

Version : **2023.01**

Dernière mise-à-jour : 2023/07/14 10:43

# LCF901 - La Ligne de Commande

## Contenu du Module

- **LCF901 - La Ligne de Commande**

- Contenu du Module
- Le Shell
- LAB #1 - Le Shell /bin/bash
  - 1.1 - Les Commandes Internes et Externes au shell
  - 1.2 - Les alias
  - 1.3 - Définir le Shell d'un Utilisateur
  - 1.4 - Le Prompt
  - 1.5 - Rappeler des Commandes
  - 1.6 - Générer les fins de noms de fichiers
  - 1.7 - Le shell interactif
    - Caractère \*
    - Caractère ?
    - Caractères [ ]
  - 1.8 - L'option extglob
    - ?(expression)
    - \*(expression)
    - +(expression)
    - @(expression)
    - !(expression)
    - Caractères d'Échappement
  - 1.9 - Codes Retour
  - 1.10 - Redirections

- 1.11 - Pipes
- 1.12 - Substitutions de Commandes
- 1.13 - Chaînage de Commandes
- 1.14 - Affichage des variables du shell
  - Les variables principales
  - Les Variables de Régionalisation et d'Internationalisation
  - Les variables spéciales
- 1.15 - La Commande env
- 1.16 - Options du Shell Bash
  - Exemples
    - noclobber
    - noglob
    - nounset
- LAB #2 - Options et Arguments
- LAB #3 - Expressions Régulières
  - ERb
  - ERe
- Outils de Manipulation de Fichiers Texte
  - Présentation des Commandes grep, egrep et fgrep
    - La commande grep
    - La Commande egrep
    - La Commande fgrep
    - LAB #4 - Utiliser grep, egrep et fgrep
  - Présentation de la Commande sed
    - LAB #5 - Utiliser la Commande sed
  - Présentation de La Commande awk
    - Découpage en champs
    - Critères
      - Une expression régulière valide pour la ligne
      - Une expression régulière valide pour un champ
      - Une comparaison
      - Un opérateur logique
      - Une variable interne
    - Scripts awk

- La Fonction printf
- Structures de Contrôle
  - if
  - for
  - while
  - do-while
- LAB #6 - Utiliser la Commande awk
- LAB #7 -Autres Commandes Utiles
  - 7.1 - La Commande expand
  - 7.2 - La Commande unexpand
  - 7.3 - La Commande cut
  - 7.4 - La Commande uniq
  - 7.5 - La Commande tr
  - 7.6 - La Commande paste
  - 7.7 - La Commande split
  - 7.8 - La Commande diff
  - 7.9 - La Commande cmp
  - 7.10 - La commande patch
  - 7.11 - La commande strings
  - 7.12 - La commande comm
  - 7.13 - La commande head
  - 7.14 - La commande tail
- LAB #8 - Utiliser les commandes ifconfig, grep, tr et cut pour isoler l'adresse IPv4
- LAB #9 - Utiliser les commandes ip, grep, awk et sed pour isoler l'adresse IPv4

## Le Shell

Un shell est un **interpréteur de commandes** ou en anglais un **Command Line Interpreter (C.L.I.)**. Il est utilisé comme interface pour donner des instructions ou **commandes** au système d'exploitation.

Le mot shell est générique. Il existe de nombreux shells dans le monde Unix, par exemple :

<b>Shell</b>	<b>Nom</b>	<b>Date de Sortie</b>	<b>Inventeur</b>	<b>Commande</b>	<b>Commentaires</b>
tsh	Thompson Shell	1971	Ken Thompson	sh	Le premier shell
sh	Bourne Shell	1977	Stephen Bourne	sh	Le shell commun à tous les Unix. Sous RHEL/CentOS 8 : /usr/bin/sh
csh	C-Shell	1978	Bill Joy	csh	Le shell BSD. Sous RHEL/CentOS 8 : /usr/bin/csh
tcsh	Tenex C-Shell	1979	Ken Greer	tcsh	Un dérivé du shell csh. Sous RHEL/CentOS 8 : /usr/bin/tcsh
ksh	Korn Shell	1980	David Korn	ksh	Uniquement libre depuis 2005. Sous RHEL/CentOS 8 : /usr/bin/ksh
bash	Bourne Again Shell	1987	Brian Fox	bash	Le shell par défaut de Linux et de MacOS X. Sous RHEL/CentOS 8 : /usr/bin/bash
zsh	Z Shell	1990	Paul Falstad	zsh	Zsh est plutôt orienté pour l'interactivité avec l'utilisateur. Sous RHEL/CentOS 8 : /usr/bin/zsh

Sous RHEL/CentOS 8 le shell **/bin/sh** est un lien symbolique vers **/bin/bash** :

```
[trainee@centos8 ~]$ ls -l /bin/sh
lrwxrwxrwx. 1 root root 4 Jul 21 2020 /bin/sh -> bash
```

## LAB #1 - Le Shell **/bin/bash**

Ce module concerne l'utilisation du shell **bash** sous Linux. Le shell **bash** permet de:

- Rappeler des commandes
- Générer la fin de noms de fichiers
- Utiliser des alias
- Utiliser les variables tableaux
- Utiliser les variables numériques et l'arithmétique du langage C
- Gérer des chaînes de caractères
- Utiliser les fonctions

Une commande commence toujours par un mot clef. Ce mot clef est interprété par le shell selon le type de commande et dans l'ordre qui suit :

1. Les alias
2. Les fonctions

3. Les commandes internes au shell
4. Les commandes externes au shell

## 1.1 - Les Commandes Internes et Externes au shell

Les commandes internes au shell sont des commandes telles **cd**. Pour vérifier le type de commande, il faut utiliser la commande **type** :

```
[trainee@centos7 ~]$ type cd  
cd is a shell builtin
```

Les commandes externes au shell sont des binaires exécutables ou des scripts, généralement situés dans /bin, /sbin, /usr/bin ou /usr/sbin :

```
[trainee@centos8 ~]$ type cd  
cd is a shell builtin
```

## 1.2 - Les alias

Les alias sont des noms permettant de désigner une commande ou une suite de commandes et ne sont spécifiques qu'au shell qui les a créés ainsi qu'à l'environnement de l'utilisateur :

```
[trainee@centos8 ~]$ type ls  
ls is aliased to `ls --color=auto'
```



**Important** : Notez que dans ce cas l'alias **ls** est en effet un alias qui utilise la **commande ls** elle-même.

Un alias se définit en utilisant la commande **alias** :

```
[trainee@centos8 ~]$ alias dir='ls -l'  
[trainee@centos8 ~]$ dir  
total 0  
-rw-rw-r--. 1 trainee trainee 0 Apr 20 03:46 aac  
-rw-rw-r--. 1 trainee trainee 0 Apr 20 03:46 abc  
-rw-rw-r--. 1 trainee trainee 0 Apr 20 03:46 bca  
-rw-rw-r--. 1 trainee trainee 0 Apr 20 03:46 xyz
```



**Important :** Notez que la commande **dir** existe vraiment. Le fait de créer un alias qui s'appelle **dir** implique que l'alias sera exécuté à la place de la commande **dir**.

La liste des alias définis peut être visualisée en utilisant la commande **alias** :

```
[trainee@centos8 ~]$ alias  
alias dir='ls -l'  
alias egrep='egrep --color=auto'  
alias fgrep='fgrep --color=auto'  
alias grep='grep --color=auto'  
alias l.='ls -d .* --color=auto'  
alias ll='ls -l --color=auto'  
alias ls='ls --color=auto'  
alias vi='vim'  
alias which='(alias; declare -f) | /usr/bin/which --tty-only --read-alias --read-functions --show-tilde --show-dot'  
alias xzgrep='xzgrep --color=auto'  
alias xzgrep='xzgrep --color=auto'  
alias xzgrep='xzgrep --color=auto'  
alias zgrep='zgrep --color=auto'  
alias zgrep='zgrep --color=auto'  
alias zgrep='zgrep --color=auto'
```



**Important :** Notez que cette liste contient, sans distinction, les alias définis dans les fichiers de démarrage du système ainsi que l'alias **dir** créé par **trainee** qui n'est que disponible à **trainee** dans le terminal courant.

