


Dernière mise-à-jour : 2020/01/30 03:28

LSF104 - Commandes de Base et Outils de Manipulation de Fichiers Texte



A faire - Vous êtes actuellement connecté(e) en tant que root dans votre terminal. Avant de procéder plus loin, tapez la commande exit et appuyez sur la touch  Entrée.

La commande stty

Dès votre connexion à un système Linux, Il est conseillé de lancer la commande suivante

```
$ stty -a [Entrée]
```

Faites attention à utiliser des minuscules. En effet Linux différencie les minuscules et les majuscules.

```
trainee@SLES12SP1:~> stty -a
speed 38400 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echprt
```

```
echoctl echoke
```

Dans l'information qui s'affiche à l'écran, cherchez la chaîne `intr =`. Si la valeur est «Del», il faut utiliser la touche **Suppr** au lieu de la commande **^C** pour interrompre un programme en cours d'exécution dans un terminal.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **stty** pour visualiser les options de la commande.

La commande date

Cette commande affiche la date et l'heure de la machine. La commande peut aussi être utilisée pour régler la date du système :

```
trainee@SLES12SP1:~> date  
Fri 30 Sep 15:19:54 CEST 2016
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **date** pour visualiser les options de la commande.

La commande who

Cette commande affiche les utilisateurs connectés au système:

```
trainee@SLES12SP1:~> who
trainee pts/0      2016-09-30 10:24 (10.0.2.2)
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **who** pour visualiser les options de la commande.

La commande df

Cette commande affiche l'espace disque libre sur chacun des unités montés (connectés au système):

```
trainee@SLES12SP1:~> df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda2        18867200 3510848  15185952  19% /
devtmpfs         188376      0    188376   0% /dev
tmpfs            197260      0    197260   0% /dev/shm
tmpfs            197260    5716    191544   3% /run
tmpfs            197260      0    197260   0% /sys/fs/cgroup
/dev/sda2        18867200 3510848  15185952  19% /.snapshots
/dev/sda2        18867200 3510848  15185952  19% /var/lib/mariadb
/dev/sda2        18867200 3510848  15185952  19% /var/lib/libvirt/images
/dev/sda2        18867200 3510848  15185952  19% /var/lib/mailman
```

```

/dev/sda2      18867200 3510848 15185952 19% /var/tmp
/dev/sda2      18867200 3510848 15185952 19% /var/lib/pgsql
/dev/sda2      18867200 3510848 15185952 19% /usr/local
/dev/sda2      18867200 3510848 15185952 19% /tmp
/dev/sda2      18867200 3510848 15185952 19% /var/spool
/dev/sda2      18867200 3510848 15185952 19% /srv
/dev/sda2      18867200 3510848 15185952 19% /var/opt
/dev/sda2      18867200 3510848 15185952 19% /opt
/dev/sda2      18867200 3510848 15185952 19% /var/log
/dev/sda2      18867200 3510848 15185952 19% /var/lib/mysql
/dev/sda2      18867200 3510848 15185952 19% /home
/dev/sda2      18867200 3510848 15185952 19% /var/crash
/dev/sda2      18867200 3510848 15185952 19% /var/lib/named
/dev/sda2      18867200 3510848 15185952 19% /boot/grub2/x86_64-efi
/dev/sda2      18867200 3510848 15185952 19% /boot/grub2/i386-pc

```

Les unités sont en blocs. Afin d'*humaniser* la sortie, il est possible d'utiliser l'option **-h**. Une option est aussi connue sous le nom **parameter**, **switch** ou **flag** :

```

trainee@SLES12SP1:~> df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        18G   3.4G   15G   19% /
devtmpfs         184M     0   184M    0% /dev
tmpfs            193M     0   193M    0% /dev/shm
tmpfs            193M   5.6M   188M    3% /run
tmpfs            193M     0   193M    0% /sys/fs/cgroup
/dev/sda2        18G   3.4G   15G   19% /.snapshots
/dev/sda2        18G   3.4G   15G   19% /var/lib/mariadb
/dev/sda2        18G   3.4G   15G   19% /var/lib/libvirt/images
/dev/sda2        18G   3.4G   15G   19% /var/lib/mailman
/dev/sda2        18G   3.4G   15G   19% /var/tmp
/dev/sda2        18G   3.4G   15G   19% /var/lib/pgsql
/dev/sda2        18G   3.4G   15G   19% /usr/local
/dev/sda2        18G   3.4G   15G   19% /tmp

```

```
/dev/sda2      18G  3.4G  15G  19% /var/spool
/dev/sda2      18G  3.4G  15G  19% /srv
/dev/sda2      18G  3.4G  15G  19% /var/opt
/dev/sda2      18G  3.4G  15G  19% /opt
/dev/sda2      18G  3.4G  15G  19% /var/log
/dev/sda2      18G  3.4G  15G  19% /var/lib/mysql
/dev/sda2      18G  3.4G  15G  19% /home
/dev/sda2      18G  3.4G  15G  19% /var/crash
/dev/sda2      18G  3.4G  15G  19% /var/lib/named
/dev/sda2      18G  3.4G  15G  19% /boot/grub2/x86_64-efi
/dev/sda2      18G  3.4G  15G  19% /boot/grub2/i386-pc
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **df** pour visualiser les options de la commande.

La commande free

Cette commande affiche les détails de la mémoire disponible :

```
trainee@SLES12SP1:~> free
              total        used        free      shared    buffers     cached
Mem:      394524      385872        8652        5120         452      296176
-/+ buffers/cache:      89244      305280
Swap:      2103292          600      2102692
```

Les unités sont en blocs. Afin d'*humaniser* la sortie, il est possible d'utiliser l'option **-h**.

```
trainee@SLES12SP1:~> free -h
```

	total	used	free	shared	buffers	cached
Mem:	385M	376M	8.4M	5.0M	452K	289M
-/+ buffers/cache:		87M	298M			
Swap:	2.0G	600K	2.0G			

Options de la commande



A faire : Utilisez l'option **-help** de la commande **free** pour visualiser les options de la commande.

La commande whoami

Cette commande affiche le nom associé à l'UID courant effectif, autrement dit, le nom de votre compte courant :

```
trainee@SLES12SP1:~> whoami
trainee
```

Devenez maintenant l'administrateur **root** :

```
trainee@SLES12SP1:~> su -
Password: fenestros
SLES12SP1:~ #
```



Important : Notez que le mot de passe saisi ne sera PAS visible.

Saisissez maintenant la commande **whoami** de nouveau :

```
SLES12SP1:~ # whoami  
root  
SLES12SP1:~ #
```



Important : Notez maintenant que vous êtes root.

Saisissez en suite la commande **exit** pour redevenir l'utilisateur **trainee** :

```
SLES12SP1:~ # exit  
logout  
trainee@SLES12SP1:~>
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **whoami** pour visualiser les options de la commande.

La commande pwd

Cette commande affiche le répertoire courant de travail :

```
trainee@SLES12SP1:~> pwd  
/home/trainee
```

Options de la commande



A faire : Utilisez la commande **help** avec l'option **pwd** pour visualiser les options de la commande.

La commande cd

Cette commande permet de changer de répertoire courant pour le répertoire passé en argument à la commande :

```
trainee@SLES12SP1:~> cd /tmp
trainee@SLES12SP1:/tmp> pwd
/tmp
trainee@SLES12SP1:/tmp>
```

Options de la commande



A faire : Utilisez la commande **help** avec l'option **cd** pour visualiser les options de la commande.

La commande ls

Cette commande permet de lister le contenu d'un répertoire passé en argument à la commande. Si aucun argument n'est spécifié, la commande liste le contenu du répertoire courant :


```
trainee@SLES12SP1:/tmp> ls
hsperfdata_root
inode
managera1411267841657715235client
managera3336001029897679475server
managera4847938942232964844client
managera5050357016347721452server
systemd-private-04f820fa26c745be8ddba814c6292f21-rtkit-daemon.service-o4lKP5
systemicontmp5578677472245134133dat
systemicontmp7082392205020802884dat
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **ls** pour visualiser les options de la commande.

La commande lsof

La commande **lsof** affiche des informations sur les fichiers ouverts par des processus :

```
trainee@SLES12SP1:/tmp> su -
Password:
SLES12SP1:~ # lsof | more
```

COMMAND	PID	TID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1		root	cwd	DIR	0,34	156	256	/
systemd	1		root	rtd	DIR	0,34	156	256	/
systemd	1		root	txt	REG	0,34	1218976	74832	/usr/lib/systemd/systemd
systemd	1		root	mem	REG	0,34	18904	23416	/lib64/libdl-2.19.so

```

systemd      1      root  mem      REG      0,34   416064   24152 /usr/lib64/libpcres.so.1.2.1
systemd      1      root  mem      REG      0,34  1974416   23410 /lib64/libc-2.19.so
systemd      1      root  mem      REG      0,34   137440   23438 /lib64/libpthread-2.19.so
systemd      1      root  mem      REG      0,34    42672   23442 /lib64/librt-2.19.so
systemd      1      root  mem      REG      0,34   150128   24032
/usr/lib64/libseccomp.so.2.1.0
systemd      1      root  mem      REG      0,34    43960   25095 /lib64/libapparmor.so.1.0.4
systemd      1      root  mem      REG      0,34    84864   24581 /usr/lib64/libkmod.so.2.2.7
systemd      1      root  mem      REG      0,34    18976   24919 /lib64/libcap.so.2.22
systemd      1      root  mem      REG      0,34   100312   24963
/usr/lib64/libaudit.so.1.0.0
systemd      1      root  mem      REG      0,34    60096   59333 /lib64/libpam.so.0.83.1
systemd      1      root  mem      REG      0,34    40880   23873 /lib64/libwrap.so.0.7.6
systemd      1      root  mem      REG      0,34   138792   25858 /lib64/libselinux.so.1
systemd      1      root  mem      REG      0,34   158192   23402 /lib64/ld-2.19.so
systemd      1      root   0u      CHR      1,3      0t0      4298 /dev/null
systemd      1      root   1u      CHR      1,3      0t0      4298 /dev/null
systemd      1      root   2u      CHR      1,3      0t0      4298 /dev/null
systemd      1      root   3r      0000      0,10      0      4268 anon_inode
systemd      1      root   4u      0000      0,10      0      4268 anon_inode
--More--

```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **lsuf** pour visualiser les options de la commande.

La commande touch

Cette commande sert à modifier l'horodatage de la date de dernière modification du contenu (**mtime**) et la date du dernier accès (**atime**), d'un ou de plusieurs fichiers passé(s) en argument(s), selon la date courante. Si le(s) fichier(s) n'existe(nt) pas, il(s) est (sont) créé(s) :

```
SLES12SP1:~ # exit
logout
trainee@SLES12SP1:/tmp> touch test
trainee@SLES12SP1:/tmp> ls
hsperfdata_root
inode
managera1411267841657715235client
managera3336001029897679475server
managera4847938942232964844client
managera5050357016347721452server
systemd-private-04f820fa26c745be8ddba814c6292f21-rtkit-daemon.service-o4lKP5
systemicontmp5578677472245134133dat
systemicontmp7082392205020802884dat
test
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **touch** pour visualiser les options de la commande.

La commande echo

Cette commande écrit les arguments vers la sortie standard (autrement dit à l'écran) :

```
trainee@SLES12SP1:/tmp> echo fenestros
fenestros
```

Options de la commande



A faire : Utilisez la commande **help** avec l'option **echo** pour visualiser les options de la commande.

La commande cp

La commande cp permet de copier une source vers une destination ou de multiples sources vers un répertoire :

```
trainee@SLES12SP1:/tmp> cp test ~
trainee@SLES12SP1:/tmp> ls -l ~
total 4
drwxr-xr-x 1 trainee users  0 May  1 15:58 bin
drwxr-xr-x 1 trainee users  0 May  2 16:11 Desktop
drwxr-xr-x 1 trainee users  0 May  2 16:11 Documents
drwxr-xr-x 1 trainee users  0 May  2 16:11 Downloads
drwxr-xr-x 1 trainee users  0 May  2 16:11 Music
drwxr-xr-x 1 trainee users  0 May  2 16:11 Pictures
drwxr-xr-x 1 trainee users  0 May  2 16:11 Public
drwxr-xr-x 1 trainee users 20 May  1 15:58 public_html
drwxr-xr-x 1 trainee users  0 May  2 16:11 Templates
-rw-r--r-- 1 trainee users  0 Oct  1 06:01 test
drwxr-xr-x 1 trainee users  0 May  2 16:11 Videos
```

```
-rw-r--r-- 1 trainee users 391 Sep 30 10:27 vitext
```



Important : Notez l'utilisation du caractère ~ (tilde) qui est un caractère spécial indiquant le répertoire personnel de l'utilisateur courant, dans ce cas /home/trainee.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **cp** pour visualiser les options de la commande.

