



TP LINUX

Système de gestion de fichiers



À l'aide de la fiche de synthèse répondez aux questions suivantes. Pour ce faire nous utiliserons l'environnement de simulation Linux « Xlinux », intégré au navigateur Mozilla Firefox.

Reportez-vous à la vidéo « Installer_Xlinux_sur_Firefox.mp4 » pour installer l'extension « Xlinux ».

```

atal: PATA max PIO4 mmio cmd 0x9e000000 ctl 0x9e000100 irq 15
libphy: ethoc-mdio: probed
mousedev: PS/2 mouse device common for all mice
input: 94000000.keyboard as /devices/platform/94000000.keyboard/input/input0
input: ts-lpc32xx as /devices/platform/93000000.tsc/input/input1
rtc-lpc32xx 99000000.rtc: rtc core: registered rtc-lpc32xx as rtc0
NET: Registered protocol family 17
9pnet: Installing 9P2000 support
rtc-lpc32xx 99000000.rtc: setting system clock to 2021-11-11 15:31:42 UTC (1636644702)
ALSA device list:
#0: Dummy 1
ata1.00: ATA-2: jorlk-disk, , max PIO2
ata1.00: 128 sectors, multi 0: LBA
ata1.00: configured for PIO
scsi 0:0:0:0: Direct-Access      ATA          jorlk-disk      n/a  PQ: 0 ANSI: 5
sd 0:0:0:0: [sda] 128 512-byte logical blocks: (65.5 kB/64.0 KiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sd 0:0:0:0: [sda] Attached SCSI disk
VFS: Mounted root (9p filesystem) readonly on device 0:12.
devtmpfs: mounted
Freeing unused kernel memory: 136K (c054e000 - c0570000)
udhcpd: started, v1.26.2
Setting IP address 0.0.0.0 on eth0
udhcpd: sending discover
udhcpd: sending select for 10.5.34.0
udhcpd: lease of 10.5.34.0 obtained, lease time 900
Setting IP address 10.5.34.0 on eth0
Deleting routers
route: SIOCDELRT: No such process
Adding router 10.5.0.1
Recreating /etc/resolv.conf
Adding DNS server 10.5.0.1
  
```

Le terminal est une porte d'entrée et de sortie entre l'utilisateur et l'ordinateur, il permet de naviguer au sein du **SGF** (système de gestion de fichier), d'exécuter des commandes et même de communiquer avec les programmes en cours d'exécution.

En cas de doute d'une commande, utilisez le manuel en ligne :

<http://www.linux-france.org/article/man-fr/>

Travail préliminaire :

Visionnez la vidéo d'introduction « Le terminal, à quoi ça sert ?! » disponible sur la plateforme YouTube à l'adresse <https://www.youtube.com/watch?v=U2b-MYcSCLc>

Exercice 1 : Répertoire courant et chemin absolu

Par défaut le terminal s'ouvre dans le répertoire utilisateur. Utilisez la commande **pwd** pour déterminer son nom et son chemin absolu.

Nom du répertoire courant :	user
Chemin absolu :	/home/user

Exercice 2 : Listez les fichiers d'un répertoire

Utilisez la commande 'man ls' pour ouvrir le manuel de la commande ls ou le lien ci-dessous <http://www.linux-france.org/article/man-fr/man1/ls-1.html>

2.1 Quelle option permet de lister les fichiers et répertoires en donnant leur descriptif complet ?

A compléter

2.2 Quelle option permet d'afficher les fichiers et répertoires commençant par '.' ?

A compléter

2.3 Que sont les répertoires '.' Et '..' ?

A compléter

2.4 Tapez '**ls /var**' pour lister les fichiers et répertoires du répertoires 'var' se situant à la racine du système de fichier.

A compléter

Taper '**ls /var/run**'.

