

UNIX - Parties 5 & 6

Variables & Redirections



Objectifs

- Apprendre à utiliser LINUX
- Comprendre l'environnement de travail (shell)
- Comprendre la notion de variable d'environnement
(portée d'une variable)
- Comprendre et utiliser les redirections
Présentation de quelques commandes courantes



Plan

- Environnement de travail
- Variables
- Portée d'une variable
- Redirections
- En pratique
 - Commandes en séquence
 - Liens symboliques
 - Commande wc

Qu'est-ce qu'un shell ?

Logiciel à part entière : interpréteur de commande

Terminal = shell, mais possibilité de lancer plusieurs shells simultanément

L'environnement de travail (ou shell)

Cadre fonctionnel d'une session donné par le shell lancé au login

Variables, fichiers de démarrage, etc...

Facilités diverses (rappel des dernières commandes, création d'alias, etc...)

Adaptable par chacun

Les différentes familles de shell

- Bourne Shell (sh),
- CShell (csh, tcsh),
- Korn Shell (ksh) ...

L'utilisateur peut choisir son shell
et en changer temporairement ou définitivement

Caractéristiques communes aux différents shells

- Définition de variables
- Redirection des entrées et des sorties
- Communication par des « pipe»
- Lancement de programmes en arrière plan
- Programmation (scripts shell)

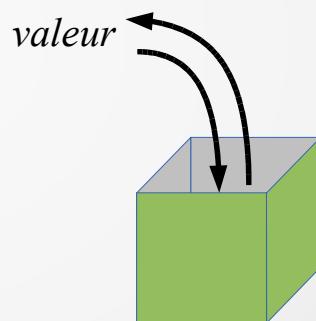
NB : syntaxe, commandes, etc... différent selon l'environnement choisi
(en TP : Cshell, tcsh ==> `#!/bin/csh`)

Les variables en Cshell - Généralités

Variables locales et variables d'environnement

Affectation :

```
set <nom_variable>=valeur
ou
setenv <nom_variable> valeur
```



Référence :

`$nom_variable`

Manipulation des variables locales

- **set [[<variable>] | [<variable>=valeur]]**

liste les variables et leur contenu ou crée une variable vide ou affecte une valeur à une variable

- **unset <variable>**

supprime une variable

- **echo \$<variable>**

affiche le contenu d'une variable

Les variables locales en Cshell – Premiers exemples

```
dupond@@k-etu-052% set nom=Dupond
dupond@@k-etu-052% echo bonjour $nom
bonjour Dupond
dupond@@k-etu-052%
```

```
dupond@@k-etu-052% set repertoire=/amd/ladyburn
dupond@@k-etu-052% cd $repertoire
dupond@@k-etu-052% pwd
/amd/ladyburn
dupond@@k-etu-052%
```

Quelques variables locales prédefinies

- noclobber
 - empêche redirection vers fichier existant
- home
 - chemin du répertoire d'accueil de l'utilisateur
- prompt
 - définit l'invite de la ligne de commande
- cwd
 - contient le nom du répertoire courant
- \$<
 - permet de saisir une information sur l'entrée standard (le clavier)

Exemple : set saisie=\$<

Expressions arithmétiques

$@(\text{variable}) = (\text{expression})$

Exemple : $@c = $a + b

Manipulation des variables d'environnement (variables globales)

■ env, setenv, printenv

liste les variables globales et leur contenu

■ setenv <variable> <valeur>

affecte une valeur à une variable globale

■ unsetenv <variable>

supprime une variable globale

■ echo <\$variable>

affiche le contenu d'une variable globale (idem locale)