La commande file

Cette commande permet de connaître le type d'un fichier:

```
trainee@SLES12SP1:/tmp> file ~/test  
/home/trainee/test: empty
```



Important : Notez que la commande vous indique le type de fichier en fonction de son contenu. Dans l'exemple précédent, puisque le fichier est vide, la commande file ne peut pas indiquer le type de fichier.

Redirigez, en utilisant le caractère >, la sortie de la commande **echo** vers le fichier **/home/trainee/test** de façon à ce que ce dernier contient le texte **fenestros** :

```
trainee@SLES12SP1:/tmp> echo "fenestros" > ~/test
```

En utilisant de nouveau la commande **file**, celle-ci est capable de vous indiquer le type de fichier :

```
trainee@SLES12SP1:/tmp> file ~/test  
/home/trainee/test: ASCII text
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **file** pour visualiser les options de la commande.

La commande cat

La commande **cat** permet de concaténer les fichiers passés en argument, ou de l'entrée standard (le **clavier**), vers la sortie standard (l'**écran**). Dans le cas où il n'y a qu'un seul fichier passé en argument, le contenu de celui-ci est affiché à l'écran :

```
trainee@SLES12SP1:/tmp> cat ~/test  
fenestros
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **cat** pour visualiser les options de la commande.

La commande mv

La commande **mv** permet déplacer ou de renommer un fichier ou répertoire.

Utilisez la commande **mv** pour déplacer le fichier **test** de votre répertoire personnel vers le répertoire courant :

```
trainee@SLES12SP1:/tmp> mv test TeSt
trainee@SLES12SP1:/tmp> ls -l
total 12
drwxr-xr-x 1 root    root      8 Sep 30 10:26 hsperfdata_root
drwxr-xr-x 1 trainee users    48 Aug 18 13:14 inode
-rwxrwxrwx 1 root    root    50048 May  3 14:01 managera1411267841657715235client
-rwxrwxrwx 1 root    root    50048 May  3 14:01 managera3336001029897679475server
-rwxrwxrwx 1 root    root    50048 May  3 14:01 managera4847938942232964844client
-rwxrwxrwx 1 root    root    50048 May  3 14:01 managera5050357016347721452server
drwx----- 1 root    root       6 May  3 13:54 systemd-private-04f820fa26c745be8ddba814c6292f21-rtkit-
daemon.service-o4lKP5
-rw-r--r-- 1 root    root       0 May  3 14:01 systemicontmp5578677472245134133dat
drwxr-xr-x 1 root    root       0 May  3 14:01 systemicontmp7082392205020802884dat
-rw-r--r-- 1 trainee users    10 Oct  1 06:06 TeSt
```



Important : Notez l'utilisation du raccourci `.` pour indiquer le répertoire courant.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **mv** pour visualiser les options de la commande.

La commande mkdir

La commande **mkdir** permet de créer un répertoire.

```
trainee@SLES12SP1:/tmp> cd ~
trainee@SLES12SP1:~> mkdir testdir
trainee@SLES12SP1:~> ls
bin      Documents  Music      Public      Templates  Videos
Desktop  Downloads  Pictures   public_html testdir     vitext
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **mkdir** pour visualiser les options de la commande.

La commande rmdir

La commande **rmdir** permet de supprimer un répertoire **vide** :

```
trainee@SLES12SP1:~> rmdir testdir
trainee@SLES12SP1:~> ls
bin      Documents  Music      Public      Templates  vitext
Desktop  Downloads  Pictures   public_html Videos
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **rmdir** pour visualiser les options de la commande.

La commande rm

La commande rm permet de supprimer un répertoire **vide ou non** ou un fichier :

```
trainee@SLES12SP1:~> mkdir testdir1
trainee@SLES12SP1:~> cd /tmp
trainee@SLES12SP1:/tmp> echo "fenestros" > TeSt
trainee@SLES12SP1:/tmp> cd ~
trainee@SLES12SP1:~> mv /tmp/TeSt ~/testdir1
trainee@SLES12SP1:~> ls -lR testdir1/
testdir1/:
total 4
-rw-r--r-- 1 trainee users 10 Oct  1 06:47 TeSt
trainee@SLES12SP1:~> rmdir testdir1/
rmdir: failed to remove 'testdir1/': Directory not empty
trainee@SLES12SP1:~> rm -rf testdir1/
trainee@SLES12SP1:~> ls
bin      Documents  Music      Public      Templates  vitext
Desktop  Downloads  Pictures    public_html Videos
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **rm** pour visualiser les options de la commande.

La commande sort

Cette commande trie dans le canal d'entrée et retourne à l'écran une liste triée.

```
trainee@SLES12SP1:~> ls | sort
aac
abc
bca
bin
Desktop
Documents
Downloads
Music
Pictures
Public
public_html
Templates
Videos
vitext
xyz
trainee@SLES12SP1:~> ls | sort -r
xyz
vitext
Videos
Templates
public_html
Public
Pictures
Music
Downloads
Documents
Desktop
bin
```

```
bca  
abc  
aac
```



Important : Notez l'utilisation du caractère spécial |, appelé un tube ou encore un pipe. Un tube est utilisé pour présenter sur l'entrée standard de la commande qui suit, la sortie standard de la commande qui précède.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **sort** pour visualiser les options de la commande.




La commande more

Cette commande affiche le contenu d'un fichier texte et l'envoi page par page au canal de sortie à l'aide de la touche Espace :

```
trainee@SLES12SP1:~> more /etc/services  
#  
# Network services, Internet style  
#  
# Note that it is presently the policy of IANA to assign a single well-known  
# port number for both TCP and UDP; hence, most entries here have two entries  
# even if the protocol doesn't support UDP operations.  
#  
# This list could be found on:
```

```
#          http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbe
rs.xml
#
# See also: services(5)
#
#
# SERVICE NAME AND TRANSPORT PROTOCOL PORT NUMBER REGISTRY
#
# (last updated on 2011-11-23)
#
# *****
# * Since we merged and converted this registry it loads more slowly. We *
# * know it should load faster and are working on ways to achieve that.  *
# * Thank you for your patience.                                         *
# *****
#
--More-- (0%)
```



Important : L'utilisation de la touche  **Entrée** permet de défiler le fichier ligne par ligne.
L'utilisation de la touche  **Barre d'espace** permet de défiler le fichier écran par écran.
L'utilisation de la touche  **Q** permet de revenir au prompt.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **more** pour visualiser les options de la commande.

La commande less

La commande **less** produit un résultat similaire à la commande **more**. Utilisez la commande **less** en vous référant à l'aide de la commande avec **less -help**. Laquelle des deux commandes vous semble la plus puissante ?

```
trainee@SLES12SP1:~> less /etc/services
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
#
# This list could be found on:
#      http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml
#
# See also: services(5)
#
# SERVICE NAME AND TRANSPORT PROTOCOL PORT NUMBER REGISTRY
#
# (last updated on 2011-11-23)
#
# *****
# * Since we merged and converted this registry it loads more slowly. We *
# * know it should load faster and are working on ways to achieve that. *
# * Thank you for your patience. *
# *****
/etc/services lines 1-22/16228 0%
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **less** pour visualiser les options de la commande.

La commande find

Cette commande sert à rechercher un ou des fichiers dans le répertoire courant ou le répertoire spécifié en argument :

```
trainee@SLES12SP1:~> find acc
find: 'acc': No such file or directory
trainee@SLES12SP1:~> find aac
aac
```



Important : Notez que si le fichier n'existe pas le système vous en informe clairement. Notez aussi que ce fichier existe le système vous en informe en vous indiquant son nom.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **find** pour visualiser les options de la commande.

La commande su

La commande **su** permet d'assumer l'identité d'un autre utilisateur du système à condition de connaître son mot de passe. Exécutée sans argument, le système suppose que vous souhaitez devenir **root** :

```
trainee@SLES12SP1:~> su -  
Password: fenestros
```



Important : Notez que le mot de passe saisi ne sera PAS visible.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **su** pour visualiser les options de la commande.

Les commandes locate et updatedb

La commande **locate** sert à rechercher un ou des fichiers dans l'ensemble du système de fichiers en commençant à la racine (/) en spécifiant une chaîne à rechercher en argument à la commande. La commande **locate** utilise une base de données afin d'effectuer sa recherche. Pour construire ou mettre à jour cette base de données avant l'utilisation de la commande pour une recherche, il faut utiliser la commande **updatedb** en tant que **root**.

La base de données par défaut est **/var/lib/mlocate/mlocate.db** :

```
SLES12SP1:~ # ls -l /var/lib/mlocate/mlocate.db
```

```
-rw-r--r-- 1 root root 3804323 Sep 30 10:30 /var/lib/mlocate/mlocate.db
```



Important : Pour plus d'information concernant le format de la base de données, consultez **man 5 locatedb**.

La commande **updatedb** peut être configurée en éditant son fichier de configuration **/etc/updatedb.conf** :

```
SLES12SP1:~ # cat /etc/updatedb.conf
# /etc/updatedb.conf: config file for mlocate

# This file sets variables that are used by updatedb.
# For more info, see the updatedb.conf(5) manpage.

# Filesystems that are pruned from updatedb database
PRUNEFS="afs anon_inodefs auto autofs bdev binfmt binfmt_misc cgroup cifs coda configfs cramfs cpuset debugfs
devfs devpts devtmps ecryptfs eventpollfs exofs futexfs ftpfs fuse fusectl gfs gfs2 hostfs hugetlbfs inotifyfs
iso9660 jffs2 lustre misc mqueue ncfs nfs NFS nfs4 nfsd nnpfs ocfs ocfs2 pipefs proc ramfs rpc_pipefs securityfs
selinuxfs sfs shfs smbfs sockfs spufs sshfs subfs supermount sysfs tmpfs ubifs udf usbfs vboxsf vperfctrfs"

# Paths which are pruned from updatedb database
PRUNEPATHS="/tmp /var/tmp /var/cache /var/lock /var/run /var/spool /mnt /cdrom /usr/tmp /proc /media /sys
/.snapshots /var/run/media"

# Folder names that are pruned from updatedb database
PRUNENAMES=".git .hg .svn CVS"

# Skip bind mounts.
PRUNE_BIND_MOUNTS="yes"
```

L'utilisation des deux commandes est illustrée ci-après :


```
SLES12SP1:~ # locate aac
/lib/modules/3.12.49-11-default/kernel/drivers/scsi/aacraid
/lib/modules/3.12.49-11-default/kernel/drivers/scsi/aacraid/aacraid.ko
/usr/share/YaST2/scrconf/aaconf.scr
/usr/share/mime/audio/aac.xml
```

Options des commandes



A faire : Utilisez l'option **-help** des commandes **updatedb** et **locate** pour visualiser les options des commandes.

La commande whereis

La commande **whereis** permet une recherche de l'emplacement des exécutables, des fichiers de configuration et des manuels pour la commande passée en argument :

```
SLES12SP1:~ # whereis passwd
passwd: /usr/bin/passwd /etc/passwd /etc/passwd.YaST2save /usr/share/man/man1/passwd.1ssl.gz
/usr/share/man/man1/passwd.1.gz /usr/share/man/man5/passwd.5.gz
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **whereis** pour visualiser les options de la commande.

La commande which

La commande **which** permet une recherche de l'emplacement d'un exécutable dans le PATH de l'utilisateur courant et retourne le premier qui est trouvé :

```
SLES12SP1:~ # which passwd
/usr/bin/passwd
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **which** pour visualiser les options de la commande.

La commande uptime

Cette commande nous indique l'heure actuelle, la durée depuis laquelle le système fonctionne, le nombre d'utilisateurs actuellement connectés et la charge système moyenne pour les dernières 1 minute, 5 minutes et 15 minutes :

```
SLES12SP1:~ # uptime
07:48am up 21:25, 1 user, load average: 0.10, 0.05, 0.05
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **uptime** pour visualiser les options de la commande.

La commande w

Cette commande reprend les informations de la commande **uptime** et y ajoute des détails sur les utilisateurs connectés via un terminal :

```
SLES12SP1:~ # w
07:57:55 up 21:34,  1 user,  load average: 0.00, 0.03, 0.05
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
trainee   pts/0    10.0.2.2        04:09    0.00s  0.12s  0.00s sshd: trainee [priv]
```

La valeur JCPU indique le temps processeur utilisé par tous les processus attachés au terminal de la connexion. Cette valeur n'inclut pas les temps des anciens processus en arrière plan.

La valeur PCPU indique le temps processeur utilisé par les processus attachés au terminal de la connexion et actuellement en cours (autrement dit le processus identifié dans la colonne **WHAT**).

Options de la commande



A faire : Utilisez l'option **-help** de la commande **w** pour visualiser les options de la commande.

La commande uname

Cette commande affiche des informations sur le système :

```
SLES12SP1:~ # uname -a
Linux SLES12SP1 3.12.49-11-default #1 SMP Wed Nov 11 20:52:43 UTC 2015 (8d714a0) x86_64 x86_64 x86_64 GNU/Linux
SLES12SP1:~ # uname -s
```

```
Linux
SLES12SP1:~ # uname -n
SLES12SP1
SLES12SP1:~ # uname -r
3.12.49-11-default
SLES12SP1:~ # uname -v
#1 SMP Wed Nov 11 20:52:43 UTC 2015 (8d714a0)
SLES12SP1:~ # uname -m
x86_64
SLES12SP1:~ # uname -p
x86_64
SLES12SP1:~ # uname -i
x86_64
SLES12SP1:~ # uname -o
GNU/Linux
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **uname** pour visualiser les options de la commande.

