



CS230

TP 1 Programmation Shell

Découverte de l'environnement de travail,
manipulation des commandes et écriture de scripts

A rendre dans une semaine à partir d'aujourd'hui :

- Un compte-rendu explicatif en format PDF, à déposer sur chamillo (partie "Travaux").
- Le compte-rendu doit comporter les réponses aux différentes questions, les éventuelles remarques et détails des problèmes rencontrés, les scripts Shell et les résultats d'exécutions.

Table des matières

| | |
|---|----------|
| I Aspects Généraux du Shell | 2 |
| 1 Connaître et personnaliser son environnement | 2 |
| 2 L'éditeur de texte vi | 2 |
| 3 Commandes de configuration | 3 |
| II Quelques commandes de base | 4 |
| 4 Manipulation de fichiers | 4 |
| 4.1 Création de répertoires | 4 |
| 4.2 Création de fichiers | 4 |
| 4.3 Redirections des E/S et des erreurs | 4 |
| 4.4 Redirection entre commandes | 5 |
| 4.5 Affichage des informations relatives aux fichiers | 5 |
| 4.6 Copie de fichiers | 5 |
| 4.7 Liens | 5 |
| 4.8 Droits d'accès | 5 |
| 4.9 Arborescence | 6 |
| 4.10 Caractères génériques | 6 |
| 4.11 Comparaison de fichiers | 6 |
| 5 Les filtres | 6 |
| 5.1 Commande find | 6 |
| 5.2 Commande grep | 7 |
| 5.3 Commandes tee, head, tail, wc, tr et sed | 7 |

Première partie . Aspects Généraux du Shell

1 Connaître et personnaliser son environnement

Lors de la connexion d'un utilisateur, un processus correspondant à un interprète shell est lancé. Il s'agit du login shell.

Exemple : l'exécution de la commande `ps` après la connexion affiche les informations relatives au *login shell*.

```
$ ps
PID      TTY      TIME    CMD
26470    pts/16  0.00    bash
$
```

Le login shell exécute à son lancement un certain nombre de commandes lues dans des fichiers particuliers (*initialisation de certaines variables d'environnement*, ...). Les principaux fichiers d'initialisation sont les fichiers `/etc/profile` et `~/.bash_profile` qui sont lus quand le shell est invoqué comme shell interactif de connexion. Le fichier `~/.bashrc` est lu quand le shell est invoqué comme shell interactif sans fonction de connexion. À la déconnexion, `bash` exécute le fichier `~/.bash_logout`.

Certaines de ces variables sont initialisées automatiquement : `HOME`, `USER`, `EUID`, `UID`. Il est conseillé de ne pas les modifier. D'autres sont également initialisées automatiquement mais peuvent être modifiées : `SHELL`, `PATH`, `BASH`. D'autres peuvent être initialisées par l'utilisateur dans son fichier *.profile*. La valeur de ces différentes variables peut être consultée par la commande `echo`.

```
$ echo $BASH
/bin/bash
$ echo $BASH_VERSION
4.2.24(1)-release
$ echo $EUID
157
```

- Afficher le nom du terminal associé à l'entrée standard. Vérifier cela avec les commandes `who` et `who am i`, la consultation de la variable d'environnement `TERM`. Puis changer de `tty` en utilisant `ctrl+alt+(de f1 à f6 pour les console et f7 pour le serveur X s'il est lancé)`.
- Afficher et interprétez les valeurs des variables d'environnement suivantes : `PATH`, `HOST`, `SHELL`, `HOSTNAME`, `HOSTTYPE`, `PS1`, `PWD`.

Sous *bash*, les variables peuvent être initialisées de la manière suivante : `nom_varriable=valeur`.

Exemple :

```
$ SHELL=/bin/tcsh
```

Pour certaines variables, nous pouvons avoir plusieurs valeurs possibles. Dans ce cas, on rajoute une nouvelle valeur en utilisant la syntaxe suivante : `nom_varriable=${nom_variable}:valeur`

La commande `export -p` permet de connaître la liste des variables exportées.

- Modifiez la valeur de la variable d'environnement `PATH` de manière à y rajouter le répertoire courant.
- Comment pouvez-vous procéder pour faire en sorte que la commande `cd` vous ramène par défaut dans le répertoire `TP1` par exemple ? Testez votre solution.

Remarque : les modifications des variables d'environnement sont temporaires (jusqu'à la fin du shell courant). Pour les rendre permanentes, il faut les ajouter au fichier `.bash_profile`.

