1 | commande 2 | ... | commande n

LES TUBES (PIPES)

Exemples

% ls -l | wc -l

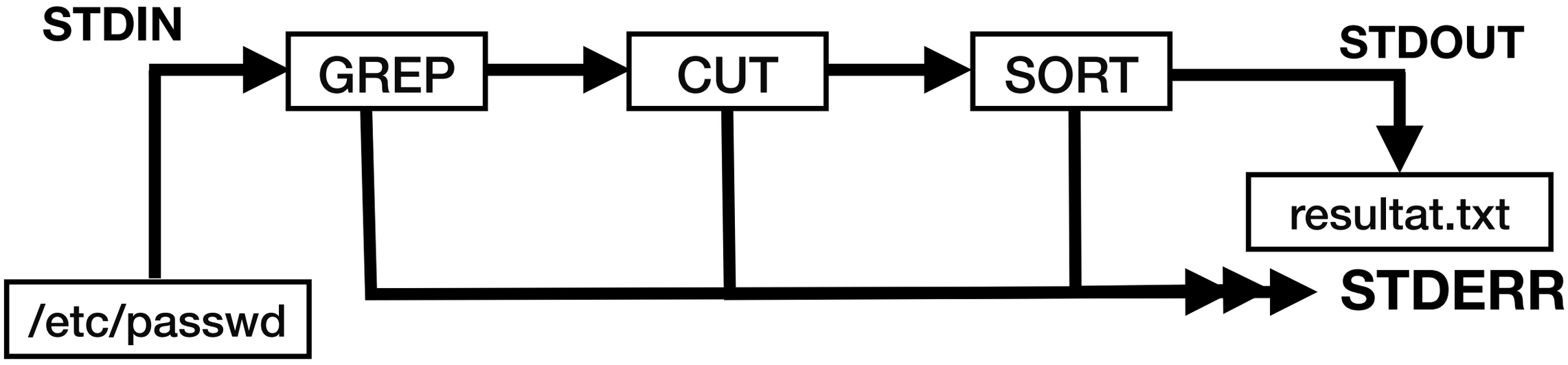


ls(1) liste les fichiers (un fichier par ligne), **wc(1)** compte les lignes. La commande composée affiche donc le nombre de fichiers dans le répertoire courant.

LES TUBES (PIPES)

Exemples

% grep '/bin/bash' < /etc/passwd | cut -d : -f1 | sort > resultat.txt



grep(1) n'affiche que les lignes qui contiennent « /bin/bash », **cut(1)** extrait les logins, **sort(1)** trie par ordre alphabétique. La commande composée écrit donc la liste des utilisateurs de bash(1) par ordre alphabétique.

FIN CHAP 1