La commande du

La commande `du` peut être utilisée pour afficher la taille des fichiers contenus dans les répertoires passés en arguments. L'utilisation suivante de la commande avec les options `-s` et `-h` sur la racine du système affiche la somme des sous-répertoires avec un affichage *humanisé* en Ko, Mo et Go :

```
SLES12SP1:~ # du -sh /* 2>/dev/null
5.1M    /bin
44M     /boot
```

```
0    /dev
19M  /etc
216K  /home
190M  /lib
16M  /lib64
0    /mnt
151M  /opt
0    /proc
6.1M  /root
5.4M  /run
11M  /sbin
0    /selinux
64K  /srv
0    /sys
60K  /tmp
2.6G  /usr
133M  /var
```



Important : Notez l'utilisation de la redirection **2>/dev/null**. Cette chaîne envoie les erreurs éventuelles, contenues dans le canal 2 appelé le canal des erreurs, à **/dev/null** de façon à ce que les erreurs n'apparaissent pas à l'écran. Le canal des erreurs sera couvert dans le cours **La Ligne de Commande**.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **du** pour visualiser les options de la commande.

La commande clear

Cette commande est utilisée pour effacer le contenu de l'écran courant du terminal :

```
SLES12SP1:~ # clear  
SLES12SP1:~ #
```

La commande exit

Cette commande ferme le terminal courant :

```
SLES12SP1:~ # exit  
logout  
trainee@SLES12SP1:~>
```

Options de la commande



A faire : Utilisez la commande **help** avec l'option **exit** pour visualiser les options de la commande.

La commande logout

Cette commande est utilisée pour se déconnecter d'un terminal de connexion en écrivant les données umtp et wmtpt dans les fichiers de journalisation.

Options de la commande



A faire : Utilisez la commande **help** avec l'option **logout** pour visualiser les options de la commande.

La commande sleep

Cette commande pause le terminal pour le nombre de secondes passé en argument.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **sleep** pour visualiser les options de la commande.

La Commande wall

wall displays a message, or the contents of a file, or otherwise its standard input, on the terminals of all currently logged in users. The command will wrap lines that are longer than 79 characters. Short lines are whitespace padded to have 79 characters. The command will always put a carriage return and new line at the end of each line. Only the superuser can write on the terminals of users who have chosen to deny messages or are using a program which automatically denies messages. Reading from a file is refused when the invoker is not superuser and the program is suid or sgid.

Ouvrez donc un deuxième terminal via ssh en tant que trainee puis dans le premier envoyez un message :

```
SLES12SP1:~ # wall this is a message from root
```

```
Broadcast message from trainee@SLES12SP1 (pts/1) (Sat Oct 1 18:39:50 2016):  
this is a message from root  
SLES12SP1:~ #
```

Constatez que le message est affiché dans le deuxième terminal :

```
Broadcast message from trainee@SLES12SP1 (pts/1) (Sat Oct 1 18:39:50 2016):  
this is a message from root
```

The seq Command

La commande **seq** affiche une séquence de nombres du PREMIER au DERNIER par pas d'un INCREMENT. La commande prend la forme suivante :

- **seq** [options] PREMIER
- **seq** [options] PREMIER DERNIER
- **seq** [options] PREMIER INCREMENT DERNIER

Par exemple :

```
SLES12SP1:~ # seq 10  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
SLES12SP1:~ # seq 20 30  
20
```



```
21
22
23
24
25
26
27
28
29
30
SLES12SP1:~ # seq 20 10 90
20
30
40
50
60
70
80
90
SLES12SP1:~ #
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **seq** pour visualiser les options de la commande.

La Commande screen

La commande **screen** est un « multiplexeur de terminaux » permettant d'ouvrir jusqu'à 10 (numérotés de 0 à 9) terminaux dans une même console,

de passer de l'un à l'autre et de les récupérer plus tard.

La commande **screen** est installée par défaut sous SLES 12 :

```
SLES12SP1:~ # which screen
/usr/bin/screen
```

Créez maintenant une session avec screen :

```
SLES12SP1:~ # screen -S mysession
```

Appuyez maintenant sur **CTRL A** puis relachez la touche **A** et appuyez sur la touche **C** pour créer un deuxième screen imbriqué dans la même session.

Pour repasser au premier screen, appuyez sur **CTRL A A**. Ceci permet de basculer entre les deux derniers screens.

Pour voir les screens actifs, utilisez la commande suivante :

```
SLES12SP1:~ # screen -ls
There is a screen on:
      11827.mysession (Attached)
1 Socket in /var/run/screens/S-root.
```

Dans votre screen, saisissez les commandes suivantes :

```
SLES12SP1:~ # sleep 9999 &
[1] 19879
SLES12SP1:~ # jobs
[1]+  Running                  sleep 9999 &
SLES12SP1:~ #
```

Pour détacher le screen, appuyez sur **CTRL A** puis relachez la touche **A** et appuyez sur la touche **D** :

```
[detached from 11827.mysession]
```

```
SLES12SP1:~ #
```

Pour rattacher le screen, saisissez la commande suivante :

```
SLES12SP1:~ # screen -r
```

Utilisez la commande jobs pour vérifier si le processus créé par la commande sleep est toujours en cours de fonctionnement :

```
SLES12SP1:~ # jobs
[1]+  Running                  sleep 9999 &
```

Pour naviguer entre les screens il convient d'appuyer sur **CTRL A** puis relachez la touche **A** et appuyez sur la touche **N** ou d'appuyer sur **CTRL A** puis relachez la touche **A** et appuyez sur la touche **P**.

Détachez de nouveau le screen actuel en appuyant sur **CTRL A** puis en relachant la touche **A** et en appuyant sur la touche **D** :

```
[detached from 11827.mysession]
SLES12SP1:~ #
```

Créez maintenant un autre screen, cette fois-ci, non imbriqué :

```
SLES12SP1:~ # screen -S mysession1
```

Constatez le résultat :

```
SLES12SP1:~ # screen -ls
There are screens on:
    26655.mysession1      (Attached)
    11827.mysession (Detached)
2 Sockets in /var/run/screens/S-root.
```

Ratachez maintenant le screen **mysession** :

```
SLES12SP1:~ # screen -r 11827
```

Constatez de nouveau le résultat :

```
SLES12SP1:~ # sleep 9999 &
[1] 19879
SLES12SP1:~ # jobs
[1]+  Running                  sleep 9999 &
SLES12SP1:~ # jobs
[1]+  Running                  sleep 9999 &
SLES12SP1:~ # screen -ls
There are screens on:
    26655.mysession1          (Attached)
    11827.mysession (Attached)
2 Sockets in /var/run/screens/S-root.
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **screen** pour visualiser les options de la commande.

Options et Arguments

Les options sous Linux peuvent être exprimées au format court ou au format long. Plusieurs différences sont importantes à noter.

Premièrement les options courtes sont précédées par un simple tiré -, tandis que les options longues sont précédées par deux tirés --.

Un exemple est l'option de l'aide pour la plupart des commandes bash :

- -h
- -help

Deuxièmement les options courtes peuvent être combinées tandis que les options longues ne peuvent pas l'être. Par exemple, la ligne de commande **ls -l -a -i** peut être aussi écrite **ls -lai**, **ls -lia** ou encore **ls -ali** :

```
SLES12SP1:~ # ls -lai /tmp
total 8
256 drwxrwxrwt 1 root    root    802 Oct  2 01:00 .
256 drwxr-xr-x 1 root    root    156 May  1 14:55 ..
258 drwxrwxrwt 1 root    root      0 May  3 13:54 .ICE-unix
261 drwxrwxrwt 1 root    root      0 May  2 15:56 .Test-unix
257 drwxrwxrwt 1 root    root      0 May  3 13:53 .X11-unix
259 drwxrwxrwt 1 root    root      0 May  2 15:56 .XIM-unix
373 drwx----- 1 trainee users   12 May  3 13:54 .esd-1000
279 drwx----- 1 gdm      gdm      0 May  3 13:54 .esd-486
260 drwxrwxrwt 1 root    root      0 May  2 15:56 .font-unix
502 srw----- 1 root    root      0 Sep 30 10:26 .java_pid2399
462 srw----- 1 root    root      0 Aug 18 12:48 .java_pid2546
448 srw----- 1 root    root      0 May  3 14:01 .java_pid3281
403 drwxr-xr-x 1 root    root      8 Sep 30 10:26 hsperfdata_root
488 drwxr-xr-x 1 trainee users    48 Aug 18 13:14 inode
444 -rwxrwxrwx 1 root    root 50048 May  3 14:01 managera1411267841657715235client
439 -rwxrwxrwx 1 root    root 50048 May  3 14:01 managera3336001029897679475server
440 -rwxrwxrwx 1 root    root 50048 May  3 14:01 managera4847938942232964844client
443 -rwxrwxrwx 1 root    root 50048 May  3 14:01 managera5050357016347721452server
388 drwx----- 1 root    root      6 May  3 13:54 systemd-private-04f820fa26c745be8ddba814c6292f21-rtkit-
daemon.service-o4lKP5
427 -rw-r--r-- 1 root    root      0 May  3 14:01 systemicontmp5578677472245134133dat
426 drwxr-xr-x 1 root    root      0 May  3 14:01 systemicontmp7082392205020802884dat
SLES12SP1:~ # ls -ali /tmp
total 8
256 drwxrwxrwt 1 root    root    802 Oct  2 01:00 .
256 drwxr-xr-x 1 root    root    156 May  1 14:55 ..
```

```

258 drwxrwxrwt 1 root    root      0 May  3 13:54 .ICE-unix
261 drwxrwxrwt 1 root    root      0 May  2 15:56 .Test-unix
257 drwxrwxrwt 1 root    root      0 May  3 13:53 .X11-unix
259 drwxrwxrwt 1 root    root      0 May  2 15:56 .XIM-unix
373 drwx----- 1 trainee users    12 May  3 13:54 .esd-1000
279 drwx----- 1 gdm      gdm      0 May  3 13:54 .esd-486
260 drwxrwxrwt 1 root    root      0 May  2 15:56 .font-unix
502 srw----- 1 root    root      0 Sep 30 10:26 .java_pid2399
462 srw----- 1 root    root      0 Aug 18 12:48 .java_pid2546
448 srw----- 1 root    root      0 May  3 14:01 .java_pid3281
403 drwxr-xr-x 1 root    root      8 Sep 30 10:26 hsperfdata_root
488 drwxr-xr-x 1 trainee users    48 Aug 18 13:14 inode
444 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera1411267841657715235client
439 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera3336001029897679475server
440 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera4847938942232964844client
443 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera5050357016347721452server
388 drwx----- 1 root    root      6 May  3 13:54 systemd-private-04f820fa26c745be8ddba814c6292f21-rtkit-
daemon.service-o4lKP5
427 -rw-r--r-- 1 root    root      0 May  3 14:01 systemicontmp5578677472245134133dat
426 drwxr-xr-x 1 root    root      0 May  3 14:01 systemicontmp7082392205020802884dat

```

La commande **ls -l -all -inode** ne peut pas être écrite **ls -l -allinode** :

```

SLES12SP1:~ # ls -l --all --inode /tmp
total 8
256 drwxrwxrwt 1 root    root    802 Oct  2 01:00 .
256 drwxr-xr-x 1 root    root    156 May  1 14:55 ..
258 drwxrwxrwt 1 root    root      0 May  3 13:54 .ICE-unix
261 drwxrwxrwt 1 root    root      0 May  2 15:56 .Test-unix
257 drwxrwxrwt 1 root    root      0 May  3 13:53 .X11-unix
259 drwxrwxrwt 1 root    root      0 May  2 15:56 .XIM-unix
373 drwx----- 1 trainee users    12 May  3 13:54 .esd-1000
279 drwx----- 1 gdm      gdm      0 May  3 13:54 .esd-486
260 drwxrwxrwt 1 root    root      0 May  2 15:56 .font-unix

```

```
502 srw----- 1 root    root      0 Sep 30 10:26 .java_pid2399
462 srw----- 1 root    root      0 Aug 18 12:48 .java_pid2546
448 srw----- 1 root    root      0 May  3 14:01 .java_pid3281
403 drwxr-xr-x 1 root    root      8 Sep 30 10:26 hsperfdata_root
488 drwxr-xr-x 1 trainee users     48 Aug 18 13:14 inode
444 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera1411267841657715235client
439 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera3336001029897679475server
440 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera4847938942232964844client
443 -rwxrwxrwx 1 root    root    50048 May  3 14:01 managera5050357016347721452server
388 drwx----- 1 root    root      6 May  3 13:54 systemd-private-04f820fa26c745be8ddba814c6292f21-rtkit-
daemon.service-04lKP5
427 -rw-r--r-- 1 root    root      0 May  3 14:01 systemictmp5578677472245134133dat
426 drwxr-xr-x 1 root    root      0 May  3 14:01 systemictmp7082392205020802884dat
```

```
SLES12SP1:~ # ls -l --allinode /tmp
ls: unrecognized option '--allinode'
Try 'ls --help' for more information.
```



Important : Les options prenant un argument ne sont pas combinées avec les autres options.

