

Cours :

IFT-3830 : A-2023

Chargé de cours :

Daniel Ouimet

Travail :

T.P.-2

Date remise :

Vendredi le 17 novembre 2023

Équipe:

Gueorgui Poklitar

Ion Hincu

1.

a.

Les CD seront montés sur « **/media/dvd** ». Les options de montage sont **noauto**, **users** et **ro**.

b.

/dev/sdb2 sera monté sur « **/export/home** ».

/dev/sdb3 sera monté sur « **/var** ».

c.

Il s'agirait d'une erreur puisque « **/dev/hdb4** » est déjà monté sur « **/export/web** », donc il peut y avoir des problèmes de cohérence et d'intégrité de donnée. Également le répertoire « **/usr/local/web** » pourrait ne pas être équiper avec l'option « **ro** » comme « **/export/web** ».

d.

La mémoire virtuelle dans ce cas désigné « swap », se trouve dans le répertoire de « **/dev/hdb1** ». De ce fait, « **hd** » est associé avec l'interface « **IDE** », le « **b** » dans **hdb1** signifie une association avec disque « **B** » et le « **1** » veut dire la première partition. Donc, la mémoire virtuelle se trouve dans la première partition du disque b de l'interface IDE.

e.

Pour l'ordre de priorité il faut se référer au dernier chiffre du tableau « **/etc/fstab** ». Le chiffre 0 signifie qu'aucune vérification est exécuté automatiquement par fsck. Par la suite, le **1** est vérifier en priorité et souvent associées au filesystem du root qui sont plus importants. Puis le **2** qui est une priorité moins importante. Si le même chiffre se répète, cela signifie que ces partitions peuvent être vérifié en parallèle.

Nous avons 3 partitions vérifiées en priorité avec l'ordre 1 de façon parallèle :

- **/dev/sda2**
- **/dev/sda1**
- **/dev/sdb2**

Puis nous avons 4 partitions vérifiées en seconde priorité avec l'ordre 2 de façon parallèle :

- **/dev/hdb4**
- **/dev/hdb2**
- **/dev/sda6**
- **/dev/sdb3**

2.

a. Usagers :

1)

On utilise la commande « **useradd -u 3000 rouge** » pour créer l'utilisateur rouge avec l'UID 3000.

```
[root@localhost ~]# useradd -u 3000 rouge
[root@localhost ~]# _
```

Et « **useradd -u 3001 bleu** » pour créer l'utilisateur bleu.

```
[root@localhost ~]# useradd -u 3001 bleu
```

Avec la commande « **id rouge** » on va voir que l'utilisateur existe ainsi que son UID, GID et son group.

```
[root@localhost ~]# id rouge
uid=3000(rouge) gid=3000(rouge) groups=3000(rouge)
[root@localhost ~]#
```

Et « **id bleu** » pour l'utilisateur bleu.

```
[root@localhost ~]# id bleu
uid=3001(bleu) gid=3001(bleu) groups=3001(bleu)
```

Pour assigner un mot de passe on utilise la commande « **passwd rouge** » pour l'utilisateur rouge et « **passwd bleu** » pour l'utilisateur bleu.

```
[root@localhost ~]# passwd rouge
Changing password for user rouge.
New password: _
[root@localhost ~]# passwd rouge
Changing password for user rouge.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

2)

Pour donner la permission « **sudo** » à l'utilisateur **rouge** on va éditer le fichier « **sudoers** » avec la commande « **visudo** ».

rouge ALL = NOPASSWD : /usr/bin/vi

```
rouge    ALL=NOPASSWD:/usr/bin/vi
```

Avec la commande **sudo -l -U rouge** on va vérifier les permissions d'utilisateur **rouge**.

```
User rouge may run the following commands on localhost:
 (root) NOPASSWD: /usr/bin/vi
[root@localhost ~]#
```

Il y a effectivement un problème de sécurité, car permettre un accès **sudo** sans mot de passe à l'éditeur vi est risqué. L'éditeur vi a la capacité de modifier n'importe quel fichier, y compris les fichiers système. Cette permission pourrait mener à une escalade de privilèges ou compromettre la sécurité du système.

3)

Pour donner la permission « **sudo** » à l'utilisateur **bleu** on va éditer le fichier « **sudoers** » avec la commande « **visudo** ».

bleu localhost =/bin/l

```
bleu localhost=/bin/l
```

On va vérifier les permissions d'utilisateur **bleu** avec la commande.

