

Licence Professionnelle

Remise à Niveau

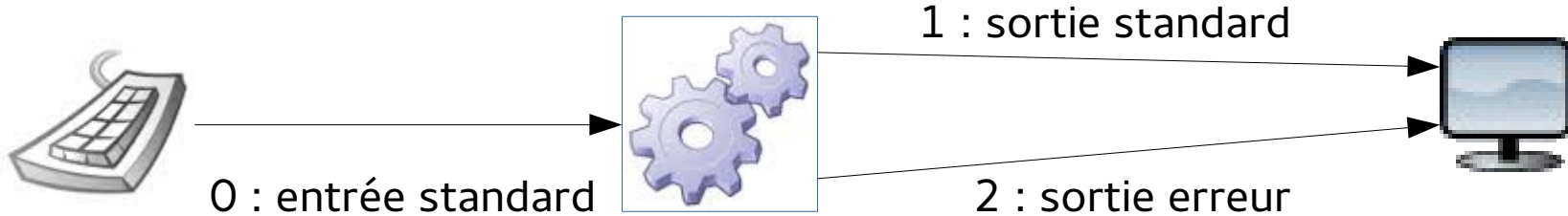
N°3 : Commandes

Commandes

- Redirections
- Filtres
- Expressions régulières

Redirection des E/S

➤ Les entrées/sorties standard



- La redirection consiste à modifier la connexion par défaut des canaux
 - on remplace le clavier et/ou l'écran par un fichier
- **>fichier** : crée un fichier vide ou l'efface
- **cat >fichier Ctrl+D** : crée un fichier contenant les caractères saisis entre la ligne de commandes et un Ctrl+D en début de ligne (*vide le fichier avant s'il existait*)
- **commande >fichier** : enregistre dans le fichier l'affichage de la commande
 - Exemple : `date +%A > jour` *# le fichier jour contient le nom du jour*

Redirections

- La syntaxe permet de spécifier le N° du canal redirigé
 - `cat *.sh 1>liste` # redirection de sortie (vide ou crée)
 - `cat *.sh 1>>liste` # redirection en ajout (crée si nécessaire)
 - `cat 0<liste` # redirection d'entrée
 - `cat *.sh 2>erreur` # redirection d'erreur

- Rediriger la sortie erreur vers la poubelle système
 - `find / -name "toto" 2>/dev/null`
évite l'affichage des répertoires inaccessibles

- Rediriger la sortie et les erreurs dans un même fichier
 - `find ~ -name ".*" 1>resultat 2>&1`

Tubes

- Le tube de communication (*pipe*) : *symbole* |
 - connecte le canal de sortie (N°1) d'un processus, au canal d'entrée (N°0) d'un autre processus
 - permet de connecter entre eux deux exécutable
(*éventuellement écrits dans des langages différents*)
 - *les résultats du 1^{er} sont les données pour le 2^{ème}*
- Exemple avec des commandes du shell :
 - afficher le nombre de lignes contenant la chaîne toto dans l'ensemble des fichiers .txt du répertoire courant

```
grep "toto" *.txt|wc -l
```

pas d'espace entre le pipe et la commande qui suit

Filtres

➤ Utilitaires destinés à être utilisés en amont/aval de l'exécution de commandes

➤ ils filtrent la sortie : ***post-traitement***

`ls -l | wc -l`

➤ wc compte le nombre de lignes de la réponse du ls -l

➤ => *compte le nombre de fichiers du répertoire courant*

➤ ils filtrent l'entrée : ***pré-traitement***

`grep "/bin/bash" /etc/passwd | wc -l`

grep filtre l'entrée du wc pour compter les lignes contenant /bin/bash

=> *compte le nombre de comptes lançant un bash*

Exemple de Filtres

- Les filtres peuvent en général être appelés de plusieurs manières
 - Exemple **more** : filtre d'affichage paginé
 - **more fichier** *comme une commande classique*
 - **more <fichier** *en redirigeant l'entrée*
 - **cat fichier |more** *en filtre de sortie*
- Principales commandes utilisables comme filtres
 - **grep** : recherche une expression dans les lignes d'un fichier
 - **sort** : tri ligne à ligne (ordre alphabétique par défaut)
 - **uniq** : suppression de doublons (lignes successives identiques)
 - **cut** : extraction de champs
 - **tr** : transcription
 - **head** : extrait les premières lignes du résultat
 - **tail** : extrait les dernières lignes du résultat
 - **tee** : réplique la sortie standard dans les fichiers spécifiés
 - **paste** : regroupe ligne à ligne les fichiers (en séparant par une tabulation)
 - **fold** : coupe chaque ligne à une longueur donnée...

Exemples

- `ls -li | sort -n`
 - *affiche la liste des fichiers triés dans l'ordre croissant de leur inode*
- `ls -li | sort -n | cut -d" " -f1`
 - *n'affiche que les inodes*
- `ls -li | cut -d" " -f1 | sort -n | uniq`
 - *ne conserve que les inodes différents*
- `ls -li | cut -d" " -f1 | sort -n | uniq | tr '\012' ' '`
 - *les affiche sur une seule ligne*
- `ps -aux | head`
 - *les 10 premiers processus lancés encore actifs*
- `last tail -12 | tr -s " " | cut -d" " -f1,5-7 | head`
 - *les 10 utilisateurs dernièrement connectés*