Expressions Régulières

La manipulation de fichiers textes utilise des **expressions régulières**. Sous Linux il existe deux types d'expressions régulières :

- expressions régulières basiques - IEEE POSIX Basic Regular Expressions, appelées **ERb**,
 - utilisées par les commandes **vi**, **grep**, **expr** et **sed**,
- expressions régulières étendues - IEEE POSIX Extended Regular Expressions, appelées **ERe**,
 - utilisées par les commandes **egrep** (**grep -E**) et **awk**.

Les expressions régulières utilisent des caractères spéciaux. Certains caractères sont communs aux Erb et aux Ere :

Caractère spécial	Description
^	Trouver la chaîne au début de la ligne
\$	Trouver la chaîne à la fin de la ligne
\	Annuler l'effet spécial du caractère suivant
[]	Trouver n'importe quel des caractères entre les crochets
[^]	Exclure les caractères entre crochets
.	Trouver n'importe quel caractère sauf à la fin de la ligne
*	Trouver 0 ou plus du caractère qui précède
\<	Trouver la chaîne au début d'un mot
\>	Trouver la chaîne à la fin d'un mot

ERb

Certains caractères spéciaux sont spécifiques aux ERb :

Caractère spécial	Description
\{x,y\}	Trouver de x à y occurrences de ce qui précède
\{x\}	Trouver exactement le nombre x d'occurrences de ce qui précède
\{x,\}	Trouver le nombre x ou plus d'occurrences de ce qui précède
\(ERb)	Mémoriser une ERb
\1	Rappeler la première ERb mémorisée
\2, \3 ...	Rappeler la deuxième ERb mémorisée, rappeler la troisième ERb mémorisée etc

ERe

Certains caractères spéciaux sont spécifiques aux ERe :

Caractère spécial	Description
?	Trouver 0 ou 1 occurrence de ce qui précède

Caractère spécial	Description
+	Trouver 1 ou n d'occurrences de ce qui précède
{x,y}	Trouver de x à y occurrences de ce qui précède
{x}	Trouver exactement le nombre x d'occurrences de ce qui précède
{x,}	Trouver le nombre x ou plus d'occurrences de ce qui précède
()	Faire un ET des expressions régulières entre les parenthèses
	Faire un OU des expressions régulières se trouvant de chaque côté du pipe

Outils de Manipulation de Fichiers Texte

La commande grep

La commande grep peut être utilisée pour rechercher des lignes contenant une chaîne de caractères dans un jeu de fichiers.

Par défaut, la commande grep est sensible à la casse. Pour rendre cette commande insensible à la casse, il faut utiliser l'option **-i**.

La commande grep peut être aussi utilisée pour faire l'inverse, autrement dit de montrer les lignes qui ne contiennent pas la chaîne recherchée. Dans ce cas, il faut utiliser l'option **-v**.

La commande grep peut être utilisée avec des **Expressions Régulières basiques**. Ceci est utile pour rechercher dans le contenu de fichiers.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **grep** pour visualiser les options de la commande.

La Commande egrep

La commande **egrep** est identique à la commande **grep -E**. Dans les deux cas, l'utilisation des expressions régulières est étendue aux ERe.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **egrep** pour visualiser les options de la commande.

La Commande fgrep

La commande **fgrep** est identique à la commande **grep -F**. Dans les deux cas et par défaut la recherche concerne une chaîne de caractères interprétés dans un sens littéral sans utilisation de caractères spéciaux ni d'expressions régulières.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **fgrep** pour visualiser les options de la commande.

LAB #1 - Utiliser grep, egrep et fgrep

Créez le fichier **/tmp/greptest** :

greptest

```
fenestr0S
fenestros
555-5555
f
.fenestros
.fe
£
```

Recherchez maintenant toute ligne du fichier **/tmp/greptest** contenant au moins une lettre :

```
SLES12SP1:~ # grep '[a-zA-Z]' /tmp/greptest
fenestr0S
fenestros
f
.fenestros
.fe
```

Recherchez maintenant toute ligne contenant au moins une lettre ou un chiffre :

```
SLES12SP1:~ # grep '[a-zA-Z0-9]' /tmp/greptest
fenestr0S
fenestros
555-5555
f
.fenestros
.fe
```



Important : Notez la présence de la ligne 555-5555.

Recherchez maintenant toute ligne contenant un numéro de téléphone au format NNN-NNNN :

```
SLES12SP1:~ # grep '[0-9]\{3\}-[0-9]\{4\}' /tmp/greptest
555-5555
```

Recherchez maintenant toute ligne contenant exactement un caractère :

```
SLES12SP1:~ # grep '^.$' /tmp/greptest
f
£
```



Important : Notez l'utilisation des caractères spéciaux le début de ligne : ^, n'importe quel caractère : . et la fin de ligne : \$.

Recherchez maintenant toute ligne commençant par un point :

```
SLES12SP1:~ # grep '^\. ' /tmp/greptest
.fenestros
.fe
```



Important : Notez l'utilisation du caractère d'échappement \ pour annuler l'effet du caractère spécial .



La commande grep peut aussi être utilisée pour rechercher une chaîne dans tous les fichiers d'un répertoire spécifié : **grep -rnw 'directory' -e "pattern"**. Vous pouvez aussi spécifier les extensions des fichiers dans lesquels vous voulez rechercher la chaîne : **grep -include={*.doc,*.xls} -rnw 'directory' -e "pattern"**. Dernièrement vous pouvez



exclure des fichiers de la recherche de la façon suivante : **grep -exclude=*.doc -rnw 'directory' -e "pattern"**.

Modifiez le fichier **/tmp/greptest** :

greptest

```
# Starting comment
fenestr0S
fenestros
# Another comment
555-5555
f

.fenestros

.fe

£
# End comment
```

Utilisez maintenant la commande **grep** avec l'option **-E** pour supprimer les lignes de commentaires ainsi que les lignes vides :

```
SLES12SP1:~ # grep -E -v '^(#|$)' /tmp/greptest
fenestr0S
fenestros
555-5555
f
.fenestros
.fe
£
```



Important : Notez l'utilisation des parenthèses pour faire un regroupement ainsi que le pipe pour représenter un OU. L'expression '`^(#|$)`' indique donc "toute ligne commençant par le caractère `#`" OU "toute ligne où le début de la ligne est aussi la fin de la ligne".

Utilisez maintenant la commande **egrep** pour envoyer le contenu du fichier **/tmp/greptest**, sans commentaires et sans lignes vides, dans le fichier **/tmp/greptest1** :

```
SLES12SP1:~ # egrep -v '^(#|$)' /tmp/greptest > /tmp/greptest1
SLES12SP1:~ # cat /tmp/greptest1
fenestr0S
fenestros
555-5555
f
.fenestros
.fe
f
```



Important : Cette commande est particulièrement utile face à un fichier de configuration de plusieurs centaines de lignes dont certaines contiennent des directives activées d'autres sont vides ou en commentaires. De cette façon vous pouvez générer facilement un fichier ne contenant que les directives activées.

Modifiez le fichier **/tmp/greptest** ainsi :

greptest

```
# Starting comment
^ This line will be used to demonstrate the use of fgrep
```

```
fenestr0S
fenestros
# Another comment
555-5555
f

.fenestros

.fe

£
# End comment
```

Utilisez maintenant la commande **fgrep** pour rechercher la ligne commençant par le caractère ^ :

```
SLES12SP1:~ # fgrep '^' /tmp/greptest
^ This line will be used to demonstrate the use of fgrep
```

Comparez le résultat ci-dessus avec celui de la commande grep :

```
SLES12SP1:~ # grep '^' /tmp/greptest
# Starting comment
^ This line will be used to demonstrate the use of fgrep
fenestr0S
fenestros
# Another comment
555-5555
f

.fenestros

.fe
```

```
£  
# End comment
```

En effet, la ligne de commande en utilisant la commande `grep` devrait être :

```
SLES12SP1:~ # grep '^\' /tmp/greptest  
^ This line will be used to demonstrate the use of fgrep
```

Le Commande sed

La commande **sed** ou *Stream Editor* est un éditeur de texte non-interactif. Les actions spécifiées par la commande `sed` sont exécutées par défaut sur chaque ligne du fichier. La commande `sed` ne modifie pas le fichier d'origine et sa sortie standard est le canal 1.

Si plusieurs actions sont spécifiées dans la ligne de commande, chacune doit être précédée par l'option **-e**.

La syntaxe de la commande `sed` est la suivante :

```
sed [adresse] commande [arguments]
```

L'**adresse** permet de stipuler les lignes concernées par la **commande**.

La syntaxe d'une adresse peut être :

adresse	Lignes concernées
a	La ligne numéro a
\$	La dernière ligne
/ERb/	Les lignes qui correspondent à l'ERb
a,b	De la ligne numéro a jusqu'à la ligne numéro b
/ERb1/, /ERb2/	Toutes les lignes entre la première occurrence correspondant à l'ERb1 jusqu'à la première occurrence correspondant à l'ERb2

Les commandes de `sed` sont :

commande	Description
d	Ne pas afficher la ou les ligne(s)
p	Afficher la ou les ligne(s)
s	Effectuer une substitution
w	Ecrire le ou les ligne(s) dans un fichier
=	Afficher le numéro de la ligne spécifiée
!	Exécuter la commande ci-dessus sur toutes les lignes sauf celle spécifiées dans l'adresse

Options de la commande



A faire : Utilisez l'option **-help** de la commande **sed** pour visualiser les options de la commande.

LAB #2 - Utiliser la Commande sed

La commande **d** de sed permet de ne pas afficher certaines lignes à l'écran. Dans l'exemple qui suit, les 10 premières lignes du fichier **/etc/services** ne sont pas affichées à l'écran :

```
SLES12SP1:~ # sed '1,10d' /etc/services | more
# See also: services(5)
#
#
# SERVICE NAME AND TRANSPORT PROTOCOL PORT NUMBER REGISTRY
#
# (last updated on 2011-11-23)
#
# *****
# * Since we merged and converted this registry it loads more slowly. We *
# * know it should load faster and are working on ways to achieve that. *
```

```
# * Thank you for your patience. *
# *****
#
# Service names and port numbers are used to distinguish between different
# services that run over transport protocols such as TCP, UDP, DCCP, and
# SCTP.
#
# Service names are assigned on a first-come, first-served process, as
# documented in [RFC6335].
#
# Port numbers are assigned in various ways, based on three ranges: System
# Ports (0-1023), User Ports (1024-49151), and the Dynamic and/or Private
# Ports (49152-65535); the difference uses of these ranges is described in
--More--
```

Dans l'exemple qui suit, sed n'affiche pas de lignes de commentaires, c'est-à-dire les lignes commençant par le caractère # :

```
SLES12SP1:~ # sed '/^#/d' /etc/services | more
tcpmux      1/tcp      # TCP Port Service Multiplexer  [Mark_Lottor]
tcpmux      1/udp      # TCP Port Service Multiplexer  [Mark_Lottor]
compressnet 2/tcp      # Management Utility
compressnet 2/udp      # Management Utility
compressnet 3/tcp      # Compression Process  [Bernie_Volz]
compressnet 3/udp      # Compression Process  [Bernie_Volz]
rje         5/tcp      # Remote Job Entry  [Jon_Postel]
rje         5/udp      # Remote Job Entry  [Jon_Postel]
echo        7/tcp      # Echo  [Jon_Postel]
echo        7/udp      # Echo  [Jon_Postel]
discard     9/tcp      # Discard  [Jon_Postel]
discard     9/udp      # Discard  [Jon_Postel]
discard     9/sctp     # Discard  [Randall_Stewart] [RFC4960]
discard     9/dccp     # Discard  [Eddie_Kohler] [RFC4340]
systat      11/tcp     # Active Users  [Jon_Postel]
systat      11/udp     # Active Users  [Jon_Postel]
```

```
daytime      13/tcp      # Daytime [RFC867] [Jon_Postel]
daytime      13/udp      # Daytime [RFC867] [Jon_Postel]
qotd         17/tcp      # Quote of the Day [Jon_Postel]
qotd         17/udp      # Quote of the Day [Jon_Postel]
msp          18/tcp      # Message Send Protocol (historic) [Rina_Nethan
iel]
msp          18/udp      # Message Send Protocol (historic) [Rina_Nethan
--More--
```



Important : Notez que l'ERb est entourée des caractères / et /.

