

Chapitre 1

Logique séquentielle

1.1

Chapitre 2

Bash

2.1 Révisions

1. Quelle commande affiche le chemin du répertoire courant ?
2. Créer un répertoire `tmp` dans le répertoire courant.
3. Faire de `tmp` le répertoire courant.
4. Copier dans le répertoire courant le fichier `words` présent dans le répertoire `dictionaries-common` lui-même présent dans le répertoire `etc` sous la racine.
5. Renommer le fichier `words` en `dico_US`.
6. Afficher dans la console les 10 premières lignes de ce fichier.
7. Afficher dans la console le nombre de lignes du fichier.
8. Afficher les lignes du fichier qui contiennent *à la fois* les chaînes de caractères « `commu` » et « `nist` ».
9. Créer un fichier `to_delete` qui contient le résultat de(s) commande(s) de la question précédente.
10. Effacer le répertoire `tmp` et son contenu.

2.2 Substitutions

1. Quelle ligne de commande permet d'afficher la liste des fichiers du répertoire courant dont le nom ne comporte que 2 lettres ?
2. Quelle ligne de commande permet d'afficher la liste des fichiers du répertoire courant dont le nom contient au moins 2 m ?
3. On désire regrouper dans un répertoire `$HOME/tmp` les fichiers du répertoire courant dont le nom contient un caractère minuscule suivi d'un caractère majuscule. Quelle est la ligne de commande à saisir ?

2.3 Commande `cat` et redirection

On rappelle que la commande `cat` affiche sur la sortie standard ce qu'elle lit

sur l'entrée standard. Dans le cas où on fournit à cette commande un ensemble de paramètres correspondant à des noms de fichiers, la commande affiche le contenu de ces fichiers l'un après l'autre (elle réalise donc une concaténation de l'ensemble des fichiers). Expliquez le résultat suivant :

```
$ cat fic1
abc
$ cat fic2
def
$ cat fic1 fic2 > fic1
cat: fic1: le fichier d'entree est aussi celui de sortie
$ cat fic1
def
```

2.4 Redirections < et >

Lors d'une erreur la plupart des commandes Unix renvoie un message de la forme : *nom commande : message d'erreur*

Expliquer les deux derniers résultats renvoyés par les commandes suivantes.

```
$ cat fic
abcdefgh
$ (rm fic ; cat fic ; cat > fic) < fic
cat: fic aucun fichier ou repertoire de ce type
$ cat fic
abcdefgh
```

2.5 Redirection &>

L'utilisation de `&>` sur la ligne de commande indique au shell une redirection de la sortie standard et de la sortie erreur standard vers un seul et même fichier. Dans cet exercice, on supposera que le fichier `essai` n'existe pas.

Expliquer les résultats suivants :

```
$ wc < essai &> stat
bash: essai: Aucun fichier ou repertoire de ce type
$ cat stat
cat: stat: Aucun fichier ou repertoire de ce type

$ wc essai &> stat
$ cat stat
wc: essai: Aucun fichier ou repertoire de ce type
```

2.6 Variable

Soit `x` une variable du shell Bash.

1. Quelle est la commande permettant d'affecter le caractère `*` à la variable `x` ?
2. Commenter l'exécution de : `echo '$x' ; echo "$x" ; echo $x`

3. Soit le script `ecrire` défini de la façon suivante :

```
#!/bin/bash
echo $1
```

Commenter l'exécution de : `ecrire '$x' ; écrire "$x" ; écrire $x`

2.7 Variable

On considère le script `change.sh` dont le contenu est le suivant :

```
#!/bin/bash
unset $1
export $1=$2
```

Que renvoie sur la sortie standard la ligne de commande suivante :

```
export v=1 ; echo a $v ; ./change.sh v 2 ; echo b $v) ; ./change.sh
v 3 ; echo c $v ; bash ; ./change.sh v 4 ; echo d $v ; exit ; echo
e $v
```

2.8 Séquence de commandes

On considère le script `tln_cd` dont le contenu est le suivant :

```
#!/bin/bash
echo repertoire courant : 'pwd'
echo repertoire destination $1
cd $1
echo Contenu du repertoire :
ls
```

Vous vous trouvez dans le répertoire `/home` dans lequel se trouve deux sous-répertoires `partage` et `public`. Quel est le résultat de :

```
(tln_cd partage ; pwd) ; tln_cd .. ; tln_cd public ; pwd
```

2.9 Buffer Q

Soit le fichier `fic` composé de 26 lignes, chaque ligne contenant un mot de 1023 caractères ainsi que le caractère de fin de ligne.

On suppose que les zones mémoire utilisées pour les opérations de lecture-écriture sur le disque ont pour taille $8Kio$.

1. Combien de résultats différents peut renvoyer l'instruction ci-dessous (le caractère spécial `&` permet de lancer la commande en arrière-plan) ? Donner pour chacun d'entre eux les informations qui sont renvoyées à l'écran.

```
$ cat < fic > /tmp/f& wc < /tmp/f
```

2. Quelle différence y a-t-il avec la commande suivante :

```
$ cat < fic | wc
```

2.10 sed

Voici un extrait du résultat de la commande `env` :

```
PWD=/home/licinfo/jean
DOMAIN=univ-tln
```

Soit les commandes suivantes :

```
$ pays=fr
$ echo $PWD | sed "s/.*\/[a-z]*\)/\1@${DOMAIN}.$pays/"
```

Que renvoie la dernière commande ?

2.11 Script `tln_minus`

Ecrire le script `tln_minus` qui accepte un paramètre. Ce script transforme en minuscules le nom de tous les fichiers et répertoires contenus dans le répertoire passé en paramètre.

Exemple :

```
$ ls TEMPO
  Fic1 Essai BIN UE23 tp4.pdf
$ ./tln_minus TEMPO
$ ls TEMPO
  fic1 essai bin ue23 tp4.pdf
```

2.12 Script `tln_rev`

Ecrire le script `tln_rev` qui prend en paramètre une chaîne de caractères et renvoie sur la sortie standard cette chaîne écrite à l'envers. Remarque : l'option `-n` de la commande `echo` supprime le retour chariot.

```
$ ./tln_rev bonjour
ruojnob
```