Quelques variables d'environnement prédéfinies

■ HOME

chemin du répertoire d'accueil de l'utilisateur

■ PATH

liste des répertoires de recherche des commandes

■ SHELL

nom du shell utilisé à la connexion

■ LANG

langue utilisée pour l'affichage des messages

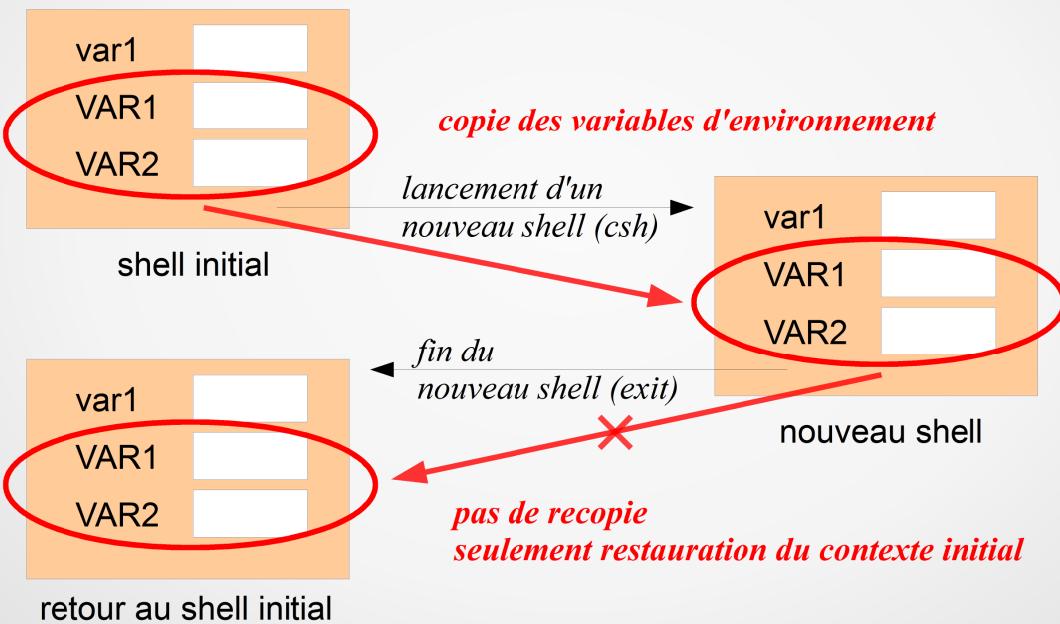
■ PRINTER

contient le nom de l'imprimante par défaut

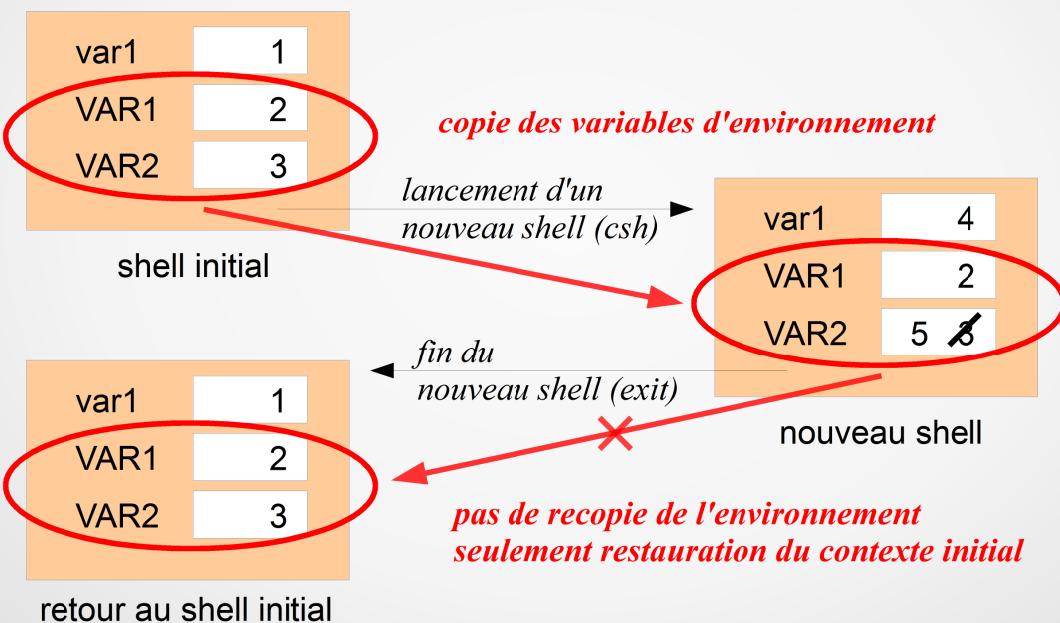
■ DISPLAY

nom de la machine servant à l'affichage

Portée des variables – Principe d'héritage



Portée des variables - Exemple



Redirections

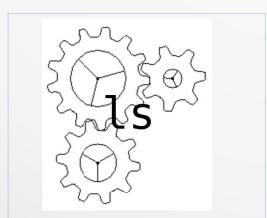
- Entrées/sorties standards
- Redirection des entrées/sorties
- Sortie erreur standard