La commande sed vous permet d'afficher à l'écran certaines lignes spécifiées en utilisant la commande **p** :

```
SLES12SP1:~ # sed '1,2p' /etc/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftpsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
...
```



Important : Notez que sed affiche également tout le contenu du fichier. Ceci implique que les lignes 1 et 2 s'affichent deux fois.

Pour n'afficher que les lignes spécifiées, il convient d'utiliser l'option **-n** :

```
SLES12SP1:~ # sed -n '1,2p' /etc/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
```

La commande **w** permet d'écrire dans un fichier. Par exemple pour écrire dans le fichier **/tmp/sedtest** toutes les lignes du fichier **/etc/services** ne commençant pas par le caractère **#**, il convient d'utiliser la commande suivante :

```
SLES12SP1:~ # sed -n '/^#/!w /tmp/sedtest' /etc/services
SLES12SP1:~ # more /tmp/sedtest
tcpmux          1/tcp      # TCP Port Service Multiplexer  [Mark_Lottor]
tcpmux          1/udp      # TCP Port Service Multiplexer  [Mark_Lottor]
compressnet     2/tcp      # Management Utility
compressnet     2/udp      # Management Utility
compressnet     3/tcp      # Compression Process  [Bernie_Volz]
compressnet     3/udp      # Compression Process  [Bernie_Volz]
rje             5/tcp      # Remote Job Entry  [Jon_Postel]
rje             5/udp      # Remote Job Entry  [Jon_Postel]
echo            7/tcp      # Echo  [Jon_Postel]
echo            7/udp      # Echo  [Jon_Postel]
discard         9/tcp      # Discard  [Jon_Postel]
discard         9/udp      # Discard  [Jon_Postel]
discard         9/sctp     # Discard  [Randall_Stewart] [RFC4960]
discard         9/dccp     # Discard  [Eddie_Kohler] [RFC4340]
sysstat         11/tcp     # Active Users  [Jon_Postel]
sysstat         11/udp     # Active Users  [Jon_Postel]
daytime         13/tcp     # Daytime  [RFC867] [Jon_Postel]
daytime         13/udp     # Daytime  [RFC867] [Jon_Postel]
qotd            17/tcp     # Quote of the Day  [Jon_Postel]
qotd            17/udp     # Quote of the Day  [Jon_Postel]
msp            18/tcp     # Message Send Protocol (historic)  [Rina_Nethan
iel]
msp            18/udp     # Message Send Protocol (historic)  [Rina_Nethan
--More-- (0%)
```

La commande **s** permet de procéder à une substitution :

```
SLES12SP1:~ # echo "user1,user2,user3" > /tmp/sedtest1
SLES12SP1:~ # cat /tmp/sedtest1 | sed 's/,/ /g'
user1 user2 user3
```



Important : Notez que dans cet exemple, la commande **s** est suivi par un argument qui prend la forme /ce qui est à remplacer (caractère, chaîne ou ERb)/chaîne de remplacement/g. Le caractère **g** force le remplacement de toutes les occurrences. Sans elle, uniquement la première occurrence serait remplacée. Dans le cas de l'exemple, on remplace donc les virgules par des espaces.

La Commande **awk**

Présentation

Le processeur de texte **awk** est un **filtre**. Une **action** **awk** est fournie sur la ligne de commande entourée de ' ou de " :

```
awk [-F séparateur] 'critère {action}' [fichier1 ... fichiern]
```



Important : Le couple critère {action} s'appelle une clause.

Dans le cas de l'utilisation d'un **script** **awk**, la syntaxe de la commande devient :

```
awk [-F séparateur] -f script [fichier1 ... fichiern]
```

Découpage en champs

awk sait identifier les champs de la ligne soit parce que ceux-ci sont séparés par un espace ou par une tabulation soit parce que la ligne de commande lui a identifié le séparateur grâce à l'option **-F**.

awk stocke les informations de la ligne dans des variables :

Variable	Description
\$0	Contient toute la ligne
\$1, \$2 ...	Contient le premier champ de la ligne, contient le deuxième champ de la ligne ...

Par exemple :

```
SLES12SP1:~ # ls -l | awk '{print $8 $3 $4}'
```

```
06:55traineeusers
06:55traineeusers
06:55traineeusers
09:04traineeusers
15:41traineeusers
2016traineeusers
2016traineeusers
14:00traineeusers
15:59traineeusers
16:16traineeusers
16:16traineeusers
17:00traineeusers
16:14traineeusers
2016traineeusers
2016traineeusers
2016traineeusers
2016traineeusers
2016traineeusers
```

```
15:46traineeusers
03:59traineeusers
2016traineeusers
10:27traineeusers
06:55traineeusers
```

Comme vous pouvez constater, awk a extrait du résultat de la commande **ls -l** les champs **nom de l'élément**, **le propriétaire** et le **groupe**.

Afin de le rendre un peu plus lisible, saisissez la commande suivante :

```
SLES12SP1:~ # ls -l | awk '{print $8 " " $3 " " $4}'
06:55 trainee users
06:55 trainee users
06:55 trainee users
09:04 trainee users
15:41 trainee users
2016 trainee users
2016 trainee users
14:00 trainee users
15:59 trainee users
16:16 trainee users
16:16 trainee users
17:00 trainee users
16:14 trainee users
2016 trainee users
2016 trainee users
2016 trainee users
2016 trainee users
2016 trainee users
15:46 trainee users
03:59 trainee users
2016 trainee users
10:27 trainee users
06:55 trainee users
```

Critères

Les **critères** conditionnent l'exécution d'une **action** dans une **clause**.

Plusieurs types de critères sont possibles. Les plus utilisées sont les suivantes :

Une expression régulière valide pour la ligne

- Format:
- /expression régulière/ {instruction}
- Exemple:
- /ERe/ {print \$0}

Une expression régulière valide pour un champ

- Format:
- \$n ~/expression régulière/ {instruction}
- \$n!~/expression régulière/ {instruction}
- Exemple:
- \$1 ~/ERe/ {print \$0}
- \$1!~/ERe/ {print \$0}

awk sélectionne des lignes en utilisant un opérateur de correspondance ou de non-correspondance :

Opérateur	Condition
~	Correspondance
!~	Non-correspondance

Une comparaison

- Format:

- \$n opérateur critère de comparaison {action}
- Exemple:
- \$1 > 20 {print \$0}

Les opérateurs sont :

Opérateur	Condition
<	Inférieur
<=	Inférieur ou égal
==	Egal
!=	Différent
>	Supérieur
>=	Supérieur ou égal

Un opérateur logique

- Format:
- test1 opérateur logique test2 {action}
- Exemple:
- \$1 ~/ERe/ && \$2 > 20 {print \$0}

Les opérateurs sont :

Opérateur logique	Condition
	OU
&&	ET
!	NON

Une variable interne

- Format:
- expression1, expression2 {action}
- Exemple:

- NR==7, NR==10 {print \$0}

Les variables sont :

Variable	Description
NR	Nombre total de lignes
NF	Nombre total de champs
FILENAME	Le nom du fichier en entrée
FS	Le séparateur de champs en entrée. Par défaut un espace ou une tabulation
RS	Le séparateur de lignes en entrée. Par défaut une nouvelle ligne
OFS	Le séparateur de champs en sortie. Par défaut un espace
ORS	Le séparateur de lignes en sortie. Par défaut une nouvelle ligne
OFMT	Le format numérique. Par défaut "%.6g"

Scripts awk

Quand un programme awk comporte plusieurs **clauses** composées de **critères** et d'**actions**, il convient de d'écrire un **script awk**. Ce script comporte trois sections :

- La section **BEGIN**
 - Cette section est exécutée avant la lecture du script
- La section **principale**
 - Cette section contient les clauses
- La section **END**
 - Cette section est exécutée une fois à la fin du script

Par exemple :

```
SLES12SP1:~ # cat > awkscript
BEGIN {
    print "List of the currently mounted file systems"}
{print $0}
END {
```

```
print "====="}  
[ ^D]
```



Important : Dans l'exemple ci-dessus, la ligne [^D] indique que vous devez appuyer simultanément sur les touches **CTRL** et **D**.

Ensuite saisissez la commande suivante :

```
SLES12SP1:~ # awk -f awkscript /etc/fstab  
List of the currently mounted file systems  
UUID=9e0e4497-5e47-419a-8ff4-9e2dd7d6a11b swap swap defaults 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 / btrfs defaults 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /boot/grub2/i386-pc btrfs subvol=@/boot/grub2/i386-pc 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /boot/grub2/x86_64-efi btrfs subvol=@/boot/grub2/x86_64-efi 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /home btrfs subvol=@/home 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /opt btrfs subvol=@/opt 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /srv btrfs subvol=@/srv 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /tmp btrfs subvol=@/tmp 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /usr/local btrfs subvol=@/usr/local 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/crash btrfs subvol=@/var/crash 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/libvirt/images btrfs subvol=@/var/lib/libvirt/images 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/mailman btrfs subvol=@/var/lib/mailman 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/mariadb btrfs subvol=@/var/lib/mariadb 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/mysql btrfs subvol=@/var/lib/mysql 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/named btrfs subvol=@/var/lib/named 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/pgsql btrfs subvol=@/var/lib/pgsql 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/log btrfs subvol=@/var/log 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/opt btrfs subvol=@/var/opt 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/spool btrfs subvol=@/var/spool 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/tmp btrfs subvol=@/var/tmp 0 0  
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /.snapshots btrfs subvol=@/.snapshots 0 0
```



Important : Notez l'utilisation de l'option -f qui applique le script awk au fichier donné en argument.

La Fonction printf

La fonction intégrée **printf** permet de formater des affichages. Elle a la syntaxe suivante :

```
printf ("chaîne",expression1,expression2,...,expressionn)
```

chaîne contient autant de formats qu'il y a d'expressions.

Les formats de printf sont, par exemple :

Format	Description
%30s	Affichage d'une chaîne (s=string) sur 30 positions avec cadrage à droite
%-30s	Affichage d'une chaîne (s=string) sur 30 positions avec cadrage à gauche
%4d	Affichage d'un entier sur 4 positions avec cadrage à droite
%-4d	Affichage d'un entier sur 4 positions avec cadrage à gauche

Structures de Contrôle

awk peut utiliser des structures de contrôle.

if

La syntaxe de la commande if est la suivante :

```
if condition {  
    commande  
    commande  
    ...  
}  
  
else {  
    commande  
    commande  
    ...  
}
```

ou dans le cas d'une seule commande :

```
if condition  
    commande  
  
else  
    commande
```

for

La syntaxe de la structure de contrôle **for** est la suivante :

```
for variable in liste_variables {
```

```
    commande
    commande
    ...
}
```

ou dans le cas d'une seule commande :

```
for variable in liste_variables
    commande
```

ou dans le cas d'un tableau :

```
for clef dans tableau {
    print clef , tableau[clef]
}
```

while

La syntaxe de la structure de contrôle **while** est la suivante :

```
while condition {

    commande
    commande
    ...

}
```

do-while

La syntaxe de la structure de contrôle **do-while** est la suivante :

```
do {  
    commande  
    commande  
    ...  
} while condition
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **awk** pour visualiser les options de la commande.

LAB #3 - Utiliser la Commande awk

Pour illustrer l'utilisation des tableaux, créez le fichier **/tmp/sales.txt** :

sales

```
# Fenestr0s.com  
# Annual sales by French department  
# 83  
Desktops$100  
Portables$50
```

```
Servers$21
Ipads$4

# 06
Desktops$99
Portables$60
Servers$8
Ipads$16

# 13
Desktops$130
Portables$65
Servers$12
Ipads$56
```

Ce fichier contient des statistiques de vente par type de PC et par département.

Créez maintenant le script awk **/tmp/sales.awk** :

[sales.awk](#)

```
# BEGIN
BEGIN {
    FS="$"
}
# TABLE
$1 !~ /^#/ && $1 !~ /^$/ {
    sales[$1]+=$2
}
# END
END {
    for (pc in sales)
        printf("PC Type :  %s \t Sales (06+13+83) : %10d\n",pc,sales[pc]);
}
```



```
}
```

Ce script comporte 13 lignes et a pour but de calculer le nombre total de PC vendus dans les trois départements cités dans le fichier **sales.txt** :

