

# Windows managers

---

1. Afin d'observer l'enchaînement des commandes sur une même ligne, tapez `xterm ; xclock` .

2. Pourquoi le programme xclock ne s'est-il pas lancé et à quel moment le fera-t-il ?

i. Les programmes se lancent de manière linéaire, bash attend la fin de l'exécution de xterm pour lancer xclock

3. La commande kill permet de tuer un processus en lui envoyant un signal.

i. Tapez `man 7 signal` pour afficher la liste des signaux disponibles pour kill

```
man 7 signal
# out
: '
Term Terminate process
Ign Ignore signal
Core Terminate process and dump core
Stop Stop process
Cont Continue stopped process
'
```

ii. Retrouvez le numéro du signal de type KILL qui est le plus efficace pour tuer un processus.

a. Terminate, `kill(2)` donc 2?

4. Les commandes pgrep et pkill permettent de gérer les processus par leur nom.

i. Lancez deux commandes xeyes en tâches de fond.

```
xeyes &
xeyes &
```

ii. Utilisez les options de pgrep pour récupérer les numéros et les noms de ces 2 processus.

```
pgrep -l xeyes
```

iii. Utilisez la commande pkill pour tuer uniquement le plus vieux des deux.

```
pkill -o xeyes
```

5. Lancez xterm

i. Connectez-vous ensuite dans la première Console Virtuelle par la séquence Ctrl-Alt-F1 puis repérez et tuez le processus xterm qui tourne votre environnement graphique.

```
pgrep -l xterm
# 16379 xterm
killall xterm
```

ii. Retournez ensuite dans votre environnement graphique en tapant Ctrl-Alt-F7.

6. Installer le paquet psmisc

```
sudo apt install psmisc
```

7. Ce paquet contient divers utilitaires qui utilisent le système de fichiers proc

- i. `fuser` : identifie les processus utilisant des fichiers ou sockets.
- ii. `killall` : tue les processus par leur nom (ex « `killall -HUP named` »).
- iii. `peekfd` : affiche les données passant par un descripteur de fichier.
- iv. `pstree` : affiche un arbre des processus actifs.
- v. `prstat` : imprime le contenu de `/proc//stat`

8. Lancez une commande `xterm` puis placez-vous dans cette nouvelle fenêtre et lancer une nouvelle commande `xclock`.

- i. Avec la commande `ps alx` et le contenu de la colonne PPID, retrouvez le numéro du processus parent de votre nouvelle fenêtre `xterm`.

```
# use / to search for xclock
ps alx | less
# F  UID    PID   PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
# 0  1000  16713  16664  25   5  33996  6564 x64_sy SN   pts/1      0:00 xclock
```

- ii. Retrouvez ce lien de parenté à l'aide de la commande `pstree`.

```
pstree 16664
#zsh---xclock
```

- iii. Tuez le processus parent. Que se passe-t-il ?

```
pkill -P 16664
# kill la shell mais le terminal reste present et relance une session
```

9. Lancez 3 fois le programme `xclock` en tâche de fond.

```
for i in {1..3}; do xclock &; done
```

- i. Tuez les programme `xclock` en une seule fois avec la commande `killall`.
  - a. `killall xclock`
- ii. Quel est l'inconvénient de cette commande ?
  - a. tue tous les process du nom donné

10. Lancez `xterm` en tâche de fond et retrouvez son numéro de processus.

```
xterm&
psl alx | grep xterm
# F  UID    PID   PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
# 0  1000  17190  17138  25   5  39316 10320 core_s SN   pts/3      0:00 xterm
```

11. Déplacez-vous ensuite dans le répertoire `/proc` puis dans celui correspondant au numéro de processus trouvé.

```
cd /proc/17190
```