2 L'éditeur de texte vi

`vi` est un éditeur très populaire mais pour certains archaïque. Cet éditeur qu'on est pratiquement sûr de retrouver sur toutes les machines Unix doit sa popularité sans doute à ses très bonnes performances en terme de rapidité et de faible consommation d'espace mémoire. Trois modes sont permis par `vi` : le mode d'insertion, le mode de commande et le mode de commande ligne.

Le mode initial est le mode de commande. Ce mode est utilisé pour déplacer le curseur ainsi que pour la majorité des commandes d'édition (suppression, déplacement, répétition, etc.)

Le mode d'insertion vous permet d'insérer du texte.

Le mode de commande ligne vous permet des possibilités d'édition assez avancées.

En consultant le manuel en ligne d'Unix, effectuez les manipulation suivantes sous vi :

1. Insérez un texte d'une vingtaine de ligne.
2. Sauvegardez le texte inséré (en cours d'insertion, puis à la fin de la saisie)
3. Déplacez le curseur.
4. Supprimez des mots.
5. Supprimez des lignes (1 ligne, puis 2 puis 3)
6. Déplacez et dupliquez 1 ligne, 2 lignes puis 3 lignes à la fois.
7. Annulez une opération précédente.
8. Recherchez toutes les occurrences d'un mot dans le texte.

3 Commandes de configuration

La plupart des commandes de configuration sont exécutées à la connexion ou lors du lancement d'un nouveau shell. **Etudiez les commandes suivantes avec le man, testez-les et commentez leur utilisation :**

stty : est utilisée pour initialiser les paramètres du terminal. L'option **-a** permet d'afficher l'ensemble des paramètres du terminal. Il est bien entendu possible de modifier ces caractères de contrôle en utilisant **stty**, par exemple : **stty erase "^H"**, affecte **Ctrl-H** à l'effacement de caractère. De même, on peut modifier le paramétrage de la ligne : vitesse, nombre de bits de données, parité, traitement des signaux de contrôle.

umask : est utilisée pour définir les droits d'accès par défaut lors de la création de fichiers ou de répertoires (par défaut est à 022). A savoir que la durée de vie de la commande **umask** est limitée à la session shell en cours. Vous pouvez définir sa valeur dans les fichiers **/etc/bashrc** or **/etc/profile** comme suit.

```
vi ~/.bashrc
```

```
vi ~/.profile
```

Il y a trois groupes d'utilisateurs pour chaque fichier ou répertoire.

[Propriétaire Groupe Autres] par exemple : 077

Les valeurs des permissions sont les suivantes :

| | | |
|-----------------------------|-----------------------|----------------------|
| 0 : read, write and execute | 1 : read and write | 2 : read and execute |
| 3 : read only | 4 : write and execute | 5 : write only |
| 6 : execute only | 7 : no permissions | |

Il est également possible d'affecter des droits avec les options **u,g,o**, comme par exemple :

```
umask u=rwx,g=,o=
```

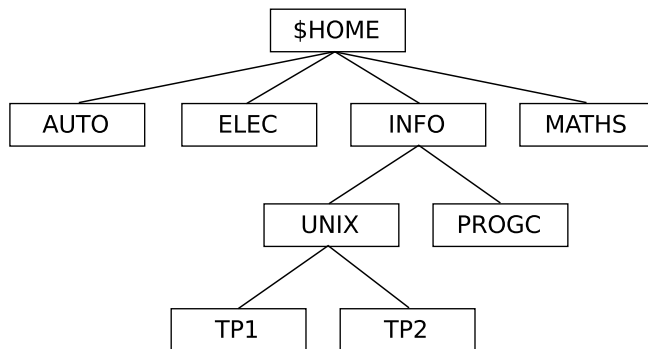
Changez les droits d'accès en utilisant **umask** puis utiliser la commande **touch** pour créer des fichiers et vérifiez en utilisant la commande **ls -l** si ces droits ont été appliqués.