```
User bleu may run the following commands on localhost:
(root) /bin/l
[root@localhost ~]# _
```

Il est important de noter que même si cette permission semble inoffensive car elle est restreinte à la commande **ls**, il est toujours judicieux de limiter les permissions **sudo** autant que possible pour des raisons de sécurité. En l'occurrence, la restriction à **localhost** est une bonne pratique de sécurité, car elle empêche l'exécution de la commande depuis d'autres hôtes.

4)

Il faut l'ajouter au répertoire « **/etc/profile.d** ». Ce répertoire contient des scripts qui sont exécutés additionnellement à la connexion à une session utilisateur, pour tous les utilisateurs.

De ce fait, on doit y créer un script **.sh** puis y inclure la ligne suivante :

export PATH="\$PATH:/usr/lib/debug/bin"

Donc pour tous les utilisateurs du système, le lien sera placé à la fin du PATH au moment de leur connexion.

5)

Avec la commande **vipw -s** on ajoute le signe (!) avant le mot de passe encrypté de l'utilisateur rouge.

```
rouge: !$6$1qrEb1SHp3zi3gzus$btQ11BpIR8/aQEchoETS7h/NSWPAX0/uF2m3Lsi57j0ADp9/2N7BXTpR5JwExiZsxDUYEQs
r3kSQBdpkAbn0:19656:0:99999:7:::
:wq! _
```

```
localhost login: rouge
Password:
Login incorrect
localhost login: _
```


2)

```
root@localhost ~# systemctl enable ypsero.service
Created symlink /etc/systemd/system/multi-user.target.wants/ypserv.service → /usr/lib/systemd/system/ypserv.service.
```

```
root@localhost ~# systemctl enable ypbind.service
Created symlink /etc/systemd/system/multi-user.target.wants/ypbind.service → /usr/lib/systemd/system/ypbind.service.
```

3)

```
NISDOMAIN=DOMAIN1
```

"sysconfig/network" 2L, 40C written

Redémarrez pour que le nom de domaine NIS prenne effet.

La commande « **systemctl restart ypbind** » nous permet de redémarrer et pour vérifier si on a bien changé le nom on utilise la commande « **domainname** ».

```
[root@localhost ~]# domainname  
DOMAIN1  
[root@localhost ~]#
```

4)

Pour compiler la base de données NIS, on va utiliser la commande « **make** » dans le répertoire « **/var/yp** ».

```
[root@localhost ~]# cd /var/yp  
[root@localhost yp]# make  
gmake[1]: Entering directory '/var/yp/DOMAIN1'  
Updating passwd.byname...  
Updating passwd.byuid...  
Updating group.byname...  
Updating group.bygid...  
Updating hosts.byname...  
Updating hosts.byaddr...  
Updating rpc.byname...  
Updating rpc.bynumber...  
Updating services.byname...  
Updating services.byservicename...  
Updating netid.byname...  
Updating protocols.bynumber...  
Updating protocols.byname...  
Updating mail.aliases...  
gmake[1]: Leaving directory '/var/yp/DOMAIN1'  
[root@localhost yp]# _
```

5)

Dans le fichier on a ajouté la ligne « **ypserver localhost** ».

```
# Valid entries are  
#  
# domain NISDOMAIN server HOSTNAME  
#     Use server HOSTNAME for the domain NISDOMAIN.  
#  
# domain NISDOMAIN broadcast  
#     Use broadcast on the local net for domain NISDOMAIN  
#  
# ypserver HOSTNAME  
#     Use server HOSTNAME for the local domain. The  
#     IP-address of server must be listed in /etc/hosts.  
#  
# broadcast  
#     If no server for the default domain is specified or  
#     none of them is reachable, try a broadcast call to  
#     find a server.  
#  
ypserver localhost
```

Pour que les changements s'appliquent on doit redémarrer le service NIS avec la commande « **systemctl restart ypbind** ».

6)

La commande « **ypcat passwd** » nous affiche les informations du fichier **passwd** qui provient du service NIS, dans notre cas les clients **bleu, guest, et rouge**.