Pour forcer l'exécution d'une commande et non l'alias il faut faire précéder la commande par le caractère \ :

```
[trainee@centos8 ~]$ \dir  
aac abc bca xyz
```

Pour supprimer un alias, il convient d'utiliser la commande **unalias** :

```
[trainee@centos8 ~]$ unalias dir  
[trainee@centos8 ~]$ dir  
aac abc bca xyz
```

### 1.3 - Définir le Shell d'un Utilisateur

Le shell des utilisateurs est défini par **root** dans le dernier champs du fichier **/etc/passwd** :

```
[trainee@centos8 ~]$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin
```

```
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
libstoragemgmt:x:996:993:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
cockpit-ws:x:995:991:User for cockpit-ws:/nonexisting:/sbin/nologin
sssd:x:994:990:User for sssd:/:/sbin/nologin
setroubleshoot:x:993:989::/var/lib/setroubleshoot:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:992:988::/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

Cependant l'utilisateur peut changer son shell grâce à la commande **chsh**. Les shells disponibles aux utilisateurs du système sont inscrits dans le fichier **/etc/shells**. Saisissez la commande **cat /etc/shells** :

```
[trainee@centos8 ~]$ cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
```

```
/usr/bin/bash
```

Ensuite utilisez la commande **echo** pour afficher le shell actuel de **trainee** :

```
[trainee@centos8 ~]$ echo $SHELL  
/bin/bash
```



**Important** : Notez sous RHEL/CentOS 8 que le système nous informe que le shell courant de l'utilisateur **trainee** est **/bin/bash** et non **/usr/bin/bash**. Ceci est du au fait que le répertoire **/bin** est un lien symbolique pointant vers le répertoire **/usr/bin**.

Changez ensuite le shell de **trainee** en utilisant la commande **chsh** en indiquant la valeur de **/bin/sh** pour le nouveau shell :

```
[trainee@centos8 ~]$ chsh  
Changing shell for trainee.  
New shell [/bin/bash]  
/bin/sh  
Password: trainee  
Shell changed.
```



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

Vérifiez ensuite le shell actif pour **trainee** :

```
[trainee@centos8 ~]$ echo $SHELL  
/bin/bash
```

Dernièrement contrôlez le shell stipulé dans le fichier **/etc/passwd** pour **trainee** :

```
[trainee@centos8 ~]$ cat /etc/passwd | grep trainee  
trainee:x:1000:1000:trainee:/home/trainee:/bin/sh
```



**Important :** Vous noterez que le shell actif est toujours **/bin/bash** tandis que le shell stipulé dans le fichier `/etc/passwd` est le **/bin/sh**. Le shell **/bin/sh** ne deviendra le shell actif de **trainee** que lors de sa prochaine connexion au système.

Modifiez votre shell à **/bin/bash** de nouveau en utilisant la commande chsh :

```
[trainee@centos8 ~]$ chsh  
Changing shell for trainee.  
New shell [/bin/sh]: /bin/bash  
Password: trainee  
Shell changed.
```



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

## 1.4 - Le Prompt

Le prompt d'un utilisateur dépend de son statut :

- \$ pour un utilisateur normal,
- # pour root.

## 1.5 - Rappeler des Commandes

Le shell **/bin/bash** permet le rappel des dernières commandes saisies. Afin de connaître la liste des commandes mémorisées, utilisez la commande **history** :

```
[trainee@centos8 ~]$ history | more
 1 su -
 2 exit
 3 su -
 4 nmcli c show
 5 stty -a
 6 date
 7 who
 8 df
 9 df -h
10 free free -h
11 free
12 free -h
13 whoami
14 su -
15 pwd
16 cd /tmp
17 pwd
18 ls
19 su -
20 touch test
21 ls
22 echo fenestros
23 cp test ~
--More--
```



**Important:** L'historique est spécifique à chaque utilisateur.

L'historique des commandes est en mode **emacs** par défaut. De ce fait, le rappel de la dernière commande se fait en utilisant la touche **[Flèche vers le haut]** ou bien les touches **[CTRL]-[P]** et le rappel de la commande suivante se fait en utilisant la touche **[Flèche vers le bas]** ou bien les touches **[CTRL]-[N]** :

Caractère de Contrôle	Définition
[CTRL]-[P] (= flèche vers le haut)	Rappelle la commande précédente
[CTRL]-[N] (= flèche vers le bas)	Rappelle la commande suivante

Pour se déplacer dans la ligne de l'historique :

Caractère de Contrôle	Définition
[CTRL]-[A]	Se déplacer au début de la ligne
[CTRL]-[E]	Se déplacer à la fin de la ligne
[CTRL]-[B]	Se déplacer un caractère à gauche
[CTRL]-[F]	Se déplacer un caractère à droite
[CTRL]-[D]	Supprimer le caractère sous le curseur

Pour rechercher dans l'historique il convient d'utiliser les touches :

Caractère de Contrôle	Définition
[CTRL]-[R] chaine	Recherche en arrière de <i>chaine</i> dans l'historique. L'utilisation successive de la combinaison de touches par la suite recherche d'autres occurrences de <i>chaine</i>
[CTRL]-[S] chaine	Recherche en avant de <i>chaine</i> dans l'historique. L'utilisation successive de la combinaison de touches par la suite recherche d'autres occurrences de <i>chaine</i>
[CTRL]-[G]	Sortir du mode recherche

Il est aussi possible de rappeler la dernière commande de l'historique en utilisant les caractères **!!**:

```
[trainee@centos8 ~]$ ls
aac abc bca xyz
[trainee@centos8 ~]$ !!
ls
aac abc bca xyz
```

Vous pouvez rappeler une commande spécifique de l'historique en utilisant le caractère ! suivi du numéro de la commande à rappeler :

```
[trainee@centos8 ~]$ history
 1  su -
...
80  history | more
81  ls
82  history
[trainee@centos8 ~]$ !81
ls
aac abc bca xyz
```

Le paramétrage de la fonction du rappel des commandes est fait pour tous les utilisateurs dans le fichier **/etc/profile**. Dans ce fichier, les variables concernant le rappel des commandes peuvent être définis. Le plus important est **HISTSIZE** :

```
[trainee@centos8 ~]$ cat /etc/profile | grep HISTSIZE
HISTSIZE=1000
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
```

Vous noterez que dans le cas précédent, la valeur de **HISTSIZE** est de **1000**. Ceci implique que les dernières mille commandes sont mémorisées.

Les commandes mémorisées sont stockées dans le fichier **~/.bash\_history**. Les commandes de la session en cours ne sont sauvegardées dans ce fichier qu'à la fermeture de la session :

```
[trainee@centos8 ~]$ nl .bash_history | tail
54  ls
55  ls | sort
56  ls | sort -r
57  more /etc/services
58  less /etc/services
59  find acc
60  find aac
61  su -
62  sleep 10
```

```
63 su -
```



**Important :** Notez l'utilisation de la commande **nl** pour numérotter les lignes de l'affichage du contenu du fichier **.bash\_history**.

## 1.6 - Générer les fins de noms de fichiers

Le shell /bin/bash permet la génération des fins de noms de fichiers. Celle-ci est accomplie grâce à l'utilisation de la touche **[Tab]**. Dans l'exemple qui suit, la commande saisie est :

```
$ ls .b [Tab][Tab][Tab]
```

```
[trainee@centos8 ~]$ ls .bash
.bash_history .bash_logout .bash_profile .bashrc
```



**Important :** Notez qu'en appuyant sur la touche **Tab** trois fois le shell propose 4 possibilités de complétion de nom de fichier. En effet, sans plus d'information, le shell ne sait pas quel fichier est concerné.

La même possibilité existe pour la génération des fins de noms de commandes. Dans ce cas saisissez la commande suivante :

```
$ mo [Tab][Tab]
```

Appuyez sur la touche **Tab** deux fois. Vous obtiendrez une fenêtre similaire à celle-ci :

```
[trainee@centos8 ~]$ mo
modinfo          more          mount.nfs4
```

modprobe	mount	mountpoint
modulemd-validator	mount.fuse	mountstats
modulemd-validator-v1	mount.nfs	

## 1.7 - Le shell interactif

Lors de l'utilisation du shell, nous avons souvent besoin d'exécuter une commande sur plusieurs fichiers au lieu de les traiter individuellement. A cette fin nous pouvons utiliser les caractères spéciaux.

Caractère Spéciaux	Description
*	Représente 0 ou plus de caractères
?	Représente un caractère
[abc]	Représente un caractère parmi ceux entre crochets
[!abc]	Représente un caractère ne trouvant pas parmi ceux entre crochets
?(expression1 expression2  ...)	Représente 0 ou 1 fois l'expression1 ou 0 ou 1 fois l'expression2 ...
*(expression1 expression2  ...)	Représente 0 à x fois l'expression1 ou 0 à x fois l'expression2 ...
+(expression1 expression2  ...)	Représente 1 à x fois l'expression1 ou 1 à x fois l'expression2 ...
@(expression1 expression2  ...)	Représente 1 fois l'expression1 ou 1 fois l'expression2 ...
!(expression1 expression2  ...)	Représente 0 fois l'expression1 ou 0 fois l'expression2 ...

### Caractère \*

Dans votre répertoire individuel, créez un répertoire **training**. Ensuite créez dans ce répertoire 5 fichiers nommés respectivement f1, f2, f3, f4 et f5 :

```
[trainee@centos8 ~]$ mkdir training
[trainee@centos8 ~]$ cd training
[trainee@centos8 training]$ touch f1 f2 f3 f4 f5
[trainee@centos8 training]$ ls
f1  f2  f3  f4  f5
```

Afin de démontrer l'utilisation du caractère spécial \*, saisissez la commande suivante :

```
[trainee@centos8 training]$ echo f*
f1 f2 f3 f4 f5
```



**Important :** Notez que le caractère \* remplace un caractère ou une suite de caractères.

## Caractère ?

Créez maintenant les fichiers f52 et f62 :

```
[trainee@centos8 training]$ touch f52 f62
```

Saisissez ensuite la commande suivante :

```
[trainee@centos8 training]$ echo f?2
f52 f62
```



**Important :** Notez que le caractère ? remplace **un seul** caractère.

## Caractères [ ]

L'utilisation peut prendre plusieurs formes différentes :

Joker	Description
[xyz]	Représente le caractère x ou y ou z
[m-t]	Représente le caractère m ou n .... t

Joker	Description
[!xyz]	Représente un caractère autre que x ou y ou z
[!m-t]	Représente un caractère autre que m ou n .... t

Afin de démontrer l'utilisation des caractères [ et ], créez le fichier a100 :

```
[trainee@centos8 training]$ touch a100
```

Ensuite saisissez les commandes suivantes et notez le résultat :

```
[trainee@centos8 training]$ echo [a-f]*  
a100 f1 f2 f3 f4 f5 f52 f62
```



**Important** : Notez ici que tous les fichiers commençant par les lettres **a**, **b**, **c**, **d**, **e** ou **f** sont affichés à l'écran.