Deuxième partie . Quelques commandes de base

4 Manipulation de fichiers

4.1 Création de répertoires

1. Affichez le chemin absolu de votre répertoire privé.
2. Créez l'arborescence suivante à partir de votre répertoire privé :



3. En partant du répertoire `home`, déplacez-vous dans le répertoire `TP1`.

4.2 Création de fichiers

Pour créer un fichier, on utilise un éditeur de texte. Il existe plusieurs éditeurs de texte, parmi lesquels `vi` (éditeur historique d'Unix) et `emacs` (ou `Xemacs`). Sans avoir recours à un éditeur de texte, il est possible de créer un fichier à l'aide des 2 commandes suivantes :

- commande `touch`
- commande `cat`

1. Créez un fichier `file1` en utilisant la commande `touch` dans le répertoire `TP1`.
2. Visualisez le contenu du répertoire courant.
3. Créez un fichier `file2` en utilisant la commande `cat` comme suit :

```
$ cat > file2
Ceci est mon premier fichier sous Unix
CTRL-D
```

4.3 Redirections des E/S et des erreurs

1. Visualisez la liste des utilisateurs connectés en utilisant la commande `who`.
2. Insérez le résultat de la commande `who` dans un fichier `file3`.
3. Visualisez le contenu du fichier `file3`.
4. Tapez la commande `pwd > file3`
5. Visualisez le contenu du fichier `file3`. Que remarquez-vous ?
6. Tapez la commande `ls >> file3`
7. Visualisez le contenu de `file3`. Que remarquez-vous ?
8. Tapez les commandes suivantes et notez vos observations
`cat file3 >> file3`
`cat < file3`
9. Tapez la commande `ls-1` (ne pas laisser d'espace entre le nom de la commande et l'option pour générer une erreur)
10. Tapez la commande `ls-1 > file3`. Que remarquez-vous ?
11. Tapez la commande `ls-1 2> file3`. Que remarquez-vous ?

4.4 Redirection entre commandes

Nous allons dans cette partie compter le nombre de fichiers contenus dans un répertoire. Une façon de procéder consiste à lister le contenu du répertoire en redirigeant la sortie sur un fichier `file4`. Une fois ce fichier créé, on peut appliquer dessus la commande `wc` avec l'option `-l` qui permet de compter le nombre de ligne du fichier. Le résultat correspond au nombre de fichiers dans le répertoire.

1. Effectuez cette manipulation selon les indications ci-dessus.
2. Quel est le rôle assuré par le fichier `file4` ici ?
3. Pour éviter d'avoir recours à un tel fichier, il est possible d'utiliser la redirection entre commandes dont la syntaxe est la suivante :
`commande1 | commande2 | ... | commandeN`
Appliquez ce principe pour déterminer le nombre de fichiers dans un répertoire.

4.5 Affichage des informations relatives aux fichiers

1. Etudiez la commande `ls` et ses différentes options.
2. En utilisant la commande `ls`, déterminez
 - (a) les caractéristiques de tous les fichiers du répertoire courant
 - (b) le type des fichiers du répertoire courant
 - (c) l'*i-nœud* du répertoire courant

4.6 Copie de fichiers

1. Copiez le fichier `file4` dans un nouveau fichier `file5`.
2. Modifiez le fichier `file5` en tapant la commande `pwd >> file5`
3. Visualisez le fichier `file4`. Quel effet a eu la modification de `file5` sur `file4` ?
4. Supprimez le fichier `file5`.

4.7 Liens

1. Créez un lien physique `file6` sur le fichier `file4`.
2. Affichez les caractéristiques de ces 2 fichiers au format long. Que remarquez-vous ?
3. Modifiez le fichier `file6` en tapant la commande `pwd >> file6`
4. Visualisez le fichier `file4`. Que remarquez-vous ?
5. Supprimez le fichier `file4`.
6. Visualisez les caractéristiques du fichier `file6` au format long. Que remarquez-vous ?
7. Déplacez-vous dans le répertoire `TP2`.
8. Créez un lien symbolique `lien_symb` sur le fichier `file6` du répertoire `TP1`.
9. Visualisez les caractéristiques au format long du fichier `lien_symb`.

4.8 Droits d'accès

1. Revenez au répertoire `TP1`
2. Affichez le masque courant.
3. Vérifiez l'utilisation de ce masque sur les droits des répertoires et des fichiers créés.
4. Modifiez le masque pour que les droits à la création soient `rwX--X---`
5. Créez un fichier `file7` et un répertoire `REP1` dans le répertoire `TP1` pour vérifier l'utilisation du nouveau masque.
6. Modifiez les droits du fichier `file7` pour le rendre exécutable par toutes les catégories d'utilisateurs. Utilisez le mode absolu.
7. En utilisant le mode symbolique, rajoutez le droit de lecture sur le fichier `file7` pour les autres utilisateurs (`others`) et supprimez ce droit pour le groupe.

4.9 Arborescence

Utilisez les commandes `cd` et `ls` pour vous déplacer et visualisez le contenu des différents répertoires du système (`/`, `/bin`, `/etc`, `/dev` ...)