```
[root@localhost ~]# ypcat passwd
bleu:$6$C1ezPthcQrn14sKH$Kqe96pVhPH8krR9YWMaPQx9Nss icanEGdTd91K0QH1Eqec2NP8fpsD.H6CUaBIuRgBAkZFdU0h
yf0H0gMt841:3001:3001::/home/bleu:/bin/bash
guest:$6$upF1UqWgKQ25tnrU$UgHoFvTPoa5SJaaw0pz i0yabBuNBqUzYM9oTbDUc1hGdpZ.a5A jM1s7qfbI91Wg IHS6ZDWtF2
2Bp7srPAQnA0:1000:1000:guest:/home/guest:/bin/bash
rouge:$6$JQJa18wb/6Sdb.jnG$427BNd14QhCSrFdJYIEtAob7MS0w06ZM1MD0IzPMtu44JwRDuQbLcM113N8he7IH/4BxTtrOuD
L3Suz.OEQJE.:3000:3000::/home/rouge:/bin/bash
[root@localhost ~]#
```

c) Imprimantes :

1)

Pour créer l'imprimante on va utiliser la commande :

lpadmin -p imp2023 -E -v file:///dev/null -m raw

```
[root@localhost ~]# lpadmin -p imp2023 -E -v file:///dev/null -m raw
[root@localhost ~]# lpadmin -d imp2023
```

Pour définir l'imprimante par default on va utiliser la commande :

lpadmin -d imp2023

```
[root@localhost ~]# lpadmin -d imp2023
```

Pour s'assurer que l'imprimante a bien été ajouté on utilise la commande :

lpstat -p -d

```
[root@localhost ~]# lpstat -p -d
printer imp2023 is idle. enabled since Mon 13 Nov 2023 07:55:03 PM EST
system default destination: imp2023
```

2)

« **lp /etc/issue** » pour imprimer le fichier.

```
[root@localhost ~]# lp /etc/issue
request id is imp2023-5 (1 file(s))
[root@localhost ~]# lpq
imp2023 is ready
no entries
```

Et « **lpq** » pour voir la queue.

```
[root@localhost ~]# lpq
imp2023 is ready
Rank  Owner  Job    File(s)                Total Size
1st   root    13     (stdin)                 0 bytes
[root@localhost ~]#
[root@localhost ~]# lp /etc/issue
request id is imp2023-5 (1 file(s))
[root@localhost ~]# lpq
imp2023 is ready
no entries
```


3)

Pour enlever le job on utilise la commande « **lprm** » et « **lpq** » pour vérifier.

```
[root@localhost ~]# lprm
lprm: No active jobs on imp2023.
[root@localhost ~]# lpq
imp2023 is ready
no entries
```

d) Backup avec dump-restore :

1)

Commande « **yum install dump** ».

```
[root@localhost ~]# yum install dump
Last metadata expiration check: 1:22:07 ago on Sat 11 Nov 2023 07:09:55 PM EST.
Dependencies resolved.
=====
Package                               Architecture      Version           Repository        Size
=====
Installing:
dump                                  x86_64            1:0.4-0.36.b46.e18    baseos            164 k
Installing dependencies:
e2fsprogs-libs                       x86_64            1.45.6-5.e18         baseos            232 k
rmt                                   x86_64            2:1.5.3-13.e18       baseos            51 k
Transaction Summary
=====
Install 3 Packages

Total download size: 447 k
Installed size: 931 k
Is this ok [y/N]: y

Downloading Packages:
(1/3): rmt-1.5.3-13.e18.x86_64.rpm    9.9 kB/s | 51 kB      00:05
(2/3): dump-0.4-0.36.b46.e18.x86_64.rpm 31 kB/s | 164 kB      00:05
(3/3): e2fsprogs-libs-1.45.6-5.e18.x86_64.rpm 45 kB/s | 232 kB      00:05
-----
Total                                       42 kB/s | 447 kB      00:10
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                     1/1
  Installing : rmt-2:1.5.3-13.e18.x86_64         1/3
  Installing : e2fsprogs-libs-1.45.6-5.e18.x86_64 2/3
  Running scriptlet: e2fsprogs-libs-1.45.6-5.e18.x86_64 2/3
  Installing : dump-1:0.4-0.36.b46.e18.x86_64    3/3
  Running scriptlet: dump-1:0.4-0.36.b46.e18.x86_64 3/3
  Verifying   : dump-1:0.4-0.36.b46.e18.x86_64    1/3
  Verifying   : e2fsprogs-libs-1.45.6-5.e18.x86_64 2/3
  Verifying   : rmt-2:1.5.3-13.e18.x86_64        3/3

Installed:
dump-1:0.4-0.36.b46.e18.x86_64  e2fsprogs-libs-1.45.6-5.e18.x86_64  rmt-2:1.5.3-13.e18.x86_64

Complete!
[root@localhost ~]#
```

2)

On va utiliser la commande « **dump -0 -f /partage/backup.dump /dev/sda3** ».