```
[trainee@centos8 training]$ echo [af]*  
a100 f1 f2 f3 f4 f5 f52 f62
```



**Important** : Notez ici que tous les fichiers commençant par les lettres **a** ou **f** sont affichés à l'écran.

```
[trainee@centos8 training]$ echo [|a]*  
f1 f2 f3 f4 f5 f52 f62
```



**Important** : Notez ici que tous les fichiers sont affichés à l'écran, à l'exception d'un fichier



commençant par la lettre **a**.

```
[trainee@centos8 training]$ echo [a-b]*  
a100
```



**Important :** Notez ici que seul le fichier commençant par la lettre **a** est affiché à l'écran car il n'existe pas de fichiers commençant par la lettre **b**.

```
[trainee@centos8 training]$ echo [a-f]  
[a-f]
```



**Important :** Notez que dans ce cas, il n'existe pas de fichiers dénommés **a**, **b**, **c**, **d**, **e** ou **f**. Pour cette raison, n'ayant trouvé aucune correspondance entre le filtre utilisé et les objets dans le répertoire courant, la commande **echo** retourne le filtre passé en argument, c'est-à-dire **[a-f]**.

## 1.8 - L'option extglob

Activez l'option **extglob** du shell bash afin de pouvoir utiliser **?(expression)**, **\*(expression)**, **+(expression)**, **@(expression)** et **!(expression)** :

```
[trainee@centos8 training]$ shopt -s extglob
```

La commande **shopt** est utilisée pour activer ou désactiver les options du comportement optional du shell. La liste des options peut être visualisée en exécutant la commande **shopt** sans options :

```
[trainee@centos8 training]$ shopt
autocd          off
cable_vars     off
cdspell         off
checkhash       off
checkjobs       off
checkwinsize    on
cmdhist         on
compat31        off
compat32        off
compat40        off
compat41        off
direxpand       off
dirspell         off
dotglob          off
execfail        off
expand_aliases  on
extdebug        off
extglob          on
extquote        on
failglob         off
force_fignore   on
globstar         off
gnu_errfmt      off
histappend      on
histreedit      off
histverify      off
hostcomplete    off
huponexit       off
interactive_comments  on
lastpipe        off
lithist         off
login_shell     on
mailwarn        off
```

```
no_empty_cmd_completion off
nocaseglob      off
nocasematch     off
nullglob        off
progcomp        on
promptvars      on
restricted_shell off
shift_verbose   off
sourcepath      on
xpg_echo        of
```

## ?(expression)

Créez les fichiers f, f.txt, f123.txt, f123123.txt, f123123123.txt :

```
[trainee@centos8 training]$ touch f f.txt f123.txt f123123.txt f123123123.txt
```

Saisissez la commande suivante :

```
[trainee@centos8 training]$ ls f?(123).txt
f123.txt  f.txt
```



**Important** : Notez ici que la commande affiche les fichiers ayant un nom contenant 0 ou 1 occurrence de la chaîne **123**.

## \*(expression)

Saisissez la commande suivante :

```
[trainee@centos8 training]$ ls f*(123).txt  
f123123123.txt  f123123.txt  f123.txt  f.txt
```



**Important :** Notez ici que la commande affiche les fichiers ayant un nom contenant de 0 jusqu'à x occurrences de la chaîne **123**.

## + (expression)

Saisissez la commande suivante :

```
[trainee@centos8 training]$ ls f+(123).txt  
f123123123.txt  f123123.txt  f123.txt
```



**Important :** Notez ici que la commande affiche les fichiers ayant un nom contenant entre 1 et x occurrences de la chaîne **123**.

## @ (expression)

Saisissez la commande suivante :

```
[trainee@centos8 training]$ ls f@(123).txt  
f123.txt
```



**Important :** Notez ici que la commande affiche les fichiers ayant un nom contenant 1



seule occurrence de la chaîne **123**.

## !(expression)

Saisissez la commande suivante :

```
[trainee@centos8 training]$ ls f!(123).txt  
f123123123.txt  f123123.txt  f.txt
```



**Important** : Notez ici que la commande n'affiche que les fichiers ayant un nom qui ne contient **pas** la chaîne **123**.

## Caractères d'Échappement

Afin d'utiliser un caractère spécial dans un contexte littéral, il faut utiliser un caractère d'échappement. Il existe trois caractères d'échappement :

Caractère	Description
\	Protège le caractère qui le suit
' '	Protège tout caractère, à l'exception du caractère ' lui-même, se trouvant entre les deux '
" "	Protège tout caractère, à l'exception des caractères " lui-même, \$, \ et ', se trouvant entre les deux "

Afin d'illustrer l'utilisation des caractères d'échappement, considérons la commande suivante :

```
$ echo * est un caractère spécial [Entrée]
```

Lors de la saisie de cette commande dans votre répertoire **training**, vous obtiendrez une fenêtre similaire à celle-ci :

```
[trainee@centos8 training]$ echo * est un caractère spécial  
a100 f f1 f123123123.txt f123123.txt f123.txt f2 f3 f4 f5 f52 f62 f.txt est un caractère spécial  
  
[trainee@centos8 training]$ echo \* est un caractère spécial  
* est un caractère spécial  
  
[trainee@centos8 training]$ echo "* est un caractère spécial"  
* est un caractère spécial  
  
[trainee@centos8 training]$ echo '* est un caractère spécial'  
* est un caractère spécial
```

## 1.9 - Codes Retour

Chaque commande retourne un code à la fin de son exécution. La variable spéciale **\$?** sert à stocker le code retour de la dernière commande exécutée.

Par exemple :

```
[trainee@centos8 training]$ cd ..  
[trainee@centos8 ~]$ mkdir codes  
[trainee@centos8 ~]$ echo $?  
0  
[trainee@centos8 ~]$ touch codes/exit.txt  
[trainee@centos8 ~]$ rmdir codes  
rmdir: failed to remove 'codes': Directory not empty  
[trainee@centos8 ~]$ echo $?  
1
```

Dans cette exemple la création du répertoire **codes** s'est bien déroulée. Le code retour stocké dans la variable **\$?** est un zéro.

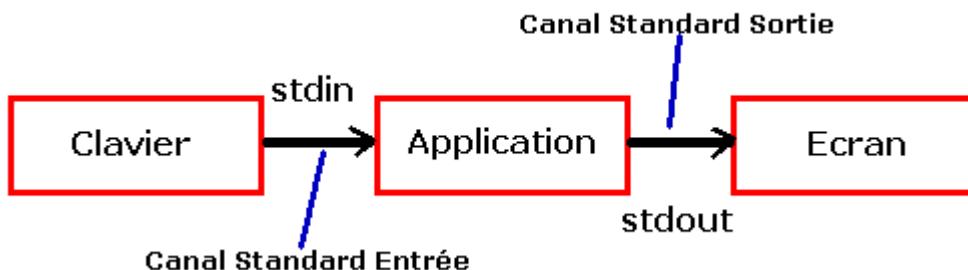
La suppression du répertoire a rencontré une erreur car **codes** contenait le fichier **retour**. Le code retour stocké dans la variable **\$?** est un **un**.

Si le code retour est **zéro** la dernière commande s'est déroulée sans erreur.

Si le code retour est **autre que zéro** la dernière commande s'est déroulée avec une erreur.

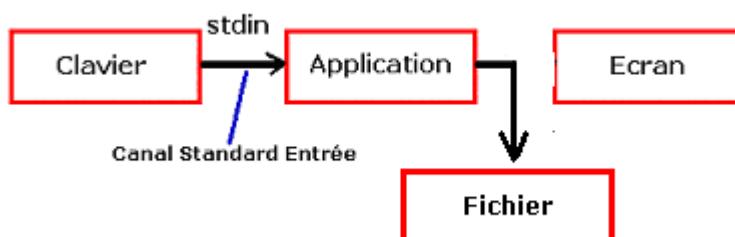
## 1.10 - Redirections

Votre dialogue avec le système Linux utilise des canaux d'entrée et de sortie. On appelle le clavier, le **canal d'entrée standard** et l'écran, le **canal de sortie standard** :



Autrement dit, en tapant une commande sur le clavier, vous voyez le résultat de cette commande à l'écran.

Parfois, cependant il est utile de re-diriger le canal de sortie standard vers un fichier. De cette façon, le résultat d'une commande telle **free** peut être stocké dans un fichier pour une consultation ultérieure :



Cet effet est obtenu en utilisant une **redirection** :

```
[trainee@centos8 ~]$ pwd  
/home/trainee  
[trainee@centos8 ~]$ cd training  
[trainee@centos8 training]$ free > file  
[trainee@centos8 training]$ cat file
```

	total	used	free	shared	buff/cache	available
Mem:	500780	192692	38916	4824	269172	260472
Swap:	2096124	0	2096124			

Si le fichier cible n'existe pas, il est créé et son contenu sera le résultat de la commande free.

Par contre si le fichier existe déjà, il sera écrasé :

```
[trainee@centos8 training]$ date > file  
[trainee@centos8 training]$ cat file  
Mon 28 Nov 15:48:09 CET 2016
```

Pour ajouter des données supplémentaires au même fichier cible, il faut utiliser une **double redirection** :