1. Déterminez le propriétaire des fichiers se trouvant sous `/bin`
2. Modifiez les droits d'un fichier de votre choix sous `/bin`. Commentez le résultat obtenu.

4.10 Caractères génériques

Dans le dossier `/usr/bin`

1. Listez les fichiers dont le nom est formé par exactement quatre lettres.
2. Listez les fichiers dont le nom est formé par au-moins quatre lettres.
3. Listez les fichiers dont le nom contient le caractère `-`.
4. Listez les fichiers dont le nom contient le caractère `-` placé en 3ème position.
5. Listez les fichiers dont le nom commence par la lettre `a`, `b` ou `c`.
6. Listez les fichiers dont le nom ne commence pas par la lettre `a`, `e`, `i`, `o` ou `u`.
7. Listez les fichiers dont le nom ne commence pas par une lettre comprise entre `c` et `v`.

Dans le dossier `/usr/include`

1. Listez les fichier dont le nom contient un `i`.
2. Listez les fichier dont le nom peut contenir un `i` mais pas en première position et qui se terminent par `.h`
3. Listez les fichier dont le nom contient la chaîne `"std"`.

4.11 Comparaison de fichiers

Etudiez les commandes `diff` et `cmp`.

5 Les filtres

5.1 Commande find

La commande `find` permet d'effectuer une recherche récursive dans une arborescence de fichiers.

- Revenez au répertoire `home`.
- A partir de ce répertoire, recherchez le fichier `file7` en tapant la commande :
`find . -name file7 -print`
- Pour rechercher les répertoires d'une arborescence et modifier leurs droits, on peut utiliser la commande `find` comme suit :
`find . -type d -exec chmod 755 {} \;`
Dans ce cas, cette commande sera appliquée à l'arborescence démarrant au répertoire courant, représenté ici par le `.`

1. Etudiez la commande `find` et les différentes possibilités qu'elle offre.
2. Recherchez, en une seule commande, tous les fichiers qui contiennent dans leur nom la chaîne de caractères « `gre` » dans les répertoires `/bin` et `/usr/bin`.
3. Recherchez dans votre dossier personnel tous les fichiers qui ont été modifiés depuis moins de 2 jours.
4. Recherchez, dans les répertoires `/bin` et `/usr/bin`, en une seule commande, tous les fichiers réguliers exécutables par tous et affichez les informations relatives à ceux-ci sous cette forme :

N° d'inoeud droit en octal nom du fichier avec son chemin

5. Recherchez dans votre répertoire personnel les fichiers qui ont une taille inférieure à 250ko.

5.2 Commande grep

Pour cet exercice, toutes les recherches vont se faire dans les fichiers `.h` du répertoire `/usr/include`.

1. Recherchez toutes les lignes qui commencent par une lettre minuscule.
2. Recherchez toutes les lignes qui commencent par trois lettres minuscules et qui se terminent par trois caractères qui ne sont pas des lettres minuscules.
3. Recherchez toutes les lignes qui ne contiennent pas de lettre majuscule.
4. Avec la commande `grep -v`, recherchez toutes les lignes qui ne contiennent pas la chaîne de caractères « `printf` ».
5. Recherchez toutes les lignes qui contiennent une succession de trois caractères identiques compris entre `a` et `e`.
6. Recherchez toutes les lignes qui contiennent un nombre compris entre 400 et 450 exclu.

5.3 Commandes tee, head, tail, wc, tr et sed

1. Etudier la commande `ps -aux | tee log.txt`
2. A l'aide de la commande `head`, récupérez les deux premières lignes du fichier `log.txt` créé avec la commande précédente. A quoi correspondent ces lignes ?
3. A l'aide de la commande `tail`, récupérez les 10 dernières lignes du fichier `log.txt`.
4. A l'aide de la commande `wc`, comptez le nombre de lignes du fichier `log.txt`.
5. Etudiez la commande :
`tail -n 20 exercice4.txt | tr -s '[:space:]*' ' '.`
6. Etudiez la commande :
`cat exercice4.txt | sed 's/\([[:alpha:]]*\)\(.*\) /Utilisateur \1/'.`
7. A partir d'un fichier `vols.txt` de la forme suivante :

| |
|------------------------------------|
| AirFrance 12h30 Paris Londres |
| Alitalia 07h44 Rome Paris |
| AirFrance 14h50 Marseille Paris |
| Sterling 09h13 Chambéry Copenhague |

Créez un fichier de la forme : `heureDépart De villeDépart A villeArrivée (compagnie)`

Exemple : `12h30 De Paris A Londres (AirFrance)`