```
1  # BEGIN
2  BEGIN {
3      FS="$"
4  }
5  # TABLE
6  $1 !~ /^#/ && $1 !~ /^$/ {
7      sales[$1]+=$2
8  }
9  # END
10 END {
11     for (pc in sales)
12         printf("PC Type :  %s \t Sales (06+13+83) : %10d\n",pc,sales[pc]);
13 }
```

Dans ce script vous noterez :

- La ligne **3**,
 - Cette ligne se trouve dans la section **BEGIN**. Elle spécifie le séparateur de champs.
- La ligne **6**,
 - Cette ligne évite le traitement de toute ligne commençant par le caractère **#** ainsi que toute ligne vide.
- La ligne **7**,
 - Ce tableau a pour clef la valeur de **\$1**, c'est-à-dire, les noms des différents types de PC. Les valeurs du tableau sont le nombre de PC vendus, ici représenté par **\$2**. Les caractères **+=** indique qu'à chaque traitement de ligne, le nombre de PC vendus sur la ligne doit être rajouté à la valeur déjà présente dans le tableau.
- La ligne **11**,
 - Cette ligne démarre une boucle **for**.⁷
- La ligne **12**,
 - Cette ligne utilise **printf** afin d'imprimer à l'écran les valeurs calculées et stockées dans le tableau.

Appliquez maintenant votre script awk au fichier **ventes_materiel.txt** :

```
SLES12SP1:~ # awk -f /tmp/sales.awk /tmp/sales.txt
PC Type :  Portables      Sales (06+13+83) :      175
PC Type :   Ipads         Sales (06+13+83) :       76
PC Type :  Desktops      Sales (06+13+83) :     329
PC Type :   Servers      Sales (06+13+83) :       41
```

Autres Commandes Utiles

La Commande expand

La commande **expand** convertit des tabulations dans un fichier en espaces et envoie le résultat à la sortie standard. Sans fichier en argument ou avec le caractère -, la commande prend son entrée de l'entrée standard.

Créez le fichier **/root/expand** :

expand

```
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
un  deux    trois   quatre   cinq
```

Visualisez le fichier avec la commande cat :

```
SLES12SP1:~ # cat expand
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
un  deux    trois    quatre    cinq
```

Utilisez les option **-vet** de la commande cat pour visualiser les caractères invisibles :

```
SLES12SP1:~ # cat -vet expand
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
```



Important : Comme vous pouvez constater, les tabulations sont représentées par ^I et les fins de lignes par \$.

Utilisez maintenant la commande **expand** pour convertir les tabulations en espaces en envoyant le résultat dans le fichier **expand1** :

```
SLES12SP1:~ # expand expand > expand1
```

Visualisez le fichier avec la commande `cat` et les options **-vet** :

```
SLES12SP1:~ # cat -vet expand1
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
```



Important : Comme vous pouvez constater, les tabulations ont été converties en espaces.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **expand** pour visualiser les options de la commande.

La Commande `unexpand`

La commande **unexpand** convertit des espaces dans un fichier en tabulations et envoie le résultat à la sortie standard. Sans fichier en argument ou

avec le caractère -, la commande prend son entrée de l'entrée standard.

Utilisez la commande **unexpand** sur le fichier **expand1** et envoyez le résultat dans le fichier **expand2** :

```
SLES12SP1:~ # cat -vet expand1
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
un      deux      trois      quatre      cinq$
SLES12SP1:~ # unexpand -a expand1 > expand2
SLES12SP1:~ # cat -vet expand2
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
un^Ideux^Itrois^Iquatre^Icinq$
```



Important : Notez que les espaces ont été remplacés par des tabulations.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **unexpand** pour visualiser les options de la commande.

La Commande cut

Chaque ligne est divisée en colonnes. Dans une ligne le premier caractère est dans la colonne numéro **un**, le deuxième dans la colonne deux et ainsi de suite. Dans une ligne il peut y avoir des champs séparés par des tabulations.

La commande **cut** permet de sélectionner des colonnes et des champs dans un fichier. La commande permet aussi d'utiliser une critère de séparation de champs autre que la tabulation en spécifiant cette critère en utilisant l'option **-d**.

Par exemple, pour sélectionner les 7 premières colonnes du fichier **/etc/passwd** la commande est :

```
SLES12SP1:~ # cut -c1-7 /etc/passwd
at:x:25
bin:x:1
daemon:
ftp:x:4
ftpsecu
games:x
gdm:x:4
lp:x:4:
mail:x:
man:x:1
message
news:x:
nobody:
nscd:x:
```

```
ntp:x:7
openslp
polkitd
postfix
pulse:x
root:x:
rpc:x:4
rtkit:x
scard:x
sshd:x:
statd:x
usbmux:
uucp:x:
vnc:x:4
wwwrun:
trainee
```

Pour sélectionner les colonnes 1 à 5, les colonnes 10 à 15 et les colonnes 30 et après, il convient d'utiliser la commande suivante :

```
SLES12SP1:~ # cut -c1-5,10-15,30- /etc/passwd
at:x:5:Batc/var/spool/atjobs:/bin/bash
bin:x:bin:/
daemon:2:Dan/bash
ftp:x:49:FTPftp:/bin/bash
ftpse:x:488FTP User:/var/lib/empty:/bin/false
games2:100:/var/games:/bin/bash
gdm:x:485:Ganager daemon:/var/lib/gdm:/bin/false
lp:x:Printi/spool/lpd:/bin/bash
mail:12:Mair/spool/clientmqueue:/bin/false
man:x:62:Maner:/var/cache/man:/bin/bash
messas:x:49 D-Bus:/var/run/dbus:/bin/false
news:13:Newnews:/bin/bash
nobod65534:var/lib/nobody:/bin/bash
nscd:6:495:/run/nscd:/sbin/nologin
```

```
ntp:x492:NTlib/ntp:/bin/false
opens:494:2n:/var/lib/empty:/sbin/nologin
polki:497:4lkitd:/var/lib/polkit:/sbin/nologin
postf:51:5ln:/var/spool/postfix:/bin/false
pulse90:489emon:/var/lib/pulseaudio:/sbin/nologin
root:0:rootsh
rpc:x:65534ind:/var/lib/empty:/sbin/nologin
rtkit91:490proc:/bin/false
scard87:487ader:/var/run/pcscd:/usr/sbin/nologin
sshd:8:498:r/lib/sshd:/bin/false
statd89:655aemon:/var/lib/nfs:/sbin/nologin
usbmu493:65emon:/var/lib/usbmuxd:/sbin/nologin
uucp::14:Uny system:/etc/uucp:/bin/bash
vnc:x:491:uar/lib/empty:/sbin/nologin
wwwru30:8:Whe:/var/lib/wwwrun:/bin/false
train:1000:ome/trainee:/bin/bash
```

Pour sélectionner les champs 2, 4 et 6 du fichier, il convient d'utiliser la commande suivante :

```
SLES12SP1:~ # cut -d: -f2,4,6 /etc/passwd
x:25:/var/spool/atjobs
x:1:/bin
x:2:/sbin
x:49:/srv/ftp
x:65534:/var/lib/empty
x:100:/var/games
x:485:/var/lib/gdm
x:7:/var/spool/lpd
x:12:/var/spool/clientmqueue
x:62:/var/cache/man
x:499:/var/run/dbus
x:13:/etc/news
x:65533:/var/lib/nobody
x:495:/run/nsd
```



```
x:492:/var/lib/ntp
x:2:/var/lib/empty
x:496:/var/lib/polkit
x:51:/var/spool/postfix
x:489:/var/lib/pulseaudio
x:0:/root
x:65534:/var/lib/empty
x:490:/proc
x:487:/var/run/pcscd
x:498:/var/lib/sshhd
x:65534:/var/lib/nfs
x:65534:/var/lib/usbmuxd
x:14:/etc/uucp
x:491:/var/lib/empty
x:8:/var/lib/wwwrun
x:100:/home/trainee
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **cut** pour visualiser les options de la commande.

La Commande uniq

La commande suivante permet d'extraire du fichier /etc/passwd les GID utilisés en tant que groupes principaux des utilisateurs :

```
SLES12SP1:~ # cut -d: -f4 /etc/passwd | sort -n | uniq
0
1
```

```
2
7
8
12
13
14
25
49
51
62
100
485
487
489
490
491
492
495
496
498
499
65533
65534
```



Important : Notez l'utilisation de la commande `uniq` qui permet de supprimer les doublons dans la sortie triée.

Options de la commande





A faire : Utilisez l'option **-help** de la commande **uniq** pour visualiser les options de la commande.

La Commande tr

La commande **tr** permet de substituer des caractères pour d'autres. Cette commande n'accepte que des données en provenance de son entrée standard et non en provenance d'un fichier.

```
SLES12SP1:~ # cat /etc/passwd | tr "[a-z]" "[A-Z]"
AT:X:25:25:BATCH JOBS DAEMON:/VAR/SPool/ATJOBS:/BIN/BASH
BIN:X:1:1:BIN:/BIN:/BIN/BASH
DAEMON:X:2:2:DAEMON:/SBIN:/BIN/BASH
FTP:X:40:49:FTP ACCOUNT:/SRV/FTP:/BIN/BASH
FTPSECURE:X:488:65534:SECURE FTP USER:/VAR/LIB/EMPTY:/BIN/FALSE
GAMES:X:12:100:GAMES ACCOUNT:/VAR/GAMES:/BIN/BASH
GDM:X:486:485:GNOME DISPLAY MANAGER DAEMON:/VAR/LIB/GDM:/BIN/FALSE
LP:X:4:7:PRINTING DAEMON:/VAR/SPool/LPD:/BIN/BASH
MAIL:X:8:12:MAILER DAEMON:/VAR/SPool/CLIENTMQUEUE:/BIN/FALSE
MAN:X:13:62:MANUAL PAGES VIEWER:/VAR/CACHE/MAN:/BIN/BASH
MESSAGEBUS:X:499:499:USER FOR D-BUS:/VAR/RUN/DBUS:/BIN/FALSE
NEWS:X:9:13:NEWS SYSTEM:/ETC/NEWS:/BIN/BASH
NOBODY:X:65534:65533:NOBODY:/VAR/LIB/NOBODY:/BIN/BASH
NSCD:X:496:495:USER FOR NSCD:/RUN/NSCD:/SBIN/NOLOGIN
NTP:X:74:492:NTP DAEMON:/VAR/LIB/NTP:/BIN/FALSE
OPENSLLP:X:494:2:OPENSLLP DAEMON:/VAR/LIB/EMPTY:/SBIN/NOLOGIN
POLKITD:X:497:496:USER FOR POLKITD:/VAR/LIB/POLKIT:/SBIN/NOLOGIN
POSTFIX:X:51:51:POSTFIX DAEMON:/VAR/SPool/POSTFIX:/BIN/FALSE
PULSE:X:490:489:PULSEAUDIO DAEMON:/VAR/LIB/PULSEAUDIO:/SBIN/NOLOGIN
ROOT:X:0:0:ROOT:/ROOT:/BIN/BASH
RPC:X:495:65534:USER FOR RPCBIND:/VAR/LIB/EMPTY:/SBIN/NOLOGIN
RTKIT:X:491:490:REALTIMEKIT:/PROC:/BIN/FALSE
```

```
SCARD:X:487:487:SMART CARD READER:/VAR/RUN/PCSCD:/USR/SBIN/NOLOGIN
SSHD:X:498:498:SSH DAEMON:/VAR/LIB/SSHD:/BIN/FALSE
STATD:X:489:65534:NFS STATD DAEMON:/VAR/LIB/NFS:/SBIN/NOLOGIN
USBMUX:X:493:65534:USBMUXD DAEMON:/VAR/LIB/USBMUXD:/SBIN/NOLOGIN
UUCP:X:10:14:UNIX-TO-UNIX COPY SYSTEM:/ETC/UUCP:/BIN/BASH
VNC:X:492:491:USER FOR VNC:/VAR/LIB/EMPTY:/SBIN/NOLOGIN
WWWRUN:X:30:8:WWW DAEMON APACHE:/VAR/LIB/WWWRUN:/BIN/FALSE
TRAINEE:X:1000:100:TRAINEE:/HOME/TRAINEE:/BIN/BASH
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **tr** pour visualiser les options de la commande.

La Commande paste

La commande **paste** concatène les lignes de n fichiers. Par exemple :

```
SLES12SP1:~ # paste -d: /etc/passwd /etc/shadow
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash:at:!:16922:~~~~~
bin:x:1:1:bin:/bin:/bin/bash:bin:!:16765:~~~~~
daemon:x:2:2:Daemon:/sbin:/bin/bash:daemon:!:16765:~~~~~
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash:ftp:!:16765:~~~~~
ftppsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false:ftppsecure:!:16922:~~~~~
games:x:12:100:Games account:/var/games:/bin/bash:games:!:16765:~~~~~
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false:gdm:!:16922:~~~~~
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash:lp:!:16765:~~~~~
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false:mail:!:16765:~~~~~
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash:man:!:16765:~~~~~
```