```
[trainee@centos8 training]$ free >> file  
[trainee@centos8 training]$ cat file  
Mon 28 Nov 15:48:09 CET 2016
```

	total	used	free	shared	buff/cache	available
Mem:	500780	192792	38516	4824	269472	260376
Swap:	2096124	0	2096124			

De cette façon, la date du jour sera rajoutée à la fin de votre fichier après les informations de la commande free.



**Important** : Notez que la sortie standard ne peut être redirigée que dans **une seule direction**.

Les canaux d'entrées et de sorties sont numérotés :

- 0 = Le Canal d'entrée Standard
- 1 = Le Canal de Sortie Standard
- 2 = Le Canal d'erreur

La commande suivante créera un fichier nommé **errorlog** qui contient les messages d'erreur de l'exécution de la commande **rmdir** :

```
[trainee@centos8 training]$ cd ..
[trainee@centos8 ~]$ rmkdir training/ 2>errorlog
[trainee@centos8 ~]$ cat errorlog
rmkdir: failed to remove 'training/': Directory not empty
```

En effet l'erreur est générée parce que le répertoire **training** n'est pas vide.

Nous pouvons également réunir des canaux. Pour mettre en application ceci, il faut comprendre que le shell traite les commandes de **gauche à droite**.

Dans l'exemple suivant, nous réunissons le canal de sortie et le canal d'erreurs :

```
[trainee@centos8 ~]$ free > file 2>&1
```

La syntaxe **2>&1** envoie la sortie du canal 2 au même endroit que le canal 1, à savoir le fichier dénommé **file**.

Il est possible de modifier le canal d'entrée standard afin de lire des informations à partir d'un fichier. Dans ce cas la redirection est obtenue en utilisant le caractère < :

```
[trainee@centos8 ~]$ wc -w < errorlog
8
```

Dans cet exemple la commande **wc** compte le nombre de mots ( **-w** ) dans le fichier **errorlog** et l'affiche à l'écran :

D'autres redirections existent :

Caractères	Définition
&>	Rediriger les canaux 1 et 2 au même endroit
<<	Permet d'utiliser le texte taper ensuite en tant que entrée standard. Par exemple programme << EOF utilisera le texte taper après en tant qu'entrée standard jusqu'à l'apparition de EOF sur une ligne seule.
<>	Permet d'utiliser le fichier spécifié en tant que entrée standard et sortie standard

## 1.11 - Pipes

Il est aussi possible de relier des commandes avec un pipe | .

Dans ce cas, le canal de sortie de la commande à gauche du pipe est envoyé au canal d'entrée de la commande à droite du pipe :

```
[trainee@centos8 ~]$ ls | wc -w
7
```

Cette commande, lancée dans votre répertoire personnel, prend la sortie de la commande **ls** et demande à la commande **wc** de compter le nombre de mots inclus dans la sortie de ls :



**Important** : Il est à noter qu'il est possible de relier plusieurs tubes dans la même commande.

Rappelez-vous que la sortie standard ne peut être redirigée que dans une seule direction. Afin de pouvoir rediriger la sortie standard vers un fichier **et** la visualiser à l'écran, nous devons utiliser la commande **tee** avec un pipe :

```
[trainee@centos8 ~]$ date | tee file1
Tue 20 Apr 10:39:47 EDT 2021
[trainee@centos8 ~]$ cat file1
Tue 20 Apr 10:39:47 EDT 2021
```

Cette même technique nous permet de créer **deux fichiers** :

```
[trainee@centos8 ~]$ date | tee file1 > file2
[trainee@centos8 ~]$ cat file1
Tue 20 Apr 10:40:36 EDT 2021
[trainee@centos8 ~]$ cat file2
Tue 20 Apr 10:40:36 EDT 2021
```



**Important :** Par défaut la commande tee écrase le fichier de destination. Pour ajouter des données supplémentaires au même fichier cible, il convient d'utiliser l'option **-a** de la commande tee.

## 1.12 - Substitutions de Commandes

Il est parfois intéressant, notamment dans les scripts, de remplacer une commande par sa valeur de sa sortie. Afin d'illustrer ce point, considérons les commandes suivantes :

```
[trainee@centos8 ~]$ echo date
date
[trainee@centos8 ~]$ echo $(date)
Tue 20 Apr 10:41:33 EDT 2021
[trainee@centos8 ~]$ echo `date`
Tue 20 Apr 10:41:45 EDT 2021
```



**Important :** Notez le format de chaque substitution **\$(commande)** ou **`commande`**. Sur un clavier français, l'anti-côte est accessible en utilisant les touches **Alt Gr** et **77**.

## 1.13 - Chainage de Commandes

Il est possible de regrouper des commandes à l'aide d'un sous-shell :

```
$ (ls -l; ps; who) > list [Entrée]
```

Cet exemple envoie le résultat des trois commandes vers le fichier **list** en les traitant en tâches de fond.

Les commandes peuvent être aussi chainées en fonction du code retour de la commande précédente.

**&&** est utilisé afin de s'assurer que la deuxième commande s'exécute dans le cas où la valeur du statut de sortie est 0, autrement dit qu'il n'y a pas eu d'erreurs.

**||** est utilisé afin de s'assurer de l'inverse.

Le syntaxe de cette commande est :

```
Commande1 && Commande2
```

Dans ce cas, Commande 2 est exécutée uniquement dans le cas où Commande1 s'est exécuté sans erreur

Ou :

```
Commande1 || Commande2
```

Dans ce cas, Commande2 est exécuté si Commande1 a rencontré une erreur.

## 1.14 - Affichage des variables du shell

Une variable du shell peut être affichée grâce à la commande :

```
$ echo $VARIABLE [Entrée]
```

## Les variables principales

Variable	Description
BASH	Le chemin complet du shell.
BASH_VERSION	La version du shell.
EUID	EUID de l'utilisateur courant.
UID	UID de l'utilisateur courant.
PPID	Le PID du processus père.
PWD	Le répertoire courant.
OLDPWD	Le répertoire avant la dernière commande cd. Même chose que la commande <b>cd -</b> .
RANDOM	Un nombre aléatoire entre 0 et 32767
SECONDS	Le nombre de secondes écoulées depuis le lancement du shell
LINES	Le nombre de lignes de l'écran.
COLUMNS	La largeur de l'écran.
HISTFILE	Le fichier historique
HISTFILESIZE	La taille du fichier historique
HISTSIZE	Le nombre de commandes mémorisées dans le fichier historique
HISTCMD	Le numéro de la commande courante dans l'historique
HISTCONTROL	<b>ignorespace</b> ou <b>ignoredups</b> ou <b>ignoreboth</b>
HOME	Le répertoire de connexion.
HOSTTYPE	Le type de machine.
OSTYPE	Le système d'exploitation.
MAIL	Le fichier contenant le courrier.
MAILCHECK	La fréquence de vérification du courrier en secondes.
PATH	Le chemin de recherche des commandes.
PROMPT_COMMAND	La commande exécutée avant chaque affichage du prompt.
PS1	Le prompt par défaut.
PS2	Le deuxième prompt par défaut
PS3	Le troisième prompt par défaut
PS4	Le quatrième prompt par défaut
SHELL	Le shell de préférence.

Variable	Description
SHLVL	Le nombre d'instances du shell.
TMOUT	Le nombre de secondes moins 60 d'inactivité avant que le shell exécute la commande <b>exit</b> .

## Les Variables de Régionalisation et d'Internationalisation

L'**Internationalisation**, aussi appelé **i18n** car il y a 18 lettres entre la lettre **I** et la lettre **n** dans le mot *Internationalization*, consiste à adapter un logiciel aux paramètres variant d'une région à l'autre :

- longueur des mots,
- accents,
- écriture de gauche à droite ou de droite à gauche,
- unité monétaire,
- styles typographiques et modèles rédactionnels,
- unités de mesures,
- affichage des dates et des heures,
- formats d'impression,
- format du clavier,
- etc ...

Le **Régionalisation**, aussi appelé **I10n** car il y a 10 lettres entre la lettre **L** et la lettre **n** du mot *Localisation*, consiste à modifier l'internationalisation en fonction d'une région spécifique.

Le code pays complet prend la forme suivante : **langue-PAYS.jeu\_de\_caractères**. Par exemple, pour la langue anglaise les valeurs de langue-PAYS sont :

- en\_GB = Great Britain,
- en\_US = USA,
- en\_AU = Australia,
- en\_NZ = New Zealand,
- en\_ZA = South Africa,
- en\_CA = Canada.

Les variables système les plus importants contenant les informations concernant le régionalisation sont :

Variable	Description
LC_ALL	Avec une valeur non nulle, celle-ci prend le dessus sur la valeur de toutes les autres variables d'internationalisation
LANG	Fournit une valeur par défaut pour les variables d'environnement dont la valeur est nulle ou non définie.
LC_CTYPE	Détermine les paramètres régionaux pour l'interprétation de séquence d'octets de données texte en caractères.

Par exemple :