A compléter

2.5 Maintenant tapez '**ls -R /var**'. Expliquez ce que semble avoir fait le paramètre '-R' :

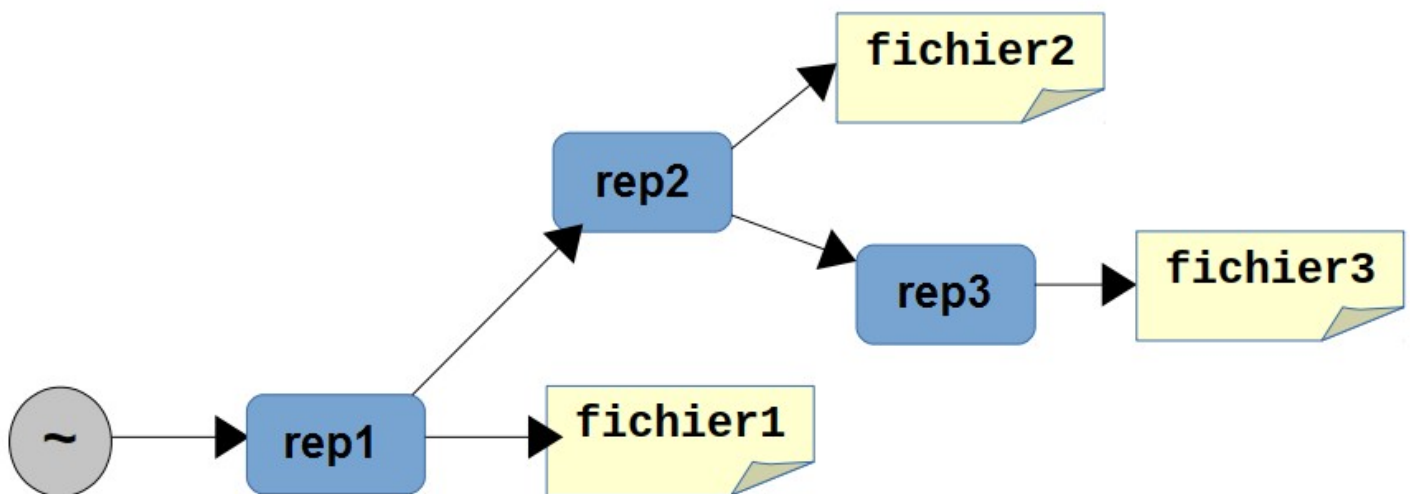
```
~ $ ls -R /var
```

```
/var:  
compilepackages empty run toppler  
/var/compilepackages:  
bc-1.06.95.tar.bz2 frotz-2.43d.tar.gz  
cmatrix-1.2a.tar.gz lua-5.2.3.tar.gz  
  
/var/empty:  
  
/var/run:  
ifstate inetd.pid udhcpc.eth0.pid  
  
/var/toppler:  
toppler.hsc
```

A compléter

Exercice 3 : Construction d'une arborescence

On envisage de construire l'arborescence suivante :



3.1 A quel répertoire se réfère ~ ? (Voir lexique du cours)

A compléter

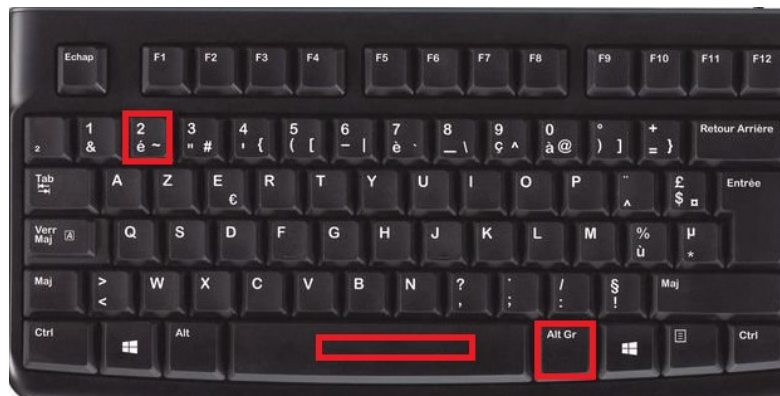
3.2 Création de répertoires et de fichiers (arborescence ci-dessus)

- La commande 'touch {nom_du_fichier}' permet de créer un fichier dans le répertoire courant ;
- La commande 'mkdir {nom_du_répertoire}' crée un répertoire dans le répertoire courant ;
- La commande 'cd {nom_du_répertoire}' permet de s'y déplacer, d'en faire le nouveau répertoire courant.

Quelles sont les commandes successives à rentrer pour créer l'arborescence ci-dessus après 'mkdir rep1' et 'cd rep1' ?

```
mkdir rep1
cd rep1
A compléter
```

3.3 Tapez la commande 'cd ~' (Alt Gr+é puis espace pour '~') pour revenir au répertoire utilisateur.



3.4 Quel est le chemin relatif et absolu de fichier2 ?

Chemin relatif	Chemin absolu
A compléter	A compléter

Exercice 4 : Gestion des droits sur un fichier

Dans le répertoire utilisateur '~' :

- tapez la commande '**chmod -R 0755 rep1**'
- lister les fichiers et répertoires à l'aide de la commande '**ls -l**'

Les droits du dossier **rep1** sont à présent à 755(rwxr-xr-x).