```
messagebus:x:499:499:User for D-Bus:/var/run/dbus:/bin/false:messagebus:!:16765:~::~:
news:x:9:13:News system:/etc/news:/bin/bash:news:!:16765:~::~:
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash:nobody:!:16765:~::~:
nscd:x:496:495:User for nscd:/run/nscd:/sbin/nologin:nscd:!:16765:~::~:
ntp:x:74:492:NTP daemon:/var/lib/ntp:/bin/false:ntp:!:16922:~::~:
openslp:x:494:2:openslp daemon:/var/lib/empty:/sbin/nologin:openslp:!:16765:~::~:
polkitd:x:497:496:User for polkitd:/var/lib/polkit:/sbin/nologin:polkitd:!:16765:~::~:
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false:postfix:!:16922:~::~:
pulse:x:490:489:PulseAudio daemon:/var/lib/pulseaudio:/sbin/nologin:pulse:!:16922:~::~:
root:x:0:0:root:/root:/bin/bash:root:$6$g0tHJ9vyIfFt$rbm.rf7p6XZMxMqbqa/BGDeA7E7RkC9n89w8cWdpAxxUmwk7BPcMv7Zy9nVAn7f/7zQJzcRcsIqp5bRxle8iX/:16922:~::~:
rpc:x:495:65534:user for rpcbind:/var/lib/empty:/sbin/nologin:rpc:!:16765:~::~:
rtkit:x:491:490:RealtimeKit:/proc:/bin/false:rtkit:!:16922:~::~:
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin:scard:!:16922:~::~:
sshd:x:498:498:SSH daemon:/var/lib/ssh:/bin/false:sshd:!:16765:~::~:
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin:statd:!:16922:~::~:
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin:usbmux:!:16922:~::~:
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash:uucp:!:16765:~::~:
vnc:x:492:491:user for VNC:/var/lib/empty:/sbin/nologin:vnc:!:16922:~::~:
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false:wwwrun:!:16765:~::~:
trainee:x:1000:100:trainee:/home/trainee:/bin/bash:trainee:$6$0ZyVqj4ekgm$Cw0T.n6gNv.vTdAT6dFxrrSeHW/V3r43jWFczPG0lxg5SB9iMUcQ6MFLz9NuTTas289xe/ULsJhE2HdJbraGA.:16922:0:99999:7::~:
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **paste** pour visualiser les options de la commande.

La Commande **split**

La commande **split** est utilisée pour découper de grands fichiers en petit morceaux d'une taille fixe ou d'un nombre de lignes fixe.

Créez d'abord un fichier d'une taille de 250Mo :

```
SLES12SP1:~ # dd if=/dev/zero of=/file bs=1024k count=250
250+0 records in
250+0 records out
262144000 bytes (262 MB) copied, 3.50609 s, 74.8 MB/s
```

Utilisez maintenant la commande **split** pour diviser ce fichier en morceaux de 50 Mo :

```
SLES12SP1:~ # split -b 50m /file filepart
SLES12SP1:~ # ls -l | grep filepart
-rw-r--r-- 1 root    root  52428800 Dec 14 15:10 filepartaa
-rw-r--r-- 1 root    root  52428800 Dec 14 15:10 filepartab
-rw-r--r-- 1 root    root  52428800 Dec 14 15:10 filepartac
-rw-r--r-- 1 root    root  52428800 Dec 14 15:10 filepartad
-rw-r--r-- 1 root    root  52428800 Dec 14 15:10 filepartae
```



Important : Notez que cinq morceaux ont été créés dans le répertoire courant. Si aucune taille n'est spécifiée, split devise le fichier en morceaux de 1 000 lignes par défaut.

Reconstruisez simplement le fichier avec la commande cat :

```
SLES12SP1:~ # cat fileparta* > newfile
SLES12SP1:~ # ls -l | grep newf
-rw-r--r-- 1 root    root  262144000 Dec 14 15:11 newfile
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **split** pour visualiser les options de la commande.

La Commande diff

La commande **diff** indique les modifications à apporter à deux fichiers pour que ceux-ci soient identique.

Pour commencer, copiez le fichier **/etc/passwd** vers le répertoire **/root** :

```
SLES12SP1:~ # cp /etc/passwd /root
```

Modifiez ensuite le fichier la ligne **trainee** du fichier **/root/passwd** ainsi :

```
...  
trainee10:x:1000:1000:trainee:/home/trainee:/bin/bash  
...
```

Supprimez la ligne **uucp** dans le fichier **/root/passwd** et ajoutez en fin de fichier la ligne suivante :

```
...  
Linux est super!
```

Comparez maintenant les deux fichiers :

```
SLES12SP1:~ # diff /etc/passwd /root/passwd  
27d26  
< uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
```

```
30c29,30
< trainee:x:1000:100:trainee:/home/trainee:/bin/bash
---
> trainee10:x:1000:100:trainee:/home/trainee:/bin/bash
> Linux est super
```

Dans cette sortie on constate le caractère **<** et le caractère **>**. Le premier indique le premier fichier qui a suivi la commande **diff** tandis que le deuxième indique le deuxième fichier.

Le message **27d26** indique que la ligne 27 dans `/etc/passwd` a besoin d'être supprimée car elle n'existe pas dans le fichier `/root/passwd`.

Le message **30c29,30** indique qu'il faut changer la ligne 30 dans `/etc/passwd` afin que celle-ci corresponde à la ligne 29 dans `/root/passwd`.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **diff** pour visualiser les options de la commande.

La Commande cmp

La commande **cmp** compare les fichiers caractère par caractère. Par défaut la commande s'arrête à la première différence rencontrée :

```
SLES12SP1:~ # cmp /root/passwd /etc/passwd
/root/passwd /etc/passwd differ: char 1413, line 27
```

L'option **-l** de la commande indique toutes les différences en trois colonnes :

```
SLES12SP1:~ # cmp -l /root/passwd /etc/passwd | more
cmp: EOF on /root/passwd
```



```
1413 166 165
1414 156 165
1416 72 160
1417 170 72
1418 72 170
1419 64 72
1420 71 61
1421 62 60
1423 64 61
1424 71 64
1425 61 72
1426 72 125
1427 165 156
1428 163 151
1429 145 170
1430 162 55
1431 40 164
1432 146 157
1433 157 55
1434 162 125
1435 40 156
1436 126 151
1437 116 170
--More--
```

La première colonne représente le numéro de caractère, la deuxième la valeur octale ASCII du caractère dans le fichier `/root/passwd` et la troisième la valeur octale ASCII du caractère dans le fichier `/etc/passwd`.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **cmp** pour visualiser les options de la commande.

La commande patch

La commande **patch** est utilisée pour appliquer des modifications à un fichier à partir d'un fichier patch qui contient les différences entre le contenu de l'ancienne version du fichier et la nouvelle version.

Rappelez-vous maintenant des modifications apportées au fichier /tmp/greptest :

```
SLES12SP1:~ # cat /tmp/greptest
# Starting comment
^ This line will be used to demonstrate the use of fgrep
fenestr0S
fenestros
# Another comment
555-5555
f

.fenestros

.fe

£
# End comment
```

Vous avez aussi supprimé tous les commentaires et toutes les lignes vides en sauvegardant le résultat en tant que **/tmp/greptest1** :

```
SLES12SP1:~ # cat /tmp/greptest1
fenestr0S
fenestros
555-5555
f
.fenestros
.fe
£
```

Afin de créer un fichier de patch, il convient d'utiliser la commande **diff** avec l'option **-u**

```
SLES12SP1:~ # cd /tmp
SLES12SP1:/tmp # diff -u greptest greptest1 > greptest.patch
```

L'examen du fichier de patch démontre les modifications à apporter au fichier **greptest** :

```
SLES12SP1:/tmp # cat /tmp/greptest.patch
--- greptest      2016-12-14 14:00:32.987471124 +0100
+++ greptest1     2016-12-14 13:59:12.319471124 +0100
@@ -1,14 +1,7 @@
-# Starting comment
-^ This line will be used to demonstrate the use of fgrep
 fenestr0S
 fenestros
-# Another comment
 555-5555
 f
-
 .fenestros
-
 .fe
-
 f
-# End comment
```

Procédez maintenant à l'application du fichier patch :

```
SLES12SP1:/tmp # patch < greptest.patch
patching file greptest
```

Contrôlez maintenant le contenu du fichier **greptest** :

```
SLES12SP1:/tmp # cat greptest
```

```
fenestr0S  
fenestros  
555-5555  
f  
.fenestros  
.fe  
£
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **patch** pour visualiser les options de la commande.

La commande strings

La commande **strings** est utilisée pour trouver toutes les chaînes de caractères qui peuvent être imprimés dans un ou plusieurs fichiers objets ou exécutables passés en argument. Un fichier objet est un fichier intermédiaire intervenant dans le processus de compilation.

Sous Linux et Unix, le format d'un fichier objet est le format **ELF**, (*Executable and Linkable Format*). Ce format est aussi utilisé pour :

- les exécutables,
- les bibliothèques partagés,
- les core dumps.

Sans option, la commande **strings** trouve toutes les chaînes d'une longueur de 4 caractères ou plus suivies par un caractère non-imprimable :

```
SLES12SP1:/tmp # strings /usr/bin/passwd | more  
/lib64/ld-linux-x86-64.so.2  
cHc`
```

```
libpam.so.0
_ITM_deregisterTMCloneTable
__gmon_start__
_Jv_RegisterClasses
_ITM_registerTMCloneTable
pam_start
pam_strerror
pam_chauthtok
pam_end
libpam_misc.so.0
misc_conv
libselinux.so.1
_init
is_selinux_enabled
security_getenforce
context_user_get
security_compute_av
matchpathcon
freecon
context_free
setfscreatecon
--More--
```

L'option **-t** de la commande retourne, en plus des chaînes concernées, la position de décalage pour chaque ligne sur laquelle une ou plusieurs chaînes se trouvent :

```
SLES12SP1:/tmp # strings -t d /usr/bin/passwd | more
 568 /lib64/ld-linux-x86-64.so.2
 650 cHc`
4977 libpam.so.0
4989 _ITM_deregisterTMCloneTable
5017 __gmon_start__
5032 _Jv_RegisterClasses
5052 _ITM_registerTMCloneTable
```

```
5078 pam_start
5088 pam_strerror
5101 pam_chauthtok
5115 pam_end
5123 libpam_misc.so.0
5140 misc_conv
5150 libselinux.so.1
5166 _init
5172 is_selinux_enabled
5191 security_getenforce
5211 context_user_get
5228 security_compute_av
5248 matchpathcon
5261 freecon
5269 context_free
5282 setfscreatecon
--More--
```

L'option **-t** prend un de trois arguments qui indique le système de numérotation à utiliser :

Argument	Système de Numérotation
d	Décimal
o	Octal
x	Héxadécimal

L'option **-n** de la commande permet de modifier le nombre de caractères minimales dans les chaînes recherchées :

```
SLES12SP1:/tmp # strings -t d -n 15 /usr/bin/passwd | more
 568 /lib64/ld-linux-x86-64.so.2
4989 _ITM_deregisterTMCloneTable
5032 _Jv_RegisterClasses
5052 _ITM_registerTMCloneTable
5123 libpam_misc.so.0
5150 libselinux.so.1
```

```
5172 is_selinux_enabled
5191 security_getenforce
5211 context_user_get
5228 security_compute_av
5510 __stack_chk_fail
5671 __errno_location
6106 __libc_start_main
6202 LIBPAM_MISC_1.0
34104 Usage: %s [options] [LOGIN]
34144  -a, --all          report password status on all accounts
34216  -d, --delete      delete the password for the named account
34296  -e, --expire      force expire the password for the named account
34384  -h, --help        display this help message and exit
34456  -k, --keep-tokens  change password only if expired
34528  -i, --inactive INACTIVE  set password inactive after expiration
--More--
```

Dans le cas de l'utilisation de la commande avec plus d'un fichier, l'option **-f** devient très utile. Par exemple, imaginons que vous souhaitez connaître les détails disponibles des Copyright des fichiers dans /bin :

```
SLES12SP1:/tmp # strings -f /bin/* | grep "(c)"
/bin/btrace: # Copyright (c) 2005 Silicon Graphics, Inc.
/bin/csh: @(#) Copyright (c) 1991 The Regents of the University of California.
/bin/ping: @(#) Copyright (c) 1989 The Regents of the University of California.
/bin/ping6: @(#) Copyright (c) 1989 The Regents of the University of California.
/bin/tcsh: @(#) Copyright (c) 1991 The Regents of the University of California.
```



Important : Notez que l'option **-f** a pour conséquence d'imprimer le nom du fichier contenant la chaîne au début de chaque ligne.