```
[trainee@centos8 ~]$ echo $LC_ALL
[trainee@centos8 ~]$ echo $LC_CTYPE
[trainee@centos8 ~]$ echo $LANG
en_GB.UTF-8
[trainee@centos8 ~]$ locale
LANG=en_GB.UTF-8
LC_CTYPE="en_GB.UTF-8"
LC_NUMERIC="en_GB.UTF-8"
LC_TIME="en_GB.UTF-8"
LC_COLLATE="en_GB.UTF-8"
LC_MONETARY="en_GB.UTF-8"
LC_MESSAGES="en_GB.UTF-8"
LC_PAPER="en_GB.UTF-8"
LC_NAME="en_GB.UTF-8"
LC_ADDRESS="en_GB.UTF-8"
LC_TELEPHONE="en_GB.UTF-8"
LC_MEASUREMENT="en_GB.UTF-8"
LC_IDENTIFICATION="en_GB.UTF-8"
LC_ALL=
```

## Les variables spéciales

Variable	Description
\$LINENO	Contient le numéro de la ligne courante du script ou de la fonction

Variable	Description
\$\$	Contient le PID du shell en cours
\$PPID	Contient le PID du processus parent du shell en cours
\$0	Contient le nom du script en cours tel que ce nom ait été saisi sur la ligne de commande
\$1, \$2 ...	Contient respectivement le premier argument, deuxième argument etc passés au script
\$#	Contient le nombre d'arguments passés au script
\$*	Contient l'ensemble des arguments passés au script
\$@	Contient l'ensemble des arguments passés au script

## 1.15 - La Commande env

La commande **env** envoie sur la sortie standard les valeurs des variables système de l'environnement de l'utilisateur qui l'invoque :

```
[trainee@centos8 ~]$ env
LS_COLORS=rs=0:di=38;5;33:ln=38;5;51:mh=00:pi=40;38;5;11:so=38;5;13:do=38;5;5:bd=48;5;232;38;5;11:cd=48;5;232;38;
5;3:or=48;5;232;38;5;9:mi=01;05;37;41:su=48;5;196;38;5;15:sg=48;5;11;38;5;16:ca=48;5;196;38;5;226:tw=48;5;10;38;5
;16:ow=48;5;10;38;5;21:st=48;5;21;38;5;15:ex=38;5;40:*.tar=38;5;9:*.tgz=38;5;9:*.arc=38;5;9:*.arj=38;5;9:*.taz=38
;5;9:*.lha=38;5;9:*.lz4=38;5;9:*.lzh=38;5;9:*.lzma=38;5;9:*.tlz=38;5;9:*.txz=38;5;9:*.tzo=38;5;9:*.t7z=38;5;9:*.z
ip=38;5;9:*.z=38;5;9:*.dz=38;5;9:*.gz=38;5;9:*.lrz=38;5;9:*.lz=38;5;9:*.lzo=38;5;9:*.xz=38;5;9:*.zst=38;5;9:*.tzs
t=38;5;9:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.tbz2=38;5;9:*.tz=38;5;9:*.deb=38;5;9:*.rpm=38;5;9:*.jar=38;5;9:*
.war=38;5;9:*.ear=38;5;9:*.sar=38;5;9:*.rar=38;5;9:*.alz=38;5;9:*.ace=38;5;9:*.zoo=38;5;9:*.cpio=38;5;9:*.7z=38;5
;9:*.rz=38;5;9:*.cab=38;5;9:*.wim=38;5;9:*.swm=38;5;9:*.dwm=38;5;9:*.esd=38;5;9:*.jpg=38;5;13:*.jpeg=38;5;13:*.mj
pg=38;5;13:*.mpeg=38;5;13:*.gif=38;5;13:*.bmp=38;5;13:*.pbm=38;5;13:*.pgm=38;5;13:*.ppm=38;5;13:*.tga=38;5;13:*.x
bm=38;5;13:*.xpm=38;5;13:*.tif=38;5;13:*.tiff=38;5;13:*.png=38;5;13:*.svg=38;5;13:*.svgz=38;5;13:*.mng=38;5;13:*
.pcx=38;5;13:*.mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;13:*.m2v=38;5;13:*.mkv=38;5;13:*.webm=38;5;13:*.ogm=38;5;13:*
.mp4=38;5;13:*.m4v=38;5;13:*.mp4v=38;5;13:*.vob=38;5;13:*.qt=38;5;13:*.nuv=38;5;13:*.wmv=38;5;13:*.ASF=38;5;13:*
.ram=38;5;13:*.rmvb=38;5;13:*.flc=38;5;13:*.avi=38;5;13:*.fli=38;5;13:*.flv=38;5;13:*.gl=38;5;13:*.dl=38;5;13:*.xc
f=38;5;13:*.xwd=38;5;13:*.yuv=38;5;13:*.cgm=38;5;13:*.emf=38;5;13:*.ovg=38;5;13:*.ogx=38;5;13:*.aac=38;5;45:*.au=
38;5;45:*.flac=38;5;45:*.m4a=38;5;45:*.mid=38;5;45:*.midi=38;5;45:*.mka=38;5;45:*.mp3=38;5;45:*.mpc=38;5;45:*.ogg
=38;5;45:*.ra=38;5;45:*.wav=38;5;45:*.oga=38;5;45:*.opus=38;5;45:*.spx=38;5;45:*.xspf=38;5;45:
SSH_CONNECTION=10.0.2.2 42834 10.0.2.15 22
LANG=en_GB.UTF-8
HISTCONTROL=ignoredupe
```

```
GUESTFISH_RESTORE=\e[0m
HOSTNAME=centos8.ittraining.loc
GUESTFISH_INIT=\e[1;34m
XDG_SESSION_ID=9
USER=trainee
GUESTFISH_PS1=\[\e[1;32m\]><fs>\[\e[0;31m\
SELINUX_ROLE_REQUESTED=
PWD=/home/trainee
HOME=/home/trainee
SSH_CLIENT=10.0.2.2 42834 22
SELINUX_LEVEL_REQUESTED=
SSH_TTY=/dev/pts/0
MAIL=/var/spool/mail/trainee
TERM=xterm-256color
SHELL=/bin/bash
SELINUX_USE_CURRENT_RANGE=
SHLVL=1
LOGNAME=trainee
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
XDG_RUNTIME_DIR=/run/user/1000
PATH=/home/trainee/.local/bin:/home/trainee/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
GUESTFISH_OUTPUT=\e[0m
HISTSIZE=1000
LESSOPEN=||/usr/bin/lesspipe.sh %s
_= /usr/bin/env
OLDPWD=/home/trainee/training
```

La commande peut aussi être utilisée pour fixer une variable lors de l'exécution d'une commande. Par exemple, pour lancer **xterm** avec la variable EDITOR fixée à **vi** :

```
$ env EDITOR=vi xterm
```

## 1.16 - Options du Shell Bash

Pour visualiser les options du shell bash, il convient d'utiliser la commande **set** :

```
$ set -o [Entrée]
```

Par exemple :

```
[trainee@centos8 ~]$ set -o
allexport      off
braceexpand    on
emacs          on
errexit        off
errtrace       off
functrace      off
hashall        on
histexpand     on
history        on
ignoreeof      off
interactive-comments  on
keyword        off
monitor        on
noclobber     off
noexec         off
noglob         off
nolog          off
notify         off
nounset        off
onecmd         off
physical       off
pipefail      off
posix          off
privileged    off
```

verbose	off
vi	off
xtrace	off

Pour activer une option il convient de nouveau à utiliser la commande **set** :

```
[trainee@centos8 ~]$ set -o allexport
[trainee@centos8 ~]$ set -o
allexport      on
braceexpand    on
...
```

Notez que l'option **allexport** a été activée.

Pour désactiver une option, on utilise la commande **set** avec l'option **+o** :

```
$ set +o allexport [Entrée]

[trainee@centos8 ~]$ set +o allexport
[trainee@centos8 ~]$ set -o
allexport      off
braceexpand    on
...
```

Parmi les options, voici la description des plus intéressantes :

Option	Valeur par Défaut	Description
allexport	off	Le shell export automatiquement toute variable
emacs	on	L'édition de la ligne de commande est au style emacs
history	on	L'historique des commandes est activé
noclobber	off	Les simples re-directions n'écrasent pas le fichier de destination
noglob	off	Désactive l'expansion des caractères génériques
nounset	off	Le shell retourne une erreur lors de l'expansion d'une variable inconnue

Option	Valeur par Défaut	Description
verbose	off	Affiche les lignes de commandes saisies
vi	off	L'édition de la ligne de commande est au style vi

## Exemples

### noclobber

```
[trainee@centos8 ~]$ set -o noclobber
[trainee@centos8 ~]$ pwd > file
-bash: file: cannot overwrite existing file
[trainee@centos8 ~]$ pwd > file
-bash: file: cannot overwrite existing file
[trainee@centos8 ~]$ pwd >| file
[trainee@centos8 ~]$ set +o noclobber
```



**Important :** Notez que l'option **noclobber** peut être contournée en utilisant la redirection suivie par le caractère |.

### noglob

```
[trainee@centos8 ~]$ set -o noglob
[trainee@centos8 ~]$ echo *
*
[trainee@centos8 ~]$ set +o noglob
[trainee@centos8 ~]$ echo *
aac abc bca codes Desktop Documents Downloads errorlog file file1 Music Pictures Public Templates training Videos
vitext xyz
```



**Important :** Notez que l'effet du caractère spécial est annulé sous l'influence de l'option **noglob**.

### nounset

```
[trainee@centos8 ~]$ set -o nounset  
[trainee@centos8 ~]$ echo $FENESTROS  
-bash: FENESTROS: unbound variable  
[trainee@centos8 ~]$ set +o nounset  
[trainee@centos8 ~]$ echo $FENESTROS  
  
[trainee@centos8 ~]$
```



**Important :** Notez que la variable inexistante **\$FENESTROS** est identifiée comme telle sous l'influence de l'option **nounset**. Or le comportement habituel de Linux est de retourner une ligne vide qui n'indique pas si la variable n'existe pas ou si elle est simplement vide.

## LAB #2 - 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** :