- Les tubes
- Exemples

La sortie standard

Ce qui s'affiche par défaut lors de l'exécution d'une commande

```
dupond@@k-etu-052% ls
README.html
HEADER.html
dupond@@k-etu-052%
```

Sortie standard de la commande ls

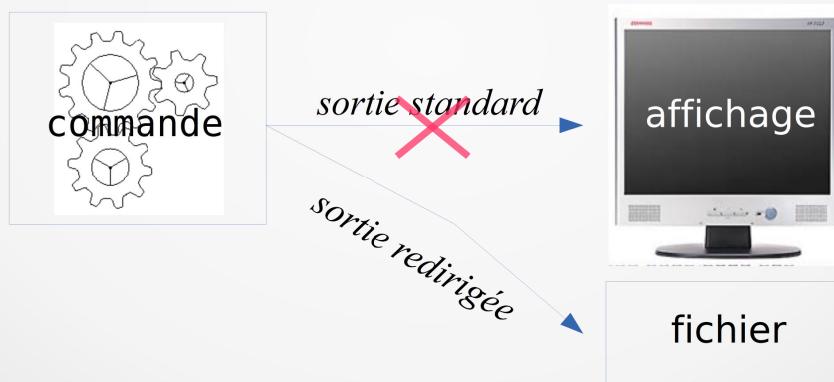


*sortie
standard*

README.html
HEADER.html

Redirection de la sortie standard – L'opérateur >

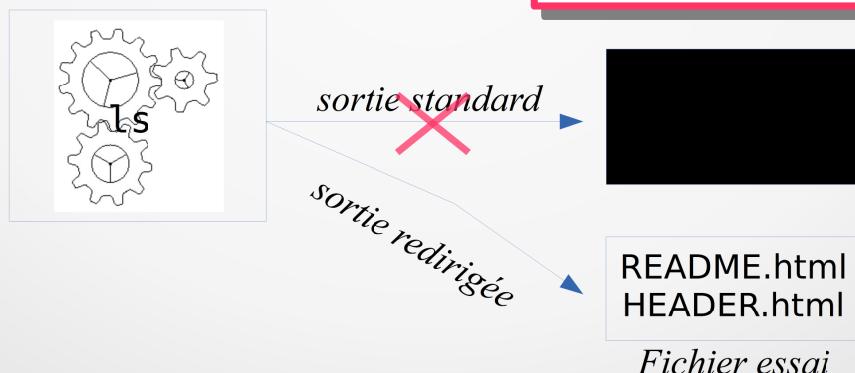
L'opérateur > redirige la sortie standard dans un fichier
Si le fichier existe déjà, il est écrasé



Redirection de la sortie standard – Exemple 1

```
dupond@@k-etu-052% ls  
README.html  
HEADER.html  
dupond@@k-etu-052% ls >essai  
dupond@@k-etu-052%
```

Rien n'est affiché !