Quelles sont les commandes (en notation symbolique et en base 8) pour lui donner les droits suivants (on suppose qu'après chaque commande on remet le répertoire à 755) :

Par défaut	<code>drwxr-xr-x</code>
Commande 1	<code>drw-r-x--x</code>
Commande 2	<code>d--xr-x---</code>
Commande 3	<code>d-w-rwxr--</code>
Commande 4	<code>dr-x-w--wx</code>

	<i>Propriétaire (u : user)</i>			<i>Groupe (g : group)</i>			<i>Autres (o : other)</i>		
	Lecture	Écriture	Accès	Lecture	Écriture	Accès	Lecture	Écriture	Accès
défaut	oui	oui	oui	oui	non	oui	oui	non	oui
Commande 1	oui	oui	non	oui	non	oui	non	non	oui
Commande 2	non	non	oui	oui	non	oui	non	non	non
Commande 3	non	oui	non	oui	oui	oui	oui	non	non
Commande 4	oui	non	oui	non	oui	non	non	oui	oui

Complétez le tableau ci-dessous :

	Notation symbolique (relatif)	Notation en base 8 (absolu)
Commande 1	chmod u-x,o-r rep1	chmod 651 rep1
Commande 2	A compléter	A compléter
Commande 3	A compléter	A compléter
Commande 4	A compléter	A compléter

Exercice 5 : Les processus en cours d'exécution

Un processus est un programme en cours d'exécution.

Utilisez la commande 'ps' pour lister les processus,

- La première colonne indique l'identifiant du processus, noté PID (Processus Identifier),
- La deuxième colonne indique à quel utilisateur appartient le processus,
- La dernière colonne indique le nom du processus.

5.1 Quels sont les noms des deux processus de l'utilisateur « user » (c'est vous!) en cours d'exécution ?

```
80 user 0:00 -sh
188 user 0:00 ps
```

Attention, cela peut être différent pour vous

A compléter

5.2 Le processus Bash (-sh) correspond au terminal actuel. Utilisez la commande '**ps -o pid,ppid,user,time,args**', et psidentifiez les numéros de PID et PPID de vos processus.

```
80 1 user 0:00 -sh
205 80 user 0:00 ps -o pid,ppid,user,time,args
```

Processus	PID	PPID
-sh	A compléter	A compléter
ps -o pid,ppid,user,time,args	A compléter	A compléter

5.3 Que remarquez-vous ? Que pouvez-vous en conclure ?

A compléter

5.4 Affiner votre conclusion précédente avec le résultat de la commande suivante : **ps -o pid,ppid,user,time,args**

```
~ $ ps -o pid,ppid,user,time,args
init(1) +-inetd(82)
    | -sh(81)
    | -sh(80) ---pstree(208)
    ` -udhcpd(76)
```

Visualisation de l'activité du processeur

La commande 'htop' permet de visualiser l'activité du processeur en temps réel. Attendez environ 10 secondes pour obtenir la visualisation ci-dessous.

```

CPU[|||||] 8.3% Tasks: 6, 0 thr; 1 running
Mem[|||||] 4/25MB Load average: 0.19 0.09 0.07
Swp[ ] 0/0MB Uptime: 01:42:53

  PID USER   PRI  NI  VIRT   RES   SHR  S CPU% MEM%   TIME+  Command
  227 user    20    0  1496    960   848  R   2.7   3.7   0:02.26 htop
    1 root     20    0   1528    312   288  S   0.0   1.2   0:00.56 init
   76 root     20    0   1528    456   432  S   0.0   1.8   0:00.22 udhcpc -R -n -p /
   80 user     20    0   1544    584   544  S   0.0   2.3   0:00.86 -sh
   81 root     20    0   1536    544   512  S   0.0   2.1   0:00.02 -sh
   82 root     20    0   1528    448   424  S   0.0   1.7   0:00.02 /usr/sbin/inetd

F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit

```


Il est possible d'utiliser les touches fonctions (F1, F2, etc...) pour configurer l'outil.

F1 : aide (help) ; **F2** : configurer (setup) ; **F4** : filtrer (filter) ; **F6** : trier (sort by) ; **F10/q** : quitter (quit)

5.5 Dans quel ordre sont listés les différents processus par défaut ? (Appuyer sur F6)

A compléter

Exercice 6 : Exécution et arrêt d'un processus

La flèche en dessous du terminal Xlinux  permet d'importer un fichier. **Importez** le fichier script.sh qui vous est fourni.



