Introduction aux systèmes d'exploitation (IS1)

TP 6 – Variables et Archives

Variables

En plus des usages que vous avez vus jusqu'ici, le shell est un véritable langage de programmation. en particulier, comme en Java, il est possible d'utiliser des *variables*.

Une variable est repérée par un nom appelé *identificateur*, qui peut être n'importe quelle suite de caractères commençant par une lettre ou le caractère « souligné » ('_') et ne contenant que des lettres, des chiffres ou le caractère souligné[1].

Chaque variable possède une *valeur*, qui peut être n'importe quelle chaîne de caractères. Par exemple, la valeur de la variable `SHELL` est `/bin/bash`.

On peut donner une valeur à la variable `var` avec l'instruction `var="`*valeur*"` (sans espace avant ni après le signe '='). On accède à la valeur associée à une variable en faisant précéder son identificateur du symbole `$`, comme par exemple dans la commande `echo $var` qui affiche la valeur de la variable `var`.

Exercice 1 – *Familiarisation avec les variables.*

1. Vérifiez que la variable `SHELL` a bien pour valeur `/bin/bash`.
2. Déclarez une variable `NOM` contenant votre nom complet (pensez à mettre des guillemets autour s'il comporte des espaces), et affichez-la pour vérifier que l'affectation est correcte.
3. La commande `set` utilisée sans argument affiche la liste de toutes les variables connues par le shell. Testez cette commande et repérez votre variable dans la liste.
4. Écrivez une ligne de commande qui affiche « `Bonjour, Machin Chose !` » (si vous vous appelez Machin Chose, bien sûr) en utilisant la variable `NOM`.
5. Depuis le terminal courant, ouvrez un nouveau shell en tapant la commande `bash`, puis affichez la valeur de la variable `NOM` (quand vous avez terminé, tapez `exit` pour revenir au shell précédent). Faites de même dans un nouveau terminal lancé depuis la barre d'icônes du bureau.

   Qu'en déduisez-vous sur la portée des variables du shell ?

Dans certains cas, on souhaite que la valeur d'une variable soit accessible à d'autres processus du système. De telles variables sont appelées variables *d'environnement*.

6. Affichez la liste des variables d'environnement grâce à la commande `env`. Que remarquez-vous par rapport à la sortie de la commande `set` ?
7. On transforme une variable `var` en variable d'environnement grâce à la commande `export var`. Transformez `NOM` en variable d'environnement et répétez la question 5.

   Qu'en déduisez-vous sur la portée des variables d'environnement ?
8. La commande `cd`, que vous connaissez bien, permet de changer de répertoire courant. Utilisée sans argument, elle fixe le répertoire courant au répertoire personnel de l'utilisateur. Ce comportement est en fait déterminé par la valeur d'une certaine variable d'environnement.

   Repérez cette variable dans la liste des variables d'environnement, et faites en sorte que la commande `cd` renvoie par défaut vers le répertoire racine.

[1] À l'exception de certaines variables spéciales du shell.

Archives

Comme vous l'avez déjà remarqué, il n'est pas possible d'attacher à vos messages électroniques des répertoires entiers. Il existe une possibilité de contourner cela en utilisant la commande `tar` (pour **t**ape **ar**chive : on l'utilisait initialement pour faire des sauvegardes sur bande magnétique). Voici un résumé succinct de son utilisation :

- `tar -c` fic_1 ... rep_n ... écrit sur la sortie *standard* un fichier archive contenant tous les fichiers et répertoires indiqués.
- `tar -x` extrait d'un fichier archive donné en ligne (donc, sur l'entrée *standard*) tous les fichiers et répertoires y figurant.
- Dans `FreeBSD`, l'entrée et la sortie *standard* représentent par défaut la "bande magnétique", i.e. le périphérique `/dev/sa0`. Si l'on veut utiliser les entrées et sortie standard du terminal, il faut utiliser l'option `-` (tout court). Ainsi par exemple :
  - `tar -c Bonjour.java Aurevoir.java` tente d'archiver ces deux fichiers sur le périphérique `/dev/sa0`
  - `tar -c -f - Bonjour.java Aurevoir.java` produira la même archive, mais dans le terminal cette fois-ci.