Redirection de la sortie standard – Exemple 1 suite

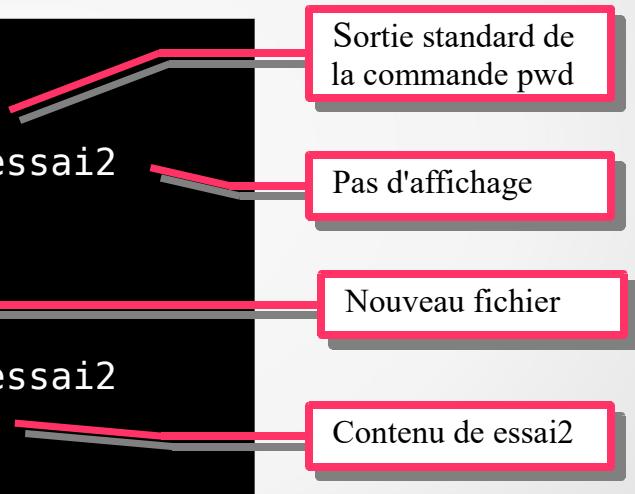
```
dupond@@k-etu-052% ls
README.html
HEADER.html
dupond@@k-etu-052% ls >essai
dupond@@k-etu-052% cat essai
README.html
HEADER.html
dupond@@k-etu-052%
```

cat (conCATenate) : commande multi-usage qui permet d'afficher, de créer, de copier et de concaténer des fichiers

NB : on aurait pu utiliser la commande more à la place de cat

Redirection de la sortie standard – Exemple 3

```
dupond@@k-etu-052% pwd
/home/dupond/public_html
dupond@@k-etu-052% pwd >essai2
dupond@@k-etu-052% ls
essai2
README.html
HEADER.html
dupond@@k-etu-052% more essai2
/home/dupond/public_html
dupond@@k-etu-052%
```



Redirection de la sortie standard – L'opérateur >>

L'opérateur >> redirige la sortie standard vers la fin d'un fichier existant
(concaténation)

Si le fichier n'existe pas, il est créé

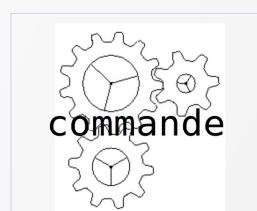
```
dupond@@k-etu-052% cat >>f1
Ajout de contenu à f1
<ctrl d>
dupond@@k-etu-052% more f1
Je saisis le contenu de f1
Ajout de contenu à f1
dupond@@k-etu-052%
```

L'entrée standard

Ce qui est saisi au clavier pour l'exécution d'une commande



entrée
standard



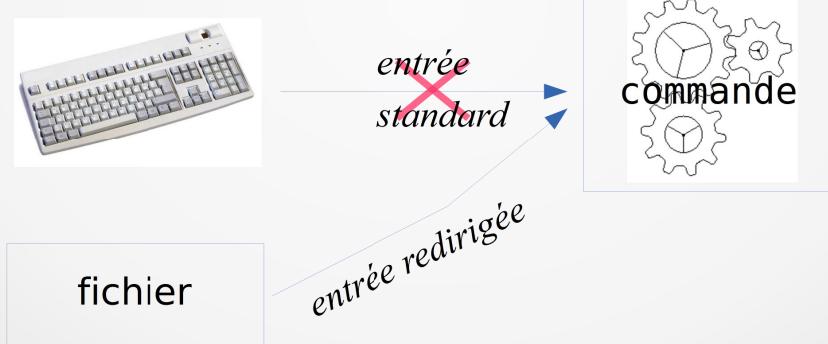
Redirection de la sortie standard – Exemple 2

```
dupond@@k-etu-052% cat >f1
Je saisir le contenu de f1
<ctrl d>
dupond@@k-etu-052% more f1
Je saisir le contenu de f1
dupond@@k-etu-052%
```

cat (conCATenate) : commande multi-usage qui permet d'afficher, de créer, de copier et de concaténer des fichiers

Redirection de l'entrée standard – l'opérateur <

L'opérateur < redirige la saisie (l'entrée standard) à partir d'un fichier (peu utilisé)



Redirection de l'entrée standard – Exemple 1

```
dupond@@k-etu-052% cat fic
Affichage du contenu de fic
dupond@@k-etu-052%
```

équivaut en fait à :

```
dupond@@k-etu-052% cat <fic
Affichage du contenu de fic
dupond@@k-etu-052%
```

