

## FICHE 6 – Manuel

### 1. `man date`

#### a. `date`

#### b. `date +%A, %dth %B %Y` (cf. FORMAT dans man date)

#### c. `date -d '4/8/1976' +%a` (%A = jour en toutes lettres ; %a = jour abrégé)

La commande date accepte des formats très variés (ex : -d "next friday"). Remarquez cependant que le format britannique inverse le jour et le mois : 'month/day/year'.

(cf. DATE STRING dans man date)

#### d. `date -d '4/8/1976' +%j`

### 2. `man cal`

#### a. `cal 4 1976`

### 3. `man less`

flèches pour reculer/avancer ; PageUp-PageDown pour se déplacer page par page ; etc.

recherche par / : */string* puis 'n' (next) ou 'N' (previous) pour rechercher les occurrences d'un mot (attention, sensible à la casse: si minuscule → recherche minuscule et majuscules ; si majuscules → recherche uniquement majuscules)

## FICHE 7 – Globbing

1. `ls /etc -ls` (-S pour « sort » → tri décroissant selon la taille des fichiers  
≠ -s pour « size » → affiche la taille allouée à chaque fichier, en nombre de blocs)
2. `cd /proc`
3. `echo *`
4. `ls -d ????` (-d pour afficher les noms des répertoires eux-mêmes, pas leur contenu)  
(`echo ????` fonctionne aussi mais on préférera opter pour la commande `ls` qui colore les fichiers selon leur type et dispose de nombreuses options)
5. `ls -d [aeiouyAEIOUY]*`
6. `ls *info`
7. affiche « \* » : le globbing ne permet pas de remplacer le méta-caractère \* car le répertoire est vide → la commande « `echo *` » est exécutée par le shell et affiche la chaîne « \* »
8. (a) `find` trouve les 2 fichiers `s.sh`  
(b) `find` trouve les 3 fichiers d'extension `.sh`

`find . -name *.sh` → le globbing remplace `*.sh` par les fichiers qui correspondent dans le répertoire courant, c-à-d `s.sh` (seul développement possible dans le répertoire courant suite au globbing) ; la commande qui sera ensuite exécutée par le shell est donc :

```
find . -name s.sh
```

Le résultat est donc bien uniquement :

```
gstenuit@PC-maison:~/Test$ find . -name *.sh
./Subtest/s.sh
./s.sh
```

Il est alors évident alors que la commande qui fonctionne et trouve tous les fichiers se terminant par `.sh` depuis un répertoire donné est la suivante :

```
gstenuit@PC-maison:~/Test$ find . -name "*.sh"
./Subtest/s.sh
./Subtest/s2.sh
./s.sh
```

En effet, les guillemets désactivent le globbing sur la chaîne `*.sh`.

9. Après avoir créé un fichier supplémentaire dans le répertoire courant dont l'extension est aussi `.sh`, le globbing remplace donc la commande du point (a) par

```
find . -name *.sh → find . -name s.sh s3.sh
```

Il y aura donc une erreur puisque l'argument ou l'option `s3.sh` n'est pas reconnu dans la commande `find`. En encapsulant la chaîne `"*.sh"` avec des guillemets, la commande (b) exécute bien la demande.

```

gstenuit@PC-maison:~/Test$ touch s3.sh
gstenuit@PC-maison:~/Test$ cd ..
gstenuit@PC-maison:~$ tree Test/
Test/
├── Subtest
│   ├── s.sh
│   └── s2.sh
├── s.sh
└── s3.sh

1 directory, 4 files
gstenuit@PC-maison:~$ cd Test/
gstenuit@PC-maison:~/Test$ find . -name *.sh
find: paths must precede expression: `s3.sh'
find: possible unquoted pattern after predicate `-name'?
gstenuit@PC-maison:~/Test$ find . -name "*.sh"
./Subtest/s.sh
./Subtest/s2.sh
./s.sh
./s3.sh

```

10. `find /etc -name "*.conf"`

( $\Leftrightarrow$  `find /etc -name *.conf` ssi globbing `*.conf` non résolu dans rép. courant)

11. `find ~ -name "*.html" -writable` (-writable = droit d'écriture pour l'utilisateur)

12. `find ~/public_html -name "*.html" -type f -exec cat {} \;`

## FICHE 8 – Scripts base

1.

a. `nano date_a.sh`

```
#!/bin/bash
date          # date système
chmod +x date_a.sh
./date_a.sh
```

b. idem `date_b.sh` → `date +"%A, le %d %B %Y"`

c. idem `date_c.sh` → `date -d '4/8/1976' +%a`

d. idem `date_d.sh` → `date -d '4/8/1976' +%j`

2. `nano scriptA.sh`

`chmod +x scriptA.sh`

`cp scriptA.sh scriptB.sh`

`scriptA.sh`

→ « -bash: scriptA.sh : commande introuvable »

`./scriptA.sh`

→ « ./scriptA.sh »

`./scriptA.sh coucou tout le monde`

→ « ./scriptA.sh coucou tout »

`./scriptA.sh "coucou tout le monde"`

→ « ./scriptA.sh coucou tout le monde »

`./scriptA.sh "coucou tout" "le monde"`

→ « ./scriptA.sh coucou tout le monde »

`./scriptB.sh coucou tout le monde`

→ « ./scriptB.sh coucou tout »

3. `nano find.sh`

```
#!/bin/bash
find "$1" -name ".*$2" -type f
chmod +x find.sh
./find.sh
```