6.1 Essayez d'exécuter le fichier en tapant './script.sh', que se passe-t-il ? Quelle commande permet de remédier à ce problème ?

A compléter

6.2 Une fois le problème réglé, **exécutez** le fichier 'script.sh' en faisant **./script.sh** celui-ci « tourne » dans le vide et vous empêche de lancer d'autres commandes. **Utilisez** les touches 'Ctrl+C' du clavier pour stopper l'exécution.

Le symbole '&' dans une commande permet une exécution en fond de tâche.

6.3 Entrez la commande './script.sh &'.

Le shell vous redonne la main sur la console de terminal, vous pouvez donc taper d'autres commandes pendant que le script tourne « derrière ».

6.4 Utilisez les commandes suivantes :

- **ps -o pid,ppid,user,time,args**
- **pstree -p**

Retracez l'arborescence des processus depuis le processus « script.sh », jusqu'au processus « init » ayant pour PID 1

```
~ $ pstree -p
init(1)---inetd(82)
    |
    |--sh(81)
    |--sh(80)---pstree(264)
    |           `--script.sh(250)
    `--udhcpd(76)
```

A compléter et insérer une copie d'écran

6.5 Stopper un processus

La commande 'kill {signal} {PID}' permet d'envoyer un signal (défini par un numéro) à un processus (défini par son PID). Le signal numéroté 9 permet de stopper n'importe quel processus quel qu'il soit, il est donc normalement réservé à l'administrateur.

Utilisez la commande '**kill 9 {PID_de_script.sh}**' pour stopper le script puis la commande 'ps' pour le voir absent de la liste.

Dans cet exemple le PID de script.sh est 127 :

```
$ kill 9 127
-bash: kill: (9) - Operation not permitted
$
```

Vous aurez ce message d'erreur, pas d'inquiétude, l'environnement WebLinux est particulier, la commande se fera quand-même.

Exercice 7: Ecriture d'un script et processus orphelin

7.1 Nous allons maintenant écrire notre propre script ; tapez la commande **nano TP.sh** pour créer le fichier 'TP.sh'.

Utilisez la commande 'nano TP.sh' pour éditer votre fichier avec l'outil 'nano'.

```
GNU nano 2.2.6 File: tp.sh
[ Read 3 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Tapez en première ligne **#!/bin/sh**

Cette ligne indique au terminal qu'il doit interpréter le reste du fichier en tant que commande bash.

Vous pouvez donc écrire toutes les commandes que vous désirez. Lorsque vous exécuterez le fichier, l'ordinateur l'interprétera comme si vous tapiez les commandes dans le terminal.

7.2 Faites en sorte que votre fichier exécute la commande './script.sh' en l'écrivant dans l'éditeur nano comme le montre la copie d'écran ci-dessous.

```
GNU nano 2.2.6 File: TP.sh Modified
#!/bin/sh
./script.sh
[ Read 2 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Pour enregistrer le fichier **TP.sh** tapez **Ctrl+o** (enregistrer), **touche entrée** puis **Ctrl+x** (sortir)

7.3 Modifiez les droits d'exécution sur le fichier Tp.sh. Précisez la commande utilisée (voir question 6.1)

7.4 Exécutez votre fichier en fond de tâche './TP.sh &', puis visualisez les PID et PPID des processus.

```
~ $ pstree -p
init(1)---inetd(82)
    |
    |--sh(81)
    |--sh(80)---TP.sh(313)---script.sh(314)
    |
    |--pstree(316)
    |--udhcpc(76)

~ $ ps -o pid,ppid,user,args | grep script
314  313 user      {script.sh} /bin/sh ./script.sh
318   80 user      grep script
```

7.5 Maintenant tapez '**kill 9 {PID_de_TP.sh}**' sur le processus 'TP.sh' (voir question 6.5) et visualisez de nouveau le PPID du processus script.sh '**ps -l**'.

```
~ $ kill 9 313
-sh: can't kill pid 9: Operation not permitted
~ $ ps -o pid,ppid,user,args | grep script
 314      1 user      {script.sh} /bin/sh ./script.sh
 320     80 user      grep script
[1]+  Terminated                  ./TP.sh
```

Que remarquez-vous ? Que s'est-il passé ? (Voir fin du cours sur les processus et/ou le site <https://www.it-connect.fr/les-processus-zombies/>)

A compléter et insérer une copie d'écran