

# Bash

---

1. Tapez la commande `history` pour afficher la liste de vos dernières commandes.

```
history
# out
1  ping 8.8.8.8
2  su root
3  zsh
4  exit
5  history
```

2. En utilisant le caractère `!`, réexécutez certaines de ces commandes en utilisant soit leur numéro dans la liste, soit le début de la ligne correspondante.

```
# execute the first command in the history
!1
# execute the last command (useful with sudo !!)
!!
```

3. Utilisez la commande `which` pour voir où se trouve la commande `rm` que vous utilisez couramment.

```
which rm
# out
/bin/rm
```

4. Afin de sécuriser l'utilisation de la commande `rm`, créez manuellement un alias sur la commande `rm`. (alias `rm='rm -i'`)

```
# temporary
alias rm='rm -i'
# user
# ~/.bashrc
alias rm='rm -i'
```

5. Vérifiez que ce nouvel alias se trouve dans la liste de l'utilisateur `cpnv`.

```
touch test.txt
rm test.txt
# out
rm: remove regular empty file 'test.txt'?
```

6. Réutilisez la commande `which` pour voir quelle commande `rm` est à présent utilisée prioritairement.

```
which rm
# out
/bin/rm
```

7. Utilisez ensuite la commande `unalias` pour supprimer l'alias `rm`.

```
unalias rm
# test
touch test.txt
```

```
rm test.txt
# removes without confirmation
```

8. Pour créer des alias *permanents* (validés dans toute nouvelle session Shell), ouvrez le fichier `~/.bashrc` avec l'éditeur de texte nano.

i. Recopiez l'alias `rm` de la question précédente,

```
# ~/.bashrc
alias lr='ls -R'
```

ii. Ajoutez un alias appelé `lr` qui affichera la sous-arborescence (avec les infos détaillées) du répertoire courant.

```
# ~/.bashrc
alias lr='ls -R'
```

iii. Ajoutez un alias appelé `back` permettant de revenir au répertoire précédent (comme "`cd -`"). Pour cela, votre alias devra utiliser le contenu de la variable d'environnement `OLDPWD`.

```
# ~/.bashrc
alias back='cd $OLDPWD'
```

iv. Refermez le fichier `~/.bashrc` et validez les modifications en tapant la commande `source ~/.bashrc`.

v. Affichez la liste de vos alias et testez-les pour voir s'ils sont bien été validés.

```
alias
# out
alias back='cd $OLDPWD'
alias la='ls -lA'
alias ll='ls -l'
alias lr='ls -lR'
alias ls='ls --color=auto'
alias rm='rm -i'
```

9. Retrouvez comment afficher la liste des Variables d'Environnement.

```
env
```

10. Afficher la page d'aide de la commande `route` en différentes langues, redéfinissez la variable d'environnement `LANG` successivement à `fr` (`fr_CH.UTF-8`), en (`en_EN`), puis de (`de_DE`) et affichez à chaque fois l'aide de `route`.

```
# default lang : LANG=en_GB.UTF-8
# fr-ch
export LANG=fr_CH.UTF-8
man route
# out
"man: can't set the locale; make sure $LC_* and $LANG are correct"
# en-en
export LANG=en_EN
# out
"man: can't set the locale; make sure $LC_* and $LANG are correct"
# de_DE
export LANG=de_DE
# out
"man: can't set the locale; make sure $LC_* and $LANG are correct"
```

11. Nous souhaitons que `cpnv` puisse lancer le script `cherche-conf.sh` (créé dans TP2) sans devoir préciser le répertoire dans lequel il se trouve.

i. Essayez de lancer le script sans donner son chemin et vérifiez qu'il n'est pas trouvé.

```
cherche-conf.sh
# out
"bash: cherche-conf.sh: command not found"
```

ii. Avec la commande `echo $PATH`, observez la valeur du PATH de l'utilisateur courant.

```
echo $PATH
# out
"/home/mon/.autojump/bin:/usr/local/bin:/usr/bin:/bin:/usr/games"
```

iii. Utilisez cette information pour créer le répertoire manquant dans votre répertoire personnel et placez-y le script (tout cela sans utiliser les droits de root).

```
mkdir ~/scripts
cp ~/TP2/test/cherche-conf.sh ~/scripts
export PATH=$PATH:~/scripts
```

iv. Utilisez la commande `which` pour vérifier que le script est bien trouvé par le shell bash et lancez-le sans en préciser le chemin.

```
which cherche-conf.sh
# out
/home/mon/scripts/cherche-conf.sh
```

12. Le répertoire `/sbin` ne figure pas dans le PATH de l'utilisateur `cpnv` et nous allons le rajouter pour avoir accès à certaines commandes système (même si cela n'est pas recommandé !)

i. Tapez la commande ~~`ip addr`~~ `shutdown` pour vérifier qu'elle n'est pas trouvée avec le PATH actuel puis affichez le contenu de votre PATH pour vérifier que le répertoire `/sbin` n'y figure pas.

ii. Ajouter manuellement le répertoire `/sbin` au contenu de votre PATH et puis affichez-le pour vérifier que cela a bien marché.

```
export PATH=$PATH:/sbin
echo $PATH
# out
/home/mon/.autojump/bin:/usr/local/bin:/usr/bin:/bin:/usr/games:/home/mon/scripts:/sbin
```

iii. Pour que cette modification soit ensuite permanente, ajoutez-la dans le fichier `~/.bashrc` comme pour la création des alias puis validez le changement.

```
vim ~/.bashrc
# ~/.bashrc
...
export PATH=$PATH:/sbin

# With append
echo "export PATH=$PATH:/sbin" >> ~/.bashrc
```

iv. Rouvrez une nouvelle session, vérifiez le contenu de votre PATH puis tapez la commande ~~`ip addr`~~ `sudo shutdown` qui devrait à présent pouvoir être lancée sans en préciser le chemin.

13. Tapez la commande `uname -r` qui vous affichera le numéro de version actuelle de votre noyau.

```
uname -r
4.19.0-16-amd64
```

14. Affichez le contenu du répertoire `/lib/modules` pour vérifier qu'il contient un répertoire par version du noyau précédemment installée.

```
ls /lib/modules
# out
4.19.0-16-amd64 4.9.0-13-amd64 4.9.0-15-amd64
```

15. A présent, faites en sorte d'afficher, en une seule commande et sans utiliser de variable intermédiaire, la liste des fichiers du répertoire `/lib/modules/ma_version_du_noyau` où `ma_version_du_noyau` est donnée par l'exécution de la commande `uname`.

```
find /lib/modules/ | grep `uname -r`
```