

TP 1

Prise en main, premières commandes

RAPPELS

- ce TP ne donne pas lieu à remise d'un compte-rendu, et n'est pas noté
- sa correction sera diffusée par e-mail à la date figurant dans l'échéancier

IMPORTANT

Toutes les commandes ci-dessous ne sont valides qu'en shell standard, sh. Or, sh n'est pas votre shell par défaut sur les machines ESIEE, mais tcsh. Deux solutions existent pour corriger ce problème :

1. lancer sh, ou un shell 100% compatible avec lui (bash, ksh) dans chaque nouveau terminal que vous ouvrez
2. demander la commutation systématique vers bash ou ksh lorsque le shell est interactif, ce qui peut se faire en ajoutant la ligne suivante à votre fichier .tcshrc (lu à chaque initialisation de shell) :

```
echo 'if ( `set | grep edit` == edit ) exec /bin/bash' >> ~/.tcshrc
```

La deuxième solution évite de devoir taper « bash » dans chaque terminal, mais elle vous prive de la possibilité d'avoir un C-shell interactif.

PREPARATION

Nous allons commencer par extraire une archive contenant quelques fichiers de travail pour le reste du TP. Ouvrez un nouveau terminal, puis entrez la ligne suivante :

```
tar -xvpzf ~hilairex/public_html/IN3R21-TPx-fichiers.tgz
```

Vérifiez que le répertoire fichiers-tp a bien été créé sur votre compte.

QUESTIONS

- 1) Entrez la ligne de commande suivante :

```
echo ~ $HOME ~xxx ~hilairex "~hilairex" ~bin ~machin
```

où xxx est votre login. Comment expliquez-vous le résultat affiché ?
- 2) Que font les quatre dernières lignes de commandes suivantes ?

```
VAR="une:liste:de:chaines de caracteres:delimites"  
echo $VAR  
echo ${VAR%:*}  
echo ${VAR%%:*}
```

```
echo ${VAR#*:*}  
echo ${VAR##*:*}
```

- 3) Quel est l'effet de la ligne de commande suivante :

```
echo /e?*/d
```

Existe-t'il une différence entre `echo /e?*/d` et `echo /e?*/????*d` ?