La commande grep

- **egrep** "*expression régulière*" *fichier*
 - affiche les lignes du fichier contenant une chaîne correspondant à une **expression régulière** (modèle de chaîne contenant des méta-caractères)
 - l'option **-v** permet d'afficher les lignes ne correspondant pas au modèle
- Penser à protéger l'expression régulière par des " " pour éviter l'expansion par le shell lui-même (*globbing*)
 - Exemple : `grep "user*" /etc/passwd`
 - *sans protection, user*, est remplacé par les noms des fichiers commençant par user du répertoire courant, avant d'exécuter le grep...*

Les expressions régulières

- Expressions régulières : moyen de définir un modèle (pattern) de chaînes
 - interprétation différente des méta-caractères (*% expansion Unix*)
 - Ex : `egrep "user*" /etc/passwd`
=> recherche user,usererr,usererrr...!!!
- Utilisées sous Linux notamment dans :
 - la commande **egrep** (recherche d'une expression)
 - l'éditeur **sed** (éditeur de flots)
- mais aussi en **PHP** par exemple
 - attention il existe différentes normes : POSIX (Unix), PCRE (Perl)

Meta-caractères

- ^** => début de ligne
- \$** => fin de ligne
- .** => n'importe quel caractère unique (. ~ ?)
- [...]** => n'importe quel caractère unique parmi ceux énumérés
- [a-z]** => n'importe quel caractère unique choisi dans l'intervalle
- [^ ...]** => n'importe quel caractère unique autre que ceux énumérés
- *** => zéro ou plus occurrences de l'élément précédent (.* ~ *)
- ** => inhibe l'interprétation du méta-caractère qui suit

Exemples

- `egrep "^a" fichier` => lignes commençant par a
- `egrep "^a*" fichier` => toutes les lignes !
- `egrep "...\$" fichier` => lignes contenant des mots de 3 caractères encadrés par un espace au début, et un \$ en fin
- `egrep "[A-Z].*[0-9]$" fichier` => lignes commençant par une majuscule et terminée par un chiffre
- `egrep "[^A-Z]" fichier` => lignes ne commençant pas par une majuscule

Expressions régulières étendues

- Apport du **egrep** par rapport au grep (~ grep -E)
- Contrôle des occurrences : répétition de l'élément qu précède
 - **+** : peut être rencontré une ou plusieurs fois (successivement)
 - **{n}** : doit être rencontré exactement n fois
 - **{n,}** : doit être rencontré n fois ou plus
 - **{n,m}** : doit être rencontré entre n et m fois

egrep "^-{2,5}x" texte.txt # lignes commençant 2 à 5 tirets suivi d'un x

- Opérateur logique **ou** (*pas de et*) : |
- Classes de caractères :

[[:alnum:]] [[:alpha:]] [[:digit:]] [[:lower:]] [[:upper:]] [[:punct:]] [[:space:]]

 egrep "t**[[:digit:]]**" texte.txt # lignes contenant un t suivi d'un chiffre

Interprétation

- Attention à bien spécifier le modèle dans son intégralité
- Exemple avec un répertoire où se trouvent les fichiers :
e, ee , eee, aea, aeea, aeeea, aeeeea, eae, eeaeee
- `ls | egrep "e{2}"` et `ls | egrep "e{2,}"` affichent le même résultat
e ee eee aea aeea aeeea aeeeea eae eeaeee
- `ls | egrep "e{2}a"` et `ls | egrep "e{2,}a"` affichent le même résultat
e ee eee aea aeea aeeea aeeeea eae eeaeee
- `ls | egrep "ae{2}a"` affiche e ee eee aea aeea aeeea aeeeea eae eeaeee
- `ls | egrep "ae{2,}a"` affiche e ee eee aea aeea aeeea aeeeea eae eeaeee
- `ls | egrep "ae{2,3}a"` affiche e ee eee aea aeea aeeea aeeeea eae eeaeee

xargs

- la commande **xargs** placée derrière un pipe permet d'exécuter une ligne de commandes avec **chacun** des paramètres envoyés sur l'entrée standard
- `find ~ -type f | egrep chaîne`
 # recherche les fichiers dont le nom contient la chaîne *chaîne*
- `find ~ -type f | xargs egrep chaîne`
 # recherche dans chaque fichier la chaîne *chaîne*
- `cat liste | xargs egrep chaîne`
 # recherche la chaîne dans chaque fichier énuméré dans le fichier *liste*

Groupement de commandes

- Les accolades **{ }** permettent de regrouper une série de commandes
- Composition avec les opérateurs suivants : **;, &, !!, &&, |**
 - **{ commande1 ; commande2; }**
 - **NPO** : le point virgule après la dernière commande
- Redirections
 - Permet d'appliquer une redirection à un ensemble des commandes


```
{ ls / ; ls ~ ; } | egrep ".t."
```

liste les fichiers de la racine et du répertoire de connexion, contenant un t qui ne soit ni au début, ni à la fin.

Expressions génériques du bash

- Activation explicite nécessaire par la commande
 - **shopt** -s extglob

- Expressions génériques
 - **?(liste de modèles)** : 0 ou une occurrence de chaque modèle
 - ***(liste de modèles)** : 0 ou plusieurs occurrence de chaque modèle
 - **+(liste de modèles)** : >1 une occurrence de chaque modèle
 - **@(liste de modèles)** : une occurrence de chaque modèle
 - **!(liste de modèles)** : tout sauf occurrences du modèle

- Exemple
 - **shopt** -s extglob
 - **ls +([[:lower:]])** : noms de fichiers constitués uniquement de minuscules