```
[root@centos8 ~]# ls -lai /tmp
total 0
16800396 drwxrwxrwt. 8 root root 172 Apr 20 04:36 .
    128 dr-xr-xr-x. 17 root root 224 Apr 19 11:50 ..
25447488 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .font-unix
 8388741 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .ICE-unix
      143 drwx----- 3 root root 17 Apr 19 12:05 systemd-private-d9ff2376a8a44f0392f860d80c839be4-
chronyd.service-6im4Ii
 621976 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .Test-unix
 621954 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .X11-unix
17319048 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .XIM-unix
[root@centos8 ~]# ls -ali /tmp
total 0
16800396 drwxrwxrwt. 8 root root 172 Apr 20 04:36 .
    128 dr-xr-xr-x. 17 root root 224 Apr 19 11:50 ..
25447488 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .font-unix
 8388741 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .ICE-unix
      143 drwx----- 3 root root 17 Apr 19 12:05 systemd-private-d9ff2376a8a44f0392f860d80c839be4-
chronyd.service-6im4Ii
 621976 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .Test-unix
 621954 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .X11-unix
17319048 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .XIM-unix
[root@centos8 ~]# ls -ial /tmp
total 0
16800396 drwxrwxrwt. 8 root root 172 Apr 20 04:36 .
    128 dr-xr-xr-x. 17 root root 224 Apr 19 11:50 ..
25447488 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .font-unix
 8388741 drwxrwxrwt. 2 root root  6 Apr 19 11:37 .ICE-unix
```

```
143 drwx----- 3 root root 17 Apr 19 12:05 systemd-private-d9ff2376a8a44f0392f860d80c839be4-
chronyd.service-6im4Ii
621976 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .Test-unix
621954 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .X11-unix
17319048 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .XIM-unix
```

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

```
[root@centos8 ~]# ls -l --all --inode /tmp
total 0
16800396 drwxrwxrwt. 8 root root 172 Apr 20 04:36 .
128 dr-xr-xr-x. 17 root root 224 Apr 19 11:50 ..
25447488 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .font-unix
8388741 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .ICE-unix
143 drwx----- 3 root root 17 Apr 19 12:05 systemd-private-d9ff2376a8a44f0392f860d80c839be4-
chronyd.service-6im4Ii
621976 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .Test-unix
621954 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .X11-unix
17319048 drwxrwxrwt. 2 root root 6 Apr 19 11:37 .XIM-unix
[root@centos8 ~]# 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.

## LAB #3 - 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 <b>x</b> à <b>y</b> occurrences de ce qui précède
\{x\}	Trouver exactement le nombre <b>x</b> d'occurrences de ce qui précède
\{x,\}	Trouver le nombre <b>x</b> 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
+	Trouver 1 ou <b>n</b> d'occurrences de ce qui précède
{x,y}	Trouver de <b>x</b> à <b>y</b> occurrences de ce qui précède
{x}	Trouver exactement le nombre <b>x</b> d'occurrences de ce qui précède
{x,}	Trouver le nombre <b>x</b> ou plus d'occurrences de ce qui précède
()	Faire un <b>ET</b> des expressions régulières entre les parenthèses
	Faire un <b>OU</b> des expressions régulières se trouvant de chaque côté du pipe

## Outils de Manipulation de Fichiers Texte

### Présentation des Commandes grep, egrep et fgrep

#### 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 aussi être 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 #4 - Utiliser grep, egrep et fgrep

Créez le fichier **/tmp/greptest** :

```
[root@centos8 ~]# cd /tmp
[root@centos8 tmp]# vi greptest
[root@centos8 tmp]# cat greptest
fenestrOS
fenestros
555-5555
f
.fenestros
.fe
£
```

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

```
[root@centos8 ~]# grep '[a-zA-Z]' /tmp/greptest
fenestrOS
fenestros
f
.fenestros
.fe
```

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

```
[root@centos8 ~]# grep '[a-zA-Z0-9]' /tmp/greptest
fenestrOS
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 :

```
[root@centos8 ~]# grep '[0-9]\{3\}-[0-9]\{4\}' /tmp/greptest
555-5555
```

Recherchez maintenant toute ligne contenant exactement un caractère :

```
[root@centos8 ~]# 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 :

```
[root@centos8 ~]# grep '^\. ' /tmp/greptest
.fenestros
.fe
```



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



**Important :** 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** selon l'exemple ci-dessous :

```
[root@centos8 tmp]# vi greptest
[root@centos8 tmp]# cat greptest
# Starting comment
fenestrOS
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 :

```
[root@centos8 ~]# grep -E -v '^(#|$)' /tmp/greptest
fenestrOS
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** :

```
[root@centos8 ~]# egrep -v '^(#|$)' /tmp/greptest > /tmp/greptest1
[root@centos8 ~]# cat /tmp/greptest1
fenestrOS
fenestros
555-5555
f
.fenestros
.fe
£
```



**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** selon l'exemple ci-dessous :

```
[root@centos8 tmp]# vi greptest
[root@centos8 tmp]# cat greptest
# Starting comment
^ This line will be used to demonstrate the use of fgrep
fenestrOS
fenestros
# Another comment
555-5555
f

.fenestros

.fe

#
# End comment
```

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

```
[root@centos8 ~]# 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 :

```
[root@centos8 ~]# grep '^' /tmp/greptest
# Starting comment
^ This line will be used to demonstrate the use of fgrep
fenestrOS
fenestros
# Another comment
555-5555
f

.fenestros
```

```
.fe  
£  
# End comment
```

La ligne de commande en utilisant la commande grep devrait être :

```
[root@centos8 ~]# grep '^ '^ /tmp/greptest  
^ This line will be used to demonstrate the use of fgrep
```

## Présentation de la 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 <b>a</b>
\$	La dernière ligne
/ERb/	Les lignes qui correspondent à l'ERb
a,b	De la ligne numéro <b>a</b> jusqu'à la ligne numéro <b>b</b>
/ERb1/, /ERb2/	Toutes les lignes entre la première occurrence correspondant à l'ERb1 jusqu'à la première occurrence correspondant à l'ERb2

Le 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 #5 - 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 :

```
[root@centos8 ~]# sed '1,10d' /etc/services | more
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#      http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
```

```
# service-name  port/protocol  [aliases ...]  [# comment]

tcpmux          1/tcp           # TCP port service multiplexer
tcpmux          1/udp           # TCP port service multiplexer
rje             5/tcp           # Remote Job Entry
rje             5/udp           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
daytime         13/tcp
--Plus--
```

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

```
[root@centos8 ~]# sed '/^#/d' /etc/services | more

tcpmux          1/tcp           # TCP port service multiplexer
tcpmux          1/udp           # TCP port service multiplexer
rje             5/tcp           # Remote Job Entry
rje             5/udp           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
daytime         13/tcp
daytime         13/udp
qotd            17/tcp          quote
qotd            17/udp          quote
msp              18/tcp          # message send protocol
```

```
msp          18/udp          # message send protocol
chargen     19/tcp          ttystt source
chargen     19/udp          ttystt source
ftp-data    20/tcp
ftp-data    20/udp
ftp         21/tcp
ftp         21/udp          fsp  fspd
--Plus--
```



**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** :

```
[root@centos8 ~]# sed '1,2p' /etc/passwd
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
...
```



**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** :

```
[root@centos8 ~]# sed -n '1,2p' /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

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 :

```
[root@centos8 ~]# sed -n '/^#/!w /tmp/sedtest' /etc/services
[root@centos8 ~]# more /tmp/sedtest

tcpmux          1/tcp           # TCP port service multiplexer
tcpmux          1/udp           # TCP port service multiplexer
rje             5/tcp           # Remote Job Entry
rje             5/udp           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
daytime         13/tcp
daytime         13/udp
qotd            17/tcp           quote
qotd            17/udp           quote
msp              18/tcp           # message send protocol
msp              18/udp           # message send protocol
chargen         19/tcp           ttyst source
chargen         19/udp           ttyst source
ftp-data        20/tcp
ftp-data        20/udp
ftp              21/tcp
ftp              21/udp           fsp fsdp
--Plus-- (0%)
```

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

```
[root@centos8 ~]# echo "user1,user2,user3" > /tmp/sedtest1
[root@centos8 ~]# 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.

## Présentation de La Commande awk

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 :

```
[root@centos8 tmp]# ls -l | awk '{print $8 $3 $4}'  
05:23rootroot  
05:21rootroot  
05:28rootroot  
05:29rootroot  
12:05rootroot
```

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 :

```
[root@centos8 tmp]# ls -l | awk '{print $8 " " $3 " " $4}'  
05:23 root root  
05:21 root root  
05:28 root root  
05:29 root root  
12:05 root root
```

## 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
!=	Different
>	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 <b>espace</b> ou une <b>tabulation</b>
RS	Le séparateur de lignes en entrée. Par défaut une <b>nouvelle ligne</b>
OFS	Le séparateur de champs en sortie. Par défaut un <b>espace</b>
ORS	Le séparateur de lignes en sortie. Par défaut une <b>nouvelle ligne</b>
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 :

```
[root@centos8 tmp]# cat > scriptawk
BEGIN {
    print "Liste des systèmes de fichiers montés"
{print $0}
END {
```