Exercice 2 – *Tar*

1. Recopiez le fichier `arch-tp6.tar` situé sur la page web du cours. Extrayez l'arborescence qui s'y trouve.
2. L'option `-t` de `tar` permet d'examiner le contenu d'un fichier (lu sur l'entrée standard) sans pour autant l'extraire. Que contient l'archive `test.tar` ? Quelle différence y a-t-il entre l'examen de cette archive et son extraction suivie de la commande `cd test ; ls -1` (le chiffre 1) ? Comment demander à `ls` d'afficher tous les fichiers cités par `tar` ? Comment demander à `tar` d'afficher un résultat proche de `ls -l` (la lettre minuscule l) ?
3. Utiliser les redirections pour écrire dans une archive ou pour en lire une est assez peu commode. Comment peut-on préciser le nom du fichier archive que l'on veut créer/lire ?
4. Créez les quatre fichiers suivants : `J'ai`, `*presque*`, `fini`, `l'exercice un !` (les virgules ne doivent pas faire partie du nom des fichiers, tout le reste, si). Puis archivez-les dans un fichier `exo1.tar` sans utiliser les redirections. Vérifiez-en le contenu.

Les fichiers produits étant parfois très gros, il est souvent utile de les compresser avant de les envoyer. C'est le but de la commande `gzip`. La commande réciproque `gunzip` permet de décompresser (on peut aussi utiliser l'option `-d` de `gzip`). Notez que cette commande s'attend à ce que les fichiers compressés aient une extension `.gz` ou `.tgz` (s'il s'agit d'une archive compressée, comme abréviation de `.tar.gz`).

Exercice 3 – *Compresser*

1. Toujours dans le répertoire `tp6`, décompressez l'archive `exo2.tar.gz` puis extrayez son contenu.
2. Au lieu de taper deux commandes distinctes, on peut utiliser les redirections : si aucun argument n'est donné à `gzip` ou `gunzip`, ceux-ci lisent et écrivent sur les entrées/sorties standard. Attention cependant : si on redirige l'entrée sur `gzip` il faut rediriger aussi la sortie.

   En une seule ligne archivez puis compressez les fichiers
   - `exo 2 1 question 2 a optimiser .`
   - `exo 2 2 question 2 a optimiser .`
   - `exo 2 3 quastion 2 a optimiser .`

   dans un fichier que vous nommerez `res` possédant les bonnes extensions.

3. Examinez le contenu de l'archive compressée pour vérifier qu'elle contient bien les fichiers voulus, toujours en utilisant les redirections.
4. Une certaine option de la commande `tar` permet de faire le même travail qu'à la question 2. Quelle est-elle ?
5. Il existe aussi les commandes `bzip2` et `bunzip2`. Décompressez l'archive de la question 1, puis utilisez `bzip2` pour la compresser. Le taux de compression est-il meilleur avec cette nouvelle commande ?
6. Essayez de compresser le fichier `exo2.tar.gz` avec `bzip2`. Que constatez-vous concernant la taille de l'archive compressée ? Décompressez l'archive en tapant une seule commande, puis essayer de la compresser en faisant le contraire (`bzip2` puis `gzip`). Que constatez-vous ?
7. Tentez le record du monde de taux de compression en compressant un fichier de votre choix (ne dépassant toutefois pas 100 Mo : prière d'utiliser le répertoire local `/tmp` pour ces expériences et de donner comme nom à votre fichier votre nom de login pour éviter les conflits). On pourra utiliser d'autres outils que `gzip`, à rechercher avec la commande `apropos` par exemple. Quels outils pouvez-vous utiliser pour générer un fichier sur lequel vous exercez ?

Les outils de messagerie modernes ont des limites de quelques Mo (1 à 5 en général) par message. La commande `split` permet de découper un fichier qui serait toujours trop volumineux en plusieurs morceaux.

Exercice 4 – *Split*

1. Par défaut, la commande `split` copie 1000 lignes du fichier d'entrée dans chaque fichier de sortie. Essayez de découper un fichier quelconque (programme java ou fichier texte de préférence) de moins de 1000 lignes. Que constatez-vous ?
2. En regardant le manuel de la commande, essayez de copier 10 lignes de votre fichier dans chaque fichier de sortie. Que constatez-vous ?
3. Comment la commande `split` nomme-t-elle les fichiers obtenus ? Peut-on changer ces noms ?
4. Recomposez votre fichier original.