```
[root@localhost ~]# dump -0 -f /partage/backup.dump /dev/sda3
DUMP: Date of this level 0 dump: Sat Nov 11 20:52:55 2023
DUMP: Dumping /dev/sda3 (/boot) to /partage/backup.dump
DUMP: Label: none
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 254132 blocks.
DUMP: Volume 1 started with block 1 at: Sat Nov 11 20:52:55 2023
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /partage/backup.dump
DUMP: Volume 1 completed at: Sat Nov 11 20:52:56 2023
DUMP: Volume 1 254730 blocks (248.76MB)
DUMP: Volume 1 took 0:00:01
DUMP: Volume 1 transfer rate: 254730 kB/s
DUMP: 254730 blocks (248.76MB) on 1 volume(s)
DUMP: finished in 1 seconds, throughput 254730 kBytes/sec
DUMP: Date of this level 0 dump: Sat Nov 11 20:52:55 2023
DUMP: Date this dump completed: Sat Nov 11 20:52:56 2023
DUMP: Average transfer rate: 254730 kB/s
DUMP: DUMP IS DONE
[root@localhost ~]#
```

3)

Command « **restore -i -f /partage/backup.dump** ».

```
[root@localhost ~]# restore -i -f /partage/backup.dump
restore > ls
.:
.vmlinuz-4.18.0-372.9.1.el8.x86_64.hmac
.vmlinuz-4.18.0-477.15.1.el8_8.x86_64.hmac
System.map-4.18.0-372.9.1.el8.x86_64
System.map-4.18.0-477.15.1.el8_8.x86_64
config-4.18.0-372.9.1.el8.x86_64
config-4.18.0-477.15.1.el8_8.x86_64
efi/
grub2/
initramfs-0-rescue-a1b1485257fc49359ed3128b2857f9d0.img
initramfs-4.18.0-372.9.1.el8.x86_64.img
initramfs-4.18.0-477.15.1.el8_8.x86_64.img
loader/
lost+found/
symvers-4.18.0-372.9.1.el8.x86_64.gz
symvers-4.18.0-477.15.1.el8_8.x86_64.gz
vmlinuz-0-rescue-a1b1485257fc49359ed3128b2857f9d0
vmlinuz-4.18.0-372.9.1.el8.x86_64
vmlinuz-4.18.0-477.15.1.el8_8.x86_64
restore >
```

4)

Pour savoir à quel fichier il correspond, on va utiliser la commande « **find / -name grub.cfg** » qui nous donne le path complet du fichier à l'origine « **/boot/grub2/grub.cfg** ».

```
[root@localhost ~]# find / -name grub.cfg
/boot/grub2/grub.cfg
[root@localhost ~]# restore -i -f /partage/backup.dump
restore > cd /grub2
restore > ls
./grub2:
device.map  fonts/      grub.cfg    grubenv     i386-pc/

restore > add grub.cfg
restore: ./grub2: File exists
restore > extract
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume # (none if no more volumes): 1
set owner/mode for './?' [yn] y
restore > q
[root@localhost ~]# ls -l grub.cfg
ls: cannot access 'grub.cfg': No such file or directory
[root@localhost ~]# restore -i -f /partage/backup.dump
restore > q
```

Le fichier récupéré a été déposé sur « **./grub2/grub.cfg** ».

```
[root@localhost ~]# find / grep grub.cfg
./grub2/grub.cfg
```

Pour savoir à quel fichier correspond, on va utiliser la commande « **find / -name grub.cfg** » qui nous donne le path complet du fichier à l'origine « **/boot/grub2/grub.cfg** ».

```
[root@localhost ~]# find / -name grub.cfg
/boot/grub2/grub.cfg
```

3.

a.

La commande « **uname -a** » pour afficher toutes les informations concernant le système d'exploitation.

```
hincuiion@maya ~$ uname -a
Linux maya 6.1.57-gentoo-dist #1 SMP PREEMPT_DYNAMIC Fri Oct 13 13:51:07 EDT 2023 x86_64 Intel(R) Xeon(R) CPU W3670 @ 3.20GHz GenuineIntel GNU/Linux
```

Et la commande « **uname -r** » pour afficher la version de kernel.