```
print "=====\"\n[^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 :

```
[root@centos8 tmp]# awk -f scriptawk /etc/mtab
Liste des systèmes de fichiers montés
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=1897604k,nr_inodes=474401,mode=755 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev 0 0
devpts /dev/pts devpts rw,seclabel,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,seclabel,nosuid,nodev,mode=755 0 0
tmpfs /sys/fs/cgroup tmpfs ro,seclabel,nosuid,nodev,noexec,mode=755 0 0
cgroup /sys/fs/cgroup/systemd cgroup
rw,seclabel,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cggroups-agent,name=systemd
0 0
pstore /sys/fs/pstore pstore rw,seclabel,nosuid,nodev,noexec,relatime 0 0
bpf /sys/fs/bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,seclabel,nosuid,nodev,noexec,relatime,hugetlb 0 0
cgroup /sys/fs/cgroup/devices cgroup rw,seclabel,nosuid,nodev,noexec,relatime,devices 0 0
cgroup /sys/fs/cgroup/cpuset cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpuset 0 0
cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpu,cpuacct 0 0
cgroup /sys/fs/cgroup/net_cls,net_prio cgroup rw,seclabel,nosuid,nodev,noexec,relatime,net_cls,net_prio 0 0
cgroup /sys/fs/cgroup/rdma cgroup rw,seclabel,nosuid,nodev,noexec,relatime,rdma 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,seclabel,nosuid,nodev,noexec,relatime,freezer 0 0
cgroup /sys/fs/cgroup/perf_event cgroup rw,seclabel,nosuid,nodev,noexec,relatime,perf_event 0 0
```

```
cgroup /sys/fs/cgroup/pids cgroup rw,seclabel,nosuid,nodev,noexec,relatime,pids 0 0
cgroup /sys/fs/cgroup/blkio cgroup rw,seclabel,nosuid,nodev,noexec,relatime,blkio 0 0
cgroup /sys/fs/cgroup/memory cgroup rw,seclabel,nosuid,nodev,noexec,relatime,memory 0 0
none /sys/kernel/tracing tracefs rw,seclabel,relatime 0 0
configfs /sys/kernel/config configfs rw,relatime 0 0
/dev/sda3 / xfs rw,seclabel,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs
rw,relatime,fd=36,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=1976 0 0
debugfs /sys/kernel/debug debugfs rw,seclabel,relatime 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime,pagesize=2M 0 0
mqueue /dev/mqueue mqueue rw,seclabel,relatime 0 0
/dev/sda1 /boot ext4 rw,seclabel,relatime 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
tmpfs /run/user/1000 tmpfs rw,seclabel,nosuid,nodev,relatime,size=382740k,mode=700,uid=1000,gid=1000 0 0
binfmt_misc /proc/sys/fs/binfmt_misc binfmt_misc rw,relatime 0 0
tracefs /sys/kernel/debug/tracing tracefs rw,seclabel,relatime 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 ("chaine",expression1,expression2,...,expressionn)
```

**chaine** 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 #6 - Utiliser la Commande awk

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

```
[root@centos8 tmp]# vi sales.txt
[root@centos8 tmp]# cat sales.txt
# 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 **sales.awk** :

```
[root@centos8 tmp]# vi sales.awk
[root@centos8 tmp]# cat 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. Le 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**.
- 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 **sales.txt** :

```
[root@centos8 tmp]# 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
```

## LAB #7 -Autres Commandes Utiles

### 7.1 - 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 **expand** :

```
[root@centos8 tmp]# vi expand
[root@centos8 tmp]# cat expand
```

```
un    deux    trois   quatre   cinq
```

&gt;

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

```
[root@centos8 tmp]# cat -vet expand
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** :

```
[root@centos8 ~]# expand expand > expand1
```

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

```
[root@centos8 ~]# cat -vet expand1
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.

#### 7.2 - 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** :

```
[root@centos8 ~]# cat -vet expand1
un      deux      trois      quatre      cinq$

[root@centos8 ~]# unexpand -a expand1 > expand2

[root@centos8 ~]# cat -vet expand2
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.

### 7.3 - 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 :

```
[root@centos8 tmp]# cut -c1-7 /etc/passwd
root:x:
bin:x:1
daemon:
adm:x:3
lp:x:4:
sync:x:
shutdow
halt:x:
mail:x:
operato
games:x
ftp:x:1
nobody:
dbus:x:
```

```
systemd
systemd
tss:x:5
polkitd
unbound
libstor
cockpit
sssd:x:
setroub
sshd:x:
chrony:
tcpdump
trainee
cockpit
rngd:x:
gluster
qemu:x:
rpc:x:3
rpcuser
saslaut
radvd:x
dnsmasq
```

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 :

```
[root@centos8 tmp]# cut -c1-5,10-15,30- /etc/passwd
root:0:rootsh
bin:x:bin:/gin
daemo2:2:dain/nologin
adm:x:adm:/nologin
lp:x:lp:/vabin/nologin
sync:0:syncnc
shutdx:6:0::/sbin/shutdown
halt:0:haltalt
```

```
mail:12:maiai:/sbin/nologin
operax:11:0t:/sbin/nologin
games2:100:es:/sbin/nologin
ftp:x50:FTP:/sbin/nologin
nobod65534:verflow User:/:/sbin/nologin
dbus::81:Syus:/:/sbin/nologin
systeoredumstemd Core Dumper:/:/sbin/nologin
systeesolvetemd Resolver:/:/sbin/nologin
tss:x59:Acche trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polki:998:9lkitd:/:/sbin/nologin
unbou:997:9 resolver:/etc/unbound:/sbin/nologin
libstremgmt:on account for libstoragemgmt:/var/run/lsm:/sbin/nologin
cockps:x:99 cockpit-ws:/nonexisting:/sbin/nologin
sssd:4:990:/:/sbin/nologin
setroshoot:r/lib/setroubleshoot:/sbin/nologin
sshd::74:Prtd SSH:/var/empty/sshd:/sbin/nologin
chron992:98rony:/sbin/nologin
tcpdu:72:72gin
train:1000:home/trainee:/bin/bash
cockpsinstaUser for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:0:986:Generator Daemon:/var/lib/rngd:/sbin/nologin
glust:989:9aemons:/run/gluster:/sbin/nologin
qemu:7:107:bin/nologin
rpc:x32:Rpcar/lib/rpcbind:/sbin/nologin
rpcus:29:29ser:/var/lib/nfs:/sbin/nologin
saslax:988:ser:/run/saslauthd:/sbin/nologin
radvd5:75:rbin/nologin
dnsma:983:9P and DNS server:/var/lib/dnsmasq:/sbin/nologi
```

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

```
[root@centos8 tmp]# cut -d: -f2,4,6 /etc/passwd
x:0:/root
x:1:/bin
```

```
x::/sbin
x::/var/adm
x::/var/spool/lpd
x::/sbin
x::/sbin
x::/sbin
x::/var/spool/mail
x::/root
x::100:/usr/games
x::50:/var/ftp
x::65534:/
x::81:/
x::997:/
x::193:/
x::59:/dev/null
x::996:/
x::994:/etc/unbound
x::993:/var/run/lsm
x::991:/nonexisting
x::990:/
x::989:/var/lib/setroubleshoot
x::74:/var/empty/sshd
x::988:/var/lib/chrony
x::72:/
x::1000:/home/trainee
x::987:/nonexisting
x::986:/var/lib/rngd
x::985:/run/gluster
x::107:/
x::32:/var/lib/rpcbind
x::29:/var/lib/nfs
x::76:/run/saslauthd
x::75:/
x::983:/var/lib/dnsmasq
```

## Options de la commande



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

### 7.4 - La Commande uniq

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

```
[root@centos8 tmp]# cut -d: -f4 /etc/passwd | sort -n | uniq
0
1
2
4
7
12
29
32
50
59
72
74
75
76
81
100
107
193
983
985
```

```
986
987
988
989
990
991
993
994
996
997
1000
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.