Redirection de l'entrée standard – Exemple 2

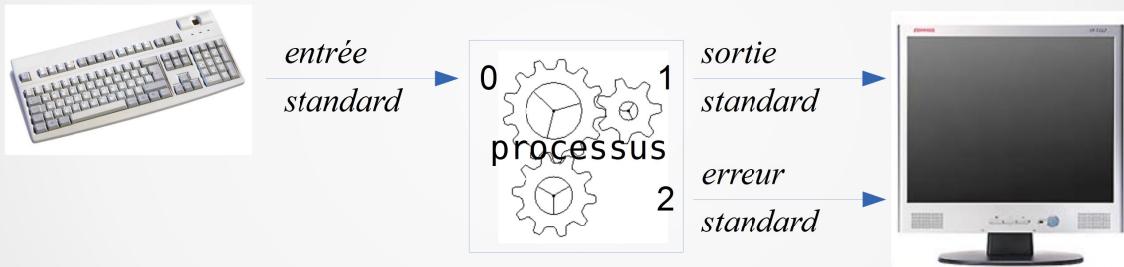
```
dupond@@k-etu-052% cat <fic1 >fic2
dupond@@k-etu-052%
```

équivaut à :

```
dupond@@k-etu-052% cp fic1 fic2
dupond@@k-etu-052%
```

Redirection des entrées/sorties d'un processus

- Cas général par défaut -

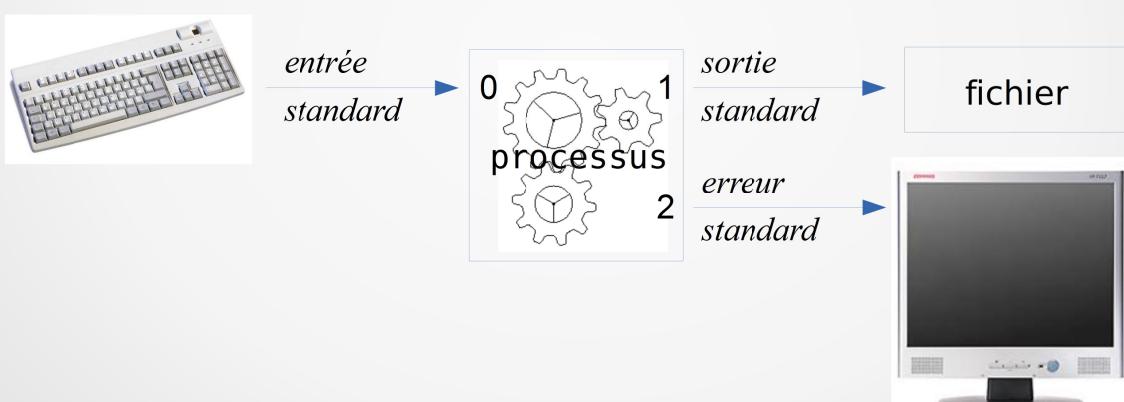


UNIX définit 3 canaux d'E/S, chacun pointant vers un périphérique par défaut :

- Canal 0 : entrée standard (par défaut le clavier)
- Canal 1 : sortie standard (par défaut l'écran)
- Canal 2 : erreur standard (par défaut l'écran)