- ii. Observez le contenu des fichiers `status` et `environ`.

```
ls
: '
attr          cwd          map_files    oom_adj       schedstat     syscall
autogroup     environ     maps         oom_score     sessionid     task
auxv          exe         mem          oom_score_adj setgroups     timers
cgroup        fd          mountinfo    pagemap       smaps         timerslack_ns
clear_refs    fdinfo     mounts       patch_state    smaps_rollup  uid_map
cmdline       gid_map    mountstats   personality    stack         wchan
comm          io         net          projid_map     stat
```

```
coredump_filter  limits    ns          root        statm
cpuset           loginuid  numa_maps   sched        status
,
```

12. Lancez la commande top pour voir fonctionner l'ordonnanceur des tâches du système.

- i. top ou htop pour un gestionnaire de tâche plus agréable

13. Affichez seulement ceux dont l'utilisateur cpnv est le propriétaire.

- i. top -u mon

14. Vérifiez l'espace RAM disponible par la commande free.

```
free
#                total        used        free      shared  buff/cache   available
# Mem:           2018216      406264      652460        6712      959492      1417552
# Swap:           498684         7768       490916

# df donne plus d'informations pour l'espace libre
df
```

15. Obtenez les informations sur votre processeur en affichant le contenu du fichier /proc/cpuinfo.

```
less /proc/cpuinfo
```

16. Dans la liste des processus de votre système, vérifiez que le serveur ssh tourne bien. Pour cela, utilisez une option de pgrep permettant de chercher le mot sshd sur toute la longueur de la ligne dans la liste des processus et une autre affichant les détails sur le processus.

```
pgrep -l sshd
# 597 sshd
```

17. Dans le répertoire /etc/init.d, observez le contenu du script ssh permettant de lancer ce serveur et notamment les options pour l'arrêter ou le démarrer.

```
cat /etc/init.d/ssh
# start
# stop
```

18. A l'aide de ces options, arrêtez le serveur puis redémarrez-le en vérifiant que cela a marché par la liste des processus actifs ou en tentant d'y accéder par le client ssh.

```
sudo systemctl stop sshd
sudo systemctl status sshd
# inactive (dead)
sudo systemctl start sshd
sudo systemctl status sshd
# active (running)
```

19. Utilisez ensuite la commande service pour redémarrer le serveur ssh.

- i. sudo service sshd restart

20. Installer le paquet chkconfig

- i. sudo apt install chkconfig => unavailable or obsolete => use sysv-rc-conf

21. A l'aide de la commande `chkconfig`, vérifiez à quels niveaux de démarrage (RunLevel) `ssh` peut être démarré.

22. Installer le paquet `sysv-rc-conf`

i. Paquet déjà installé lors d'un exercice précédent

23. A l'aide de la commande `sysv-rc-conf`, vérifiez à quels niveaux de démarrage (RunLevel) `cron` peut être démarré.

i. `sudo sysv-rc-conf => entre 2 et 5`

24. Utilisez la commande `runlevel` pour vérifier le niveau d'exécution auquel vous travaillez.

i. `/sbin/runlevel => N 5`

25. Passez en mode de maintenance mono-utilisateur (Runlevel 1).

i. `sudo /sbin/init 1`

ii. `runlevel => 5 1`

26. Dans la liste des processus de votre système, vérifiez le statut des serveurs `ssh` et `cron`.

i. `systemctl status sshd => inactive (dead)`

ii. `systemctl status cron => inactive (dead)`

27. Installer le paquet `gnome-core` et `gdm3` et supprimer `xdm`

i. `sudo apt install gnome-core gdm3`

ii. `sudo apt --purge remove xdm => désinstalle xdm et vide les fichiers de configuration liés`

28. Installer `XRDP` (serveur RDP) pour se connecter depuis un client RDP® sur un serveur Linux

i. `sudo apt install xrdp`

29. Editez le fichier `/etc/xrdp/startwm.sh`

```
#!/bin/sh
if [ -r /etc/default/locale ]; then
    . /etc/default/locale
    export LANG LANGUAGE
fi
#. /etc/X11/Xsession
/usr/bin/openbox-session
```