```
hincuiion@maya ~$ uname -r
6.1.57-gentoo-dist
```

b.

Avec la commande « **lscpu** », on remarque que la marque du processeur est donnée sous « *Model name* », avec sa vitesse de 3.20GHz. À cet effet, le motherboard ne dispose que d'un socket pour processeur physique, ce qui veut dire qu'il y a seulement 1 processeur physique. Ce processeur, lui, possède 6 core physique pour 1 socket de processeur.

```
hincuiion@maya ~$ lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 36 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 12
On-line CPU(s) list: 0-11
Vendor ID: GenuineIntel
Model name: Intel(R) Xeon(R) CPU W3670 @ 3.20GHz
CPU family: 6
Model: 44
Thread(s) per core: 2
Core(s) per socket: 6
Socket(s): 1
Stepping: 2
Frequency boost: enabled
CPU(s) scaling MHz: 51%
CPU max MHz: 3201.0000
CPU min MHz: 1600.0000
BogoMIPS: 6417.78
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ht
erfmon pebs bts rep_good nopl xtopology nonstop tsc cpuid aperfperf pni pclmulqdq dtes64 monitor ds_cpl vmx s
popcnt aes lahf_lm epb pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid dtherm ida arat flush_ll
Virtualization features:
Virtualization: VT-x
Caches (sum of all):
L1d: 192 KiB (6 instances)
L1i: 192 KiB (6 instances)
L2: 1.5 MiB (6 instances)
L3: 12 MiB (1 instance)
NUMA:
NUMA node(s): 1
NUMA node0 CPU(s): 0-11
Vulnerabilities:
Gather data sampling: Not affected
Itlb multihit: KVM: Mitigation: VMX disabled
L1tf: Mitigation; PTE Inversion; VMX conditional cache flushes, SMT vulnerable
Mds: Vulnerable; Clear CPU buffers attempted, no microcode; SMT vulnerable
Meltdown: Mitigation; PTI
Mmio stale data: Unknown; No mitigations
Retbleed: Not affected
Spec rstack overflow: Not affected
Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl
Spectre v1: Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP conditional, RSB filling, PBRBS-eIBRS Not affected
Srbds: Not affected
Tsx async abort: Not affected
hincuiion@maya ~$
```

c.

La commande « **free -h** ».

```
hincuiion@maya ~$ free -h
              total        used        free      shared  buff/cache   available
Mem:          15Gi       2.2Gi       3.8Gi         33Mi       9.6Gi       13Gi
Swap:          15Gi           0B         15Gi
```

d.

La commande « **df -h /dev/sda1** », donc on a 474M sur la partition 1 du disque A.

```
hincuiion@maya ~$ df -h /dev/sda1
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       474M  121M  325M  28% /boot
```

e.

La commande « **uptime** », dernier démarrage de système a eu lieu le 12 :09 :23.

```
hincuion@maya ~ $ uptime
12:09:23 up 12 days, 17:02, 3 users, load average: 0.17, 0.22, 0.15
```

f.

Commande « **systemctl get-default** », le target de démarrage par défaut est **graphical** qui correspond au runlevel 5.

```
hincuion@maya ~ $ systemctl get-default
graphical.target
```

g.

La commande « **timedatectl** ».

```
hincuion@maya ~ $ timedatectl
Local time: Sun 2023-11-12 17:20:15 UTC
Universal time: Sun 2023-11-12 17:20:15 UTC
RTC time: Sun 2023-11-12 17:20:15
Time zone: n/a (UTC, +0000)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

La commande « **curl ipinfo.io** », nous donne la ville de Montréal.

```
hincuion@maya ~ $ curl ipinfo.io
{
  "ip": "132.204.25.78",
  "hostname": "maya.iro.umontreal.ca",
  "city": "Montréal",
  "region": "Quebec",
  "country": "CA",
  "loc": "45.5088,-73.5878",
  "org": "AS376 Réseau d'Informations Scientifiques du Quebec (RISQ Inc.)",
  "postal": "H3H",
  "timezone": "America/Toronto",
  "readme": "https://ipinfo.io/missingauth"
}
```

h.

Avec « **last -s 2023-11-01 -t 2023-11-10 dift3830** », on peut voir que l'utilisateur dift3830 a été branché 4 fois.

```
hincuion@maya ~ $ last -s 23-11-01 -t 23-11-10 dift3830
dift3830 pts/0      132.204.24.131  Mon Nov 6 13:55 - 13:55 (00:00)
dift3830 ssh      132.204.24.131  Mon Nov 6 13:55 - 13:55 (00:00)
dift3830 pts/0      132.204.24.131  Mon Nov 6 13:53 - 13:55 (00:01)
dift3830 ssh      132.204.24.131  Mon Nov 6 13:53 - 13:55 (00:01)
```

i.