Redirection des entrées-sorties d'un processus

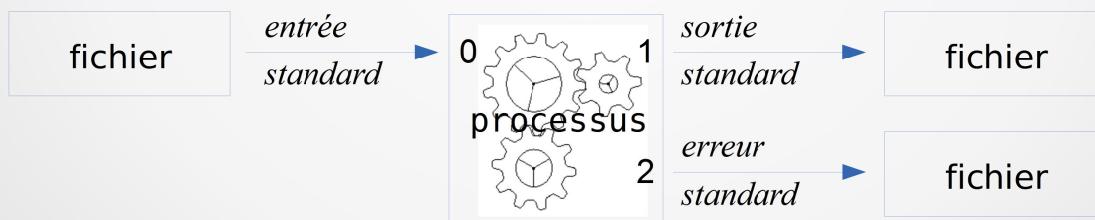
- Redirection de la sortie standard -



Redirection des entrées-sorties d'un processus

- Redirection de toutes les entrées/sortie standard -

Les opérateurs >& et >>& redirigent la sortie erreur standard dans un fichier



Redirection de l'erreur standard – Exemple

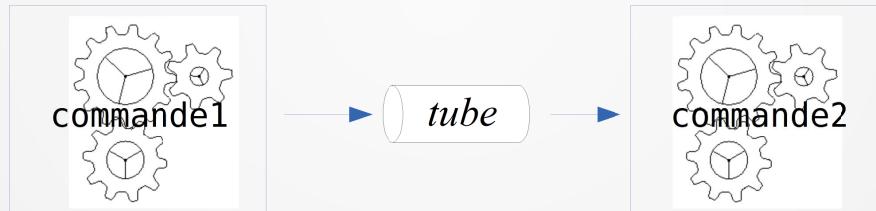
```

dupond@@k-etu-052% cat fic
Affichage du contenu de fic
dupond@@k-etu-052% cat ficinconnu
cat : Aucun fichier ou dossier de ce type
dupond@@k-etu-052% cat ficinconnu >>&log
dupond@@k-etu-052% cat log
cat : Aucun fichier ou dossier de ce type
dupond@@k-etu-052%
  
```

Pas d'affichage

Les tubes (pipe) - Principe

Redirection de la sortie standard d'une commande dans l'entrée standard d'une autre commande



commande1 | commande2

Les tubes (pipe) – Exemple 1

```
dupond@@k-etu-052% ps -aux
...
dupont 21744 1 0 Aug16 ?          00:00:38 gnome-terminal
dupont 21745 21744 0 Aug16 ?      00:00:00 gnome-pty-helper
root   31434 3096 0 Aug16 ?      00:00:03 spamd child
root   31793 3096 0 Aug16 ?      00:00:13 spamd child
dupont 9272 21744 0 Aug16 pts/1  00:00:00 csh
root   17675 1 0 06:26 ?          00:00:17 /usr/sbin/cupsd -F
dupont 23935 9272 0 10:44 pts/1  00:03:49 /usr/lib/libreoffice
dupont 24684 1 1 11:10 ?          00:07:36 /usr/lib/mozilla-fir
dupont 25653 1 0 11:40 ?          00:00:02 emacs
dupont 25660 25653 0 11:40 ?      00:00:00 /usr/bin/aspell -a
dupont 28465 21744 0 13:28 pts/2  00:00:00 csh
dupont 31731 23971 0 15:18 pts/4  00:00:00 emacs memo/plan.txt
dupont 31746 9272 0 15:18 pts/1  00:00:01 gpdf introinfo.pdf
...
```

Trop d'informations ! Long à lire ! 😞

Les tubes (pipe) – Exemple 1



Première idée : tout dans un fichier

```
dupond@@k-etu-052% ps -aux >essai.txt
dupond@@k-etu-052%
```

Puis recherche des informations pertinentes dans le fichier

```
dupond@@k-etu-052% grep csh essai.txt
dupont 9272 21744 0 Aug16 pts/1 00:00:00 csh
dupont 28465 21744 0 13:28 pts/2 00:00:00 csh
dupond@@k-etu-052%
```

Les tubes (pipe) – Exemple 1 suite



Meilleure idée : un tube (pas besoin de fichier intermédiaire)

Tous les processus contenant *csh* :

```
dupond@@k-etu-052% ps -aux | grep csh
dupont 9272 21744 0 Aug16 pts/1 00:00:00 csh
dupont 28465 21744 0 13:28 pts/2 00:00:00 csh
dupond@@k-etu-052%
```

La sortie de *ps* est redirigée dans l'entrée de *grep*

Tous les processus contenant *root* :

```
dupond@@k-etu-052% ps -aux | grep root
root 31434 3096 0 Aug16 ? 00:00:03 spamd child
root 31793 3096 0 Aug16 ? 00:00:13 spamd child
root 17675 1 0 06:26 ? 00:00:17 /usr/sbin/cupsd -F
dupond@@k-etu-052%
```

Les tubes (pipe) – Exemple 2

```
dupond@@k-etu-052% ls
image.png
index.html
estimation.txt
toto.sxc

dupond@@k-etu-052% ls | grep ima
image.png
estimation.txt
dupond@@k-etu-052%
```

La sortie de **ls** est redirigée dans l'entrée de **grep**

Dans la pratique...