Options de la commande



A faire : Utilisez l'option **-help** de la commande **strings** pour visualiser les options de la commande.

La commande comm

La commande **comm** est utilisée pour comparer deux fichiers texte. La sortie de la commande sépare les lignes en trois catégories :

- Les lignes présentes seulement dans le premier fichier,
- Les lignes présentes seulement dans le deuxième fichier,
- Les lignes présentes dans la deux fichiers.

Utilisez la commande **comm** pour comparer les fichiers **/etc/passwd** et **/root/passwd** :

```
SLES12SP1:/tmp # comm /etc/passwd /root/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftpsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
messagebus:x:499:499:User for D-Bus:/var/run/dbus:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
nscd:x:496:495:User for nscd:/run/nscd:/sbin/nologin
```



```
ntp:x:74:492:NTP daemon:/var/lib/ntp:/bin/false
openslp:x:494:2:openslp daemon:/var/lib/empty:/sbin/nologin
polkitd:x:497:496:User for polkitd:/var/lib/polkit:/sbin/nologin
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
pulse:x:490:489:PulseAudio daemon:/var/lib/pulseaudio:/sbin/nologin
root:x:0:0:root:/root:/bin/bash
rpc:x:495:65534:user for rpcbind:/var/lib/empty:/sbin/nologin
rtkit:x:491:490:RealtimeKit:/proc:/bin/false
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin
sshd:x:498:498:SSH daemon:/var/lib/sshd:/bin/false
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
vnc:x:492:491:user for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
comm: file 1 is not in sorted order
comm: file 2 is not in sorted order
trainee10:x:1000:100:trainee:/home/trainee:/bin/bash
Linux est super
trainee:x:1000:100:trainee:/home/trainee:/bin/bash
```

Pour afficher uniquement les lignes présentes dans les deux fichiers, il convient d'utiliser les options **-1** et **-2** :

```
SLES12SP1:/tmp # comm -12 /etc/passwd /root/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftpssecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
```

```
messagebus:x:499:499:User for D-Bus:/var/run/dbus:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
nscd:x:496:495:User for nscd:/run/nscd:/sbin/nologin
ntp:x:74:492:NTP daemon:/var/lib/ntp:/bin/false
openslp:x:494:2:openslp daemon:/var/lib/empty:/sbin/nologin
polkitd:x:497:496:User for polkitd:/var/lib/polkit:/sbin/nologin
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
pulse:x:490:489:PulseAudio daemon:/var/lib/pulseaudio:/sbin/nologin
root:x:0:0:root:/root:/bin/bash
rpc:x:495:65534:user for rpcbind:/var/lib/empty:/sbin/nologin
rtkit:x:491:490:RealtimeKit:/proc:/bin/false
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin
sshd:x:498:498:SSH daemon:/var/lib/ssh:/bin/false
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin
vnc:x:492:491:user for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
comm: file 1 is not in sorted order
comm: file 2 is not in sorted order
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **comm** pour visualiser les options de la commande.

La commande head

La commande **head** permet d'afficher les **x** premières lignes d'un fichier. Sans options, la valeur de **x** est de 10 par défaut :

```
SLES12SP1:/tmp # head /etc/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftppsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
```

Avec l'option **-n**, la valeur de **x** peut être spécifiée :

```
SLES12SP1:/tmp # head -n 15 /etc/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftppsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
messagebus:x:499:499:User for D-Bus:/var/run/dbus:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
nscd:x:496:495:User for nscd:/run/nscd:/sbin/nologin
ntp:x:74:492:NTP daemon:/var/lib/ntp:/bin/false
```

La commande **head** peut également être utilisée pour afficher les premiers **y** octets en utilisant l'option **-c** :

```
SLES12SP1:/tmp # head -c 150 /etc/passwd
```

```
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srvSLES12SP1:/tmp #
```

Dans le cas où le **y** est négatif, la commande **head** affiche tous les octets du fichier sauf les derniers **y** octets :

```
SLES12SP1:/tmp # head -c -150 /etc/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftpsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
messagebus:x:499:499:User for D-Bus:/var/run/dbus:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
nscd:x:496:495:User for nscd:/run/nscd:/sbin/nologin
ntp:x:74:492:NTP daemon:/var/lib/ntp:/bin/false
openslp:x:494:2:openslp daemon:/var/lib/empty:/sbin/nologin
polkitd:x:497:496:User for polkitd:/var/lib/polkit:/sbin/nologin
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
pulse:x:490:489:PulseAudio daemon:/var/lib/pulseaudio:/sbin/nologin
root:x:0:0:root:/root:/bin/bash
rpc:x:495:65534:user for rpcbind:/var/lib/empty:/sbin/nologin
rtkit:x:491:490:RealtimeKit:/proc:/bin/false
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin
sshd:x:498:498:SSH daemon:/var/lib/ssh:/bin/false
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin
```

```
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
vnc:x:492:491:usSLES12SP1:/tmp #
```

Les valeurs **x** et **y** acceptent des multiplicateurs :

```
SLES12SP1:/tmp # head -c 1b /etc/passwd
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftpsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/basSLES12SP1:/tmp #
```

Les multiplicateurs les plus utilisés sont :

Multiplicateur	Valeur en octets
b	512
KB	1000
K	1024
MB	1000*1000
M	1024*1024
GB	1000*1000*1000
G	1024*1024*1024

Options de la commande



A faire : Utilisez l'option **-help** de la commande **head** pour visualiser les options de la commande.

La commande tail

La commande **tail** permet d'afficher les **x** dernières lignes d'un fichier. Sans options, la valeur de **x** est de 10 par défaut :

```
SLES12SP1:/tmp # tail /etc/passwd
rpc:x:495:65534:user for rpcbind:/var/lib/empty:/sbin/nologin
rtkit:x:491:490:RealtimeKit:/proc:/bin/false
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin
sshd:x:498:498:SSH daemon:/var/lib/ssh:/bin/false
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
vnc:x:492:491:user for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
trainee:x:1000:100:trainee:/home/trainee:/bin/bash
```

Avec l'option **-n**, la valeur de **x** peut être spécifiée :

```
SLES12SP1:/tmp # tail -n 15 /etc/passwd
openslp:x:494:2:openslp daemon:/var/lib/empty:/sbin/nologin
polkitd:x:497:496:User for polkitd:/var/lib/polkit:/sbin/nologin
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
pulse:x:490:489:PulseAudio daemon:/var/lib/pulseaudio:/sbin/nologin
root:x:0:0:root:/root:/bin/bash
rpc:x:495:65534:user for rpcbind:/var/lib/empty:/sbin/nologin
rtkit:x:491:490:RealtimeKit:/proc:/bin/false
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin
sshd:x:498:498:SSH daemon:/var/lib/ssh:/bin/false
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
vnc:x:492:491:user for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
```

```
trainee:x:1000:100:trainee:/home/trainee:/bin/bash
```

La commande **tail** peut également être utilisée pour afficher les derniers **y** octets en utilisant l'option **-c** :

```
SLES12SP1:/tmp # tail -c 150 /etc/passwd
er for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
trainee:x:1000:100:trainee:/home/trainee:/bin/bash
```

Dans le cas où le **y** est positif, la commande **tail** affiche tous les octets du fichier à partir de la position de **y**ième octet :

```
SLES12SP1:/tmp # tail -c 150 /etc/passwd
er for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
trainee:x:1000:100:trainee:/home/trainee:/bin/bash
SLES12SP1:/tmp # tail -c +150 /etc/passwd
v/ftp:/bin/bash
ftpsecure:x:488:65534:Secure FTP User:/var/lib/empty:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
gdm:x:486:485:Gnome Display Manager daemon:/var/lib/gdm:/bin/false
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
messagebus:x:499:499:User for D-Bus:/var/run/dbus:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
nscd:x:496:495:User for nscd:/run/nscd:/sbin/nologin
ntp:x:74:492:NTP daemon:/var/lib/ntp:/bin/false
openslp:x:494:2:openslp daemon:/var/lib/empty:/sbin/nologin
polkitd:x:497:496:User for polkitd:/var/lib/polkit:/sbin/nologin
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
pulse:x:490:489:PulseAudio daemon:/var/lib/pulseaudio:/sbin/nologin
root:x:0:0:root:/root:/bin/bash
rpc:x:495:65534:user for rpcbind:/var/lib/empty:/sbin/nologin
```

```
rtkit:x:491:490:RealtimeKit:/proc:/bin/false
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin
sshd:x:498:498:SSH daemon:/var/lib/ssh:/bin/false
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
vnc:x:492:491:user for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
trainee:x:1000:100:trainee:/home/trainee:/bin/bash
```

Les valeurs **x** et **y** acceptent des multiplicateurs :

```
SLES12SP1:/tmp # tail -c 1b /etc/passwd
kit:x:491:490:RealtimeKit:/proc:/bin/false
scard:x:487:487:Smart Card Reader:/var/run/pcscd:/usr/sbin/nologin
sshd:x:498:498:SSH daemon:/var/lib/ssh:/bin/false
statd:x:489:65534:NFS statd daemon:/var/lib/nfs:/sbin/nologin
usbmux:x:493:65534:usbmuxd daemon:/var/lib/usbmuxd:/sbin/nologin
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
vnc:x:492:491:user for VNC:/var/lib/empty:/sbin/nologin
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
trainee:x:1000:100:trainee:/home/trainee:/bin/bash
```

Les multiplicateurs les plus utilisés sont :

Multiplicateur	Valeur en octets
b	512
KB	1000
K	1024
MB	1000*1000
M	1024*1024
GB	1000*1000*1000
G	1024*1024*1024

Une option intéressante pour la surveillance des fichiers de journalisation est **-f**. Cette option met à jour l'affichage au fur et à mesure que le fichier est mis à jour :

```
SLES12SP1:/tmp # tail -f /var/log/messages
2016-12-14T20:45:01.238708+01:00 SLES12SP1 systemd: pam_unix(systemd-user:session): session opened for user root
by (uid=0)
2016-12-14T20:45:01.275146+01:00 SLES12SP1 CRON[17392]: pam_unix(crond:session): session closed for user root
2016-12-14T20:45:01.288315+01:00 SLES12SP1 systemd: pam_unix(systemd-user:session): session closed for user root
2016-12-14T20:45:41.689270+01:00 SLES12SP1 sh[1320]: Sleeping ' ' '
2016-12-14T20:58:07.757988+01:00 SLES12SP1 sh[1320]: message repeated 4 times: [ Sleeping ' ' '
2016-12-14T21:00:01.284105+01:00 SLES12SP1 cron[23239]: pam_unix(crond:session): session opened for user root by
(uid=0)
2016-12-14T21:00:01.291722+01:00 SLES12SP1 systemd: pam_unix(systemd-user:session): session opened for user root
by (uid=0)
2016-12-14T21:00:01.338305+01:00 SLES12SP1 CRON[23239]: pam_unix(crond:session): session closed for user root
2016-12-14T21:00:01.351426+01:00 SLES12SP1 systemd: pam_unix(systemd-user:session): session closed for user root
2016-12-14T21:01:14.285113+01:00 SLES12SP1 sh[1320]: Sleeping ' ' '
^C
```

Options de la commande



A faire : Utilisez l'option **-help** de la commande **tail** pour visualiser les options de la commande.

LAB #4 - Utiliser les commandes grep, tr et cut pour isoler l'adresse IPv4

```
SLES12SP1:/tmp # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:10:B5:86
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
```

```
inet6 addr: fe80::a00:27ff:fe10:b586/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:8226 errors:0 dropped:0 overruns:0 frame:0
TX packets:16490 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:588832 (575.0 Kb) TX bytes:1303594 (1.2 Mb)
```

```
SLES12SP1:/tmp # ifconfig eth0 | grep "inet"
inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe10:b586/64 Scope:Link
```

```
SLES12SP1:/tmp # ifconfig eth0 | grep "inet" | grep -v "inet6"
inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
```

```
SLES12SP1:/tmp # ifconfig eth0 | grep "inet" | grep -v "inet6" | tr -s " " ":"
:inet:addr:10.0.2.15:Bcast:10.0.2.255:Mask:255.255.255.0
```

```
SLES12SP1:/tmp # ifconfig eth0 | grep "inet" | grep -v "inet6" | tr -s " " ":" | cut -d: -f4
10.0.2.15
```



Important : Notez l'utilisation de l'option -s avec la commande tr. Cette option permet de remplacer une suite de x caractères identiques par un seul caractère.

<html>

Copyright © 2019 Hugh Norris.

</html>

From:
<https://ittraining.team/> - **www.ittraining.team**

Permanent link:
<https://ittraining.team/doku.php?id=elearning:workbooks:opensuse:11:utilisateur:l104>

Last update: **2020/01/30 03:28**