- 4) A l'aide de la commande `cd`, placez-vous dans le répertoire `fichiers-tp`. Vérifiez que la commande `pwd` vous renvoie bien ce répertoire. Listez les fichiers de ce répertoire avec la commande `ls`. Quelle différence existe-t'il entre `ls` et `ls -l` ? Entre `ls -l` et `ls -la` ? Entre `ls -l` et `ls -lL` ?
- 5) La commande `xterm` permet de lancer un nouveau terminal à partir du shell (pour quitter ce nouveau terminal, vous pouvez indifféremment fermer l'entrée standard par la combinaison de touches Ctrl-D ou entrer la commande `exit`). Quelle différence observez-vous entre `xterm` et `xterm &` ?
- 6) La commande `cat` lit le ou les fichiers passés en arguments et les affiche sur la sortie standard. Examinez le résultat de `cat acces.txt`. Puis lancez un deuxième terminal, et demandez à votre binôme d'exécuter `ssh login@localhost` où `login` est son nom de login. Vous êtes à présent deux utilisateurs sur la même machine, ce que vous pouvez vérifier avec la commande `w`. Quelle commande votre binôme doit-il lancer pour voir votre fichier `acces.txt` ? Y parvient-il ? Dans la négative, que devez-vous changer et comment pouvez-vous le faire ?
- 7) Placez-vous à présent dans le répertoire `fichiers-tp/maint`. Ce répertoire contient un programme de test appelé `maint`, qui affiche diverses informations, puis ajoute son ou ses arguments sur un fichier texte `maint.txt`. Examinez le code source C `maint.c` pour comprendre ce qu'il fait. Puis compilez ce programme en lançant `cc -Wall -o maint maint.c` et listez le répertoire courant pour vous assurer que `maint` est bien présent et exécutable. Lancez alors `maint essai` puis `./maint essai`. Pourquoi seule la deuxième forme fonctionne ? Sachant que `.` (le caractère point) désigne le répertoire courant, que pouvez-vous faire pour que la première forme fonctionne aussi ? Vérifiez que vos appels à `maint` avec argument(s) affectent bien le fichier `maint.txt` au moins avec la deuxième forme.
- 8) Le programme `maint` affiche aussi le contenu de la variable d'environnement `MAVAR` si elle est définie, ou (null) dans le cas contraire. Donner deux manières différentes de faire écrire `MAVAR=hello` à `maint`.
- 9) Rendez le programme `maint` accessible et exécutable pour votre binôme. Assurez-vous qu'il arrive à se placer dans votre répertoire `fichiers-tp/maint` et à exécuter `maint` sans arguments. Puis faites un appel avec arguments. Que se passe-t'il ? Proposez deux solutions pour qu'il puisse lui aussi ajouter des lignes dans `maint.txt`. L'une de vos solutions est-elle plus avantageuse sur le plan de la sécurité ? Pourquoi ?
- 10) Remplacez-vous dans `fichiers-tp`, et éditez le fichier appelé `fichier.ext` en lançant `emacs` en arrière-plan sur lui : `emacs fichier.ext&` Quel fichier éditez-vous en réalité ? Quittez `emacs`, supprimez ce fichier (à l'aide du navigateur ou de la commande `rm`), et relancez à nouveau `emacs fichier.ext&`. Que constatez-vous ?
- 11) Ouvrez un autre terminal en arrière-plan (`xterm&`), et listez-y les fichiers avec `ls -l`. Revenez dans le premier terminal, et entrez la commande `ln acces.txt acces2.txt`, puis relistez à nouveau les fichiers (`ls -l`). Quelle différence voyez-vous sur `acces.txt` par rapport à l'ancien listing ?
- 12) Les fichiers `acces.txt` et `acces2.txt` sont-ils différents en taille et/ou en contenu ? Lancez `emacs` en arrière-plan sur `acces.txt`, puis modifiez ce fichier enregistrez-le. Que constatez-vous lorsque vous relistez les fichiers au niveau du terminal ? Que pourrait être, d'après vous, le premier

champ produit par la commande `ls -li` ?

- 13) Essayez à présent de lancer `ln acces.txt /var/tmp`. Comment interprétez-vous le message qui vous est renvoyé (indication : mount) ?
- 14) Appelée avec un seul argument, la commande `grep` filtre les lignes de l'entrée standard qui contiennent la chaîne de caractères passée en argument. Si l'argument est précédé de `-v`, le filtrage de `grep` est inversé. Appelée avec l'argument `-l`, la commande `wc` compte le nombre de lignes que comporte de l'entrée standard. Quel est l'effet de `ls -l | wc` ? De `ls -l | grep dr` ? De `ls -l | grep -v dr` ? De `ls -l | grep dr | wc -l` ? Quel est le rôle précis de l'opérateur `|` ?
- 15) Examinez le fichier `/etc/passwd` avec `emacs`. Comment pourriez-vous faire pour afficher ce fichier sans les lignes de commentaires ?
- 16) Quel est l'effet de `cat /etc/passwd | cut -d: -f1` ? De `cat /etc/passwd | cut -d: -f4,5` ?
- 17) Remplacez-vous dans `fichiers-tp`, et listez le répertoire avec `ls -l`, puis lancez `ls -l > sortie`. Que contient le fichier `sortie` ?
- 18) Entrez maintenant `ls -l /rien > sortie`. Que contient `sortie` cette fois-ci ? Donnez deux manières différentes de récupérer le message d'erreur affiché dans `sortie`.
- 19) Exécutez à nouveau `ls -l > sortie`. Y a-t-il une différence entre `cat sortie | wc -l` et `wc -l < sortie` au niveau du résultat ? Et au niveau système ?
- 20) En supposant que `FIC` est une variable d'environnement contenant un nom de fichier valide (`/etc/passwd` par exemple), donner la ligne de commande qui permet d'afficher le nombre de lignes que comporte ce fichier sous forme de message complet (par exemple : « le fichier `/etc/passwd` comporte xx lignes »).