Exécuter plusieurs commandes sur une même ligne

Syntaxe : commande1 ; commande2

```
dupond@@k-etu-052% echo bonjour ; ls
bonjour
HEADER.html README.html
dupond@@k-etu-052%
```

résultat "echo"

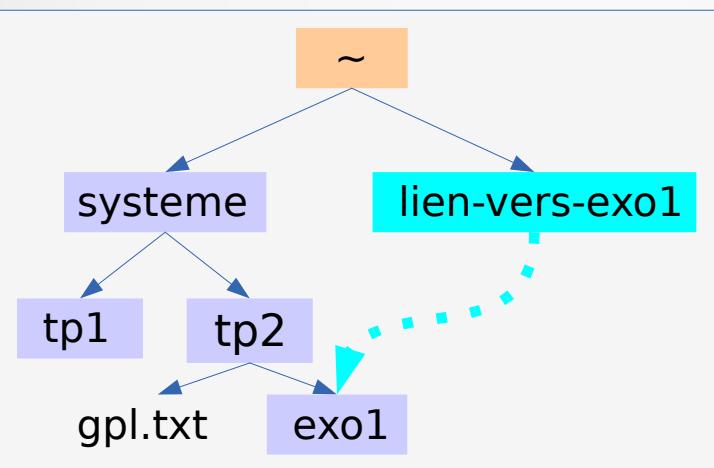
résultat "ls"

```
dupond@@k-etu-052% cd public_html ; ls ; echo bonjour
PUB.html DOCS.html
bonjour
dupond@@k-etu-052%
```

Lien symbolique

Raccourci pour nommer un fichier ou répertoire

```
ln -s systeme/tp2/exo1 lien-vers-exo1
```



Exemple :
cd ~/lien-vers-exo1

La commande wc (word count) – C'est un filtre

Compter le nombre de lignes, de mots et de caractères dans un fichier

Syntaxe : **wc [options] [fichier]**

- w compte le nombre de mots
- c compte le nombre de caractères
- l compte le nombre de lignes

La commande cut – C'est aussi un filtre

Couper en plusieurs champs des lignes d'un fichier ou de l'entrée standard

Syntaxe : **cut liste [fichier] | liste [dx] [fichier]**

- dx choix du caractère *x* comme séparateur de champ
(par défaut : *tab*)
- liste* position des caractères (en colonne, à partir de 1)
- liste* position des champs (à partir de 1)

Exemples de listes : 1,3,4,7 1-4 1-4,7,9,30- (du 30^{ème} au dernier)

La commande cut – Exemples

Affichage des portions de ligne du fichier *fic* à partir de la 20^{ème} position :

```
cut c20 fic
```

Affichage des champs 1 et 3 des lignes d'un fichier dont le caractère séparateur est le caractère ':' :

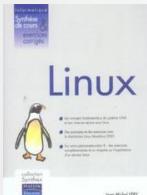
```
cut -d: -f1,3 /etc/passwd
```

La commande sed – C'est encore un filtre

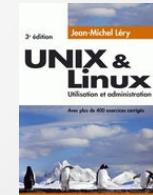
Voir TD-TP

Bibliographie

Armspach Jean-Paul, Colin Pierre, Ostré-Waerzeggers Frédérique.
1999. LINUX - Initiation et utilisation. 2^{ème} édition. Edition DUNOD.



Léry Jean-Michel. 2006. Linux. 3e édition 2011. Collection Synthex.
Edition PEARSON.



Léry Jean-Michel. 2011. UNIX & Linux - Utilisation et administration.
Avec plus de 400 exercices corrigés. 3e édition. Edition PEARSON.

Bosc Marcel. <http://www-info.iutv.univ-paris13.fr/~bosc>