Commande « **lsblk -o NAME,MODEL /dev/sda** ».

```
hincuion@maya ~ $ lsblk -o NAME,MODEL /dev/sda
NAME    MODEL
sda     INTEL SSDSC2BW180A4
├─sda1
├─sda2
└─sda3
```

j.

Pour trouver le modèle du lecteur DVD on utilise la commande « **more /proc/scsi/scsi** ».

```
hincui@maya ~ $ more /proc/scsi/scsi
```

Attached devices:

Host: scsi0 Channel: 00 Id: 00 Lun: 00

Vendor: HL-DT-ST Model: DVD-RAM GH22NS50 Rev: TN03

Type: CD-ROM ANSI SCSI revision: 05

Et avec le modèle on peut trouver sur internet la couleur du lecteur cd-rom et ça nous donne la couleur noire.



LG GH22NS50 optical disc drive Internal Black

4.

Le test du script:

```
root@localhost:~# ls
.:
bin Desktop Downloads halt initial-setup-ks.cfg mots-d-enfants pensee RPM.tar whichare
anaconda-ks.cfg breton.txt Documents firewall.pdf hello.pl jeep.jpg Music Pictures Templates Word.doc
root@localhost:~# chmod +x fuseau.pl
root@localhost:~# ./fuseau.pl 12:38 PST
9:38
root@localhost:~# ./fuseau.pl 12:38 MST
10:38
root@localhost:~# ./fuseau.pl 12:38 CST
11:38
root@localhost:~# ./fuseau.pl 12:38 AST
1:38
root@localhost:~# ./fuseau.pl 12:38 NST
2:08
root@localhost:~# ./fuseau.pl 13:38 NST
heure invalide
root@localhost:~# ./fuseau.pl 12:38 QST
fuseau horaire invalide
root@localhost:~#
```

Pour donner accès « **chmod +x fuseau.pl** » et au besoin si le script doit être convertie à Unix : « **dos2unix fuseau.pl** ».

Script du fichier « **fuseau.pl** ».

```
#!/usr/bin/perl

# Utiliser les règles strictes et warnings pour facilité la programmation
use strict;
use warnings;

# Terminer le script si le nombre de paramètres n'est pas égal à 2
die "Nombre incorrecte de parametres\n" unless @ARGV == 2;

# Diviser le premier argument en heures et minutes
my ($heure, $minutes) = split /:/, $ARGV[0];
# Assigner le deuxième argument à la variable fuseau
my $fuseau = $ARGV[1];

# Terminer le script si l'heure n'est pas comprise entre 1 et 12
die "heure invalide\n" if $heure < 1 || $heure > 12;
# Terminer le script si les minutes ne sont pas comprises entre 0 et 59
die "minutes invalide\n" if $minutes < 0 || $minutes > 59;

# Définir un hash avec les décalages horaires pour chaque fuseau
my %fuseau_offsets = (
    PST => -3,
    MST => -2,
    CST => -1,
    AST => 1,
    NST => 1.5
);

# Terminer le script si le fuseau horaire fourni n'est pas valide
die "fuseau horaire invalide\n" unless exists $fuseau_offsets{$fuseau};

# Gérer le cas spécial du fuseau horaire NST
if ($fuseau eq 'NST') {
    $minutes += 30; # Ajouter 30 minutes pour NST
    if ($minutes > 59) {
        $minutes -= 60; # Soustraire 60 si les minutes dépassent 59
        $heure = ($heure + 2) % 12; # Ajouter 2 heures en conséquence
    } else {
        $heure = ($heure + 1) % 12; # Sinon, ajouter seulement 1 heure
    }
} else {
    # Ajouter le décalage horaire au nombre d'heures et utiliser le modulo pour rester dans (1,12)
    $heure = ($heure + $fuseau_offsets{$fuseau}) % 12;
}

# Si les heures sont à 0, les ramener à 12
$heure = 12 if $heure == 0;
# Formater les minutes pour avoir toujours deux chiffres
$minutes = sprintf("%02d", $minutes);

# Afficher l'heure finale
print "$heure:$minutes\n";

# Sortir du script sans erreur
exit 0;

__END__
"fuseau.pl" 55L, 1790C
```