#### 7.5 - 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.

```
[root@centos8 tmp]# cat /etc/passwd | tr "[a-z]" "[A-Z]"
ROOT:X:0:0:ROOT:/ROOT:/BIN/BASH
BIN:X:1:1:BIN:/BIN:/SBIN/NOLOGIN
DAEMON:X:2:2:DAEMON:/SBIN:/SBIN/NOLOGIN
ADM:X:3:4:ADM:/VAR/ADM:/SBIN/NOLOGIN
LP:X:4:7:LP:/VAR/SPPOOL/LPD:/SBIN/NOLOGIN
SYNC:X:5:0:SYNC:/SBIN:/BIN/SYNC
SHUTDOWN:X:6:0:SHUTDOWN:/SBIN:/SBIN/SHUTDOWN
HALT:X:7:0:HALT:/SBIN:/SBIN/HALT
MAIL:X:8:12:MAIL:/VAR/SPPOOL/MAIL:/SBIN/NOLOGIN
OPERATOR:X:11:0:OPERATOR:/ROOT:/SBIN/NOLOGIN
GAMES:X:12:100:GAMES:/USR/GAMES:/SBIN/NOLOGIN
FTP:X:14:50:FTP USER:/VAR/FTP:/SBIN/NOLOGIN
NOBODY:X:65534:65534:KERNEL OVERFLOW USER:/:/SBIN/NOLOGIN
DBUS:X:81:81:SYSTEM MESSAGE BUS:/:/SBIN/NOLOGIN
SYSTEMD-COREDUMP:X:999:997:SYSTEMD CORE DUMPER:/:/SBIN/NOLOGIN
SYSTEMD-RESOLVE:X:193:193:SYSTEMD RESOLVER:/:/SBIN/NOLOGIN
TSS:X:59:59:ACCOUNT USED BY THE TROUSERS PACKAGE TO SANDBOX THE TCSD DAEMON:/DEV/NULL:/SBIN/NOLOGIN
POLKITD:X:998:996:USER FOR POLKITD:/:/SBIN/NOLOGIN
UNBOUND:X:997:994:UNBOUND DNS RESOLVER:/ETC/UNBOUND:/SBIN/NOLOGIN
LIBSTORAGEMGMT:X:996:993:DAEMON ACCOUNT FOR LIBSTORAGEMGMT:/VAR/RUN/LSM:/SBIN/NOLOGIN
COCKPIT-WS:X:995:991:USER FOR COCKPIT-WS:/NONEXISTING:/SBIN/NOLOGIN
SSSD:X:994:990:USER FOR SSSD:/:/SBIN/NOLOGIN
SETROUBLESHOOT:X:993:989:/:/VAR/LIB/SETROUBLESHOOT:/SBIN/NOLOGIN
SSHD:X:74:74:PRIVILEGE-SEPARATED SSH:/VAR/EMPTY/SSHD:/SBIN/NOLOGIN
CHRONY:X:992:988:/:/VAR/LIB/CHRONY:/SBIN/NOLOGIN
TCPDUMP:X:72:72:/:/SBIN/NOLOGIN
TRAINEE:X:1000:1000:TRAINEE:/HOME/TRAINEE:/BIN/BASH
COCKPIT-WSINSTANCE:X:991:987:USER FOR COCKPIT-WS INSTANCES:/NONEXISTING:/SBIN/NOLOGIN
RNGD:X:990:986:RANDOM NUMBER GENERATOR DAEMON:/VAR/LIB/RNGD:/SBIN/NOLOGIN
GLUSTER:X:989:985:GLUSTERFS DAEMONS:/RUN/GLUSTER:/SBIN/NOLOGIN
QEMU:X:107:107:QEMU USER:/:/SBIN/NOLOGIN
RPC:X:32:32:RPCBIND DAEMON:/VAR/LIB/RPCBIND:/SBIN/NOLOGIN
RPCUSER:X:29:29:RPC SERVICE USER:/VAR/LIB/NFS:/SBIN/NOLOGIN
```

```
SASLAUTH:X:988:76:SASLAUTHD USER:/RUN/SASLAUTHD:/SBIN/NOLOGIN
RADVD:X:75:75:RADVD USER::/SBIN/NOLOGIN
DNSMASQ:X:983:983:DNSMASQ DHCP AND DNS SERVER:/VAR/LIB/DNSMASQ:/SBIN/NOLOGIN
```

## Options de la commande



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

## 7.6 - La Commande paste

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

```
[root@centos8 tmp]# paste -d: /etc/passwd /etc/shadow
root:x:0:0:root:/root:/bin/bash:root:$6$9Sa1IumuS1Jc8EBg$8jGU/4xGCXy64QuBSMyK0C6/FWs41rdY5tzF5/7yHG6FRS2Y2e0JIcst
1JbcvNoqMPDU4lpZ6THW97jwGuQNf1::0:99999:7:::
bin:x:1:1:bin:/bin/nologin:bin:*:18264:0:99999:7:::
daemon:x:2:2:daemon:/sbin/nologin:daemon:*:18264:0:99999:7:::
adm:x:3:4:adm:/var/adm:/sbin/nologin:adm:*:18264:0:99999:7:::
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin:lp:*:18264:0:99999:7:::
sync:x:5:0:sync:/sbin/bin/sync:sync:*:18264:0:99999:7:::
shutdown:x:6:0:shutdown:/sbin/sbin/shutdown:shutdown:*:18264:0:99999:7:::
halt:x:7:0:halt:/sbin/sbin/halt:halt:*:18264:0:99999:7:::
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin:mail:*:18264:0:99999:7:::
operator:x:11:0:operator:/root:/sbin/nologin:operator:*:18264:0:99999:7:::
games:x:12:100:games:/usr/games:/sbin/nologin:games:*:18264:0:99999:7:::
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin:ftp:*:18264:0:99999:7:::
nobody:x:65534:65534:Kernel Overflow User::/sbin/nologin:nobody:*:18264:0:99999:7:::
dbus:x:81:81:System message bus::/sbin/nologin:dbus::18390:::::
```

```
systemd-coredump:x:999:997:systemd Core Dumper::/sbin/nologin:systemd-coredump:!!!:18390:::::  
systemd-resolve:x:193:193:systemd Resolver::/sbin/nologin:systemd-resolve:!!!:18390:::::  
tss:x:59:59:Account used by the trousers package to sandbox the tcscd  
daemon:/dev/null:/sbin/nologin:tss:!!!:18390:::::  
polkitd:x:998:996:User for polkitd::/sbin/nologin:polkitd:!!!:18390:::::  
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin:unbound:!!!:18390:::::  
libstoragemgmt:x:996:993:daemon account for  
libstoragemgmt:/var/run/lsm:/sbin/nologin:libstoragemgmt:!!!:18390:::::  
cockpit-ws:x:995:991:User for cockpit-ws:/nonexisting:/sbin/nologin:cockpit-ws:!!!:18390:::::  
sssd:x:994:990:User for sssd::/sbin/nologin:sssd:!!!:18390:::::  
setroubleshoot:x:993:989::/var/lib/setroubleshoot:/sbin/nologin:setroubleshoot:!!!:18390:::::  
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin:sshd:!!!:18390:::::  
chrony:x:992:988::/var/lib/chrony:/sbin/nologin:chrony:!!!:18390:::::  
tcpdump:x:72:72::/sbin/nologin:tcpdump:!!!:18390:::::  
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash:trainee:$6$p4H0AHX7UAzw1nQh$VZL12Lye.mR8v1IP2e4f0PCW8DzHj2MMA  
aA7r2ZLoTnQN7Ziskce3bo/xTMu1bXZm5GebJjSw7.X5tABVNoJ2/::0:99999:7:::  
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin:cockpit-  
wsinstance:!!!:18736:::::  
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin:rngd:!!!:18736:::::  
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin:gluster:!!!:18736:::::  
qemu:x:107:107:qemu user::/sbin/nologin:qemu:!!!:18736:::::  
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin:rpc:!!!:18736:0:99999:7:::  
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin:rpcuser:!!!:18736:::::  
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin:saslauth:!!!:18736:::::  
radvd:x:75:75:radvd user::/sbin/nologin:radvd:!!!:18736:::::  
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin:dnsmasq:!!!:18736:::::
```

## Options de la commande



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

## 7.7 - 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 :

```
[root@centos8 tmp]# dd if=/dev/zero of=/file bs=1024k count=250
250+0 records in
250+0 records out
262144000 bytes (262 MB, 250 MiB) copied, 0.143522 s, 1.8 GB/s
```

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

```
[root@centos8 tmp]# split -b 50m /file filepart
[root@centos8 tmp]# ls -l | grep filepart
-rw-r--r--. 1 root root 52428800 Apr 20 07:14 filepartaa
-rw-r--r--. 1 root root 52428800 Apr 20 07:14 filepartab
-rw-r--r--. 1 root root 52428800 Apr 20 07:14 filepartac
-rw-r--r--. 1 root root 52428800 Apr 20 07:14 filepartad
-rw-r--r--. 1 root root 52428800 Apr 20 07:14 filepartae
```



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

Reconstruisez simplement le fichier avec la commande cat :

```
[root@centos8 tmp]# cat fileparta* > newfile
[root@centos8 tmp]# ls -l | grep newf
-rw-r--r--. 1 root root 262144000 Apr 20 07:15 newfile
[root@centos8 tmp]# ls -l / | grep file
```

```
-rw-r--r--. 1 root root 262144000 Apr 20 07:14 file
```

### Options de la commande



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

### 7.8 - La Commande diff

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

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

```
[root@centos8 ~]# 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 **tcpdump** dans le fichier **/root/passwd** et ajoutez en fin de fichier la ligne suivante :

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

Comparez maintenant les deux fichiers :

```
[root@centos8 tmp]# diff /etc/passwd /root/passwd
```

```
26,27c26
< tcpdump:x:72:72:::/sbin/nologin
< trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
---
> trainee10:x:1000:1000:trainee:/home/trainee:/bin/bash
36a36
> 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 **26,27c26** indique qu'il faut changer la ligne 27 dans /etc/passwd afin que celle-ci corresponde à la ligne 26 dans /root/passwd.

Le message **36a36** indique qu'à la ligne 36 dans /etc/passwd il faut ajouter la ligne 36 de /root/passwd.

#### Options de la commande



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

#### 7.9 - La Commande **cmp**

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

```
[root@centos8 tmp]# cmp /root/passwd /etc/passwd
/root/passwd /etc/passwd differ: byte 1300, line 26
```

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

```
[root@centos8 tmp]# cmp -l /root/passwd /etc/passwd | more
cmp: EOF on /root/passwd after byte 1931
1300 162 143
1301 141 160
1302 151 144
1303 156 165
1304 145 155
1305 145 160
1306 61 72
1307 60 170
1309 170 67
1310 72 62
1311 61 72
1312 60 67
1313 60 62
1314 60 72
1316 61 57
1317 60 72
1318 60 57
1319 60 163
1320 72 142
1321 164 151
1322 162 156
1323 141 57
1324 151 156
--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.

## 7.10 - 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.

La commande patch n'est pas installée par défaut sous RHEL/CentOS 8 :

```
[root@centos8 ~]# dnf install patch -y
```

Rappelez-vous maintenant des modifications apportées aux fichiers /tmp/greptest et /tmp/greptest1 :

```
[root@centos8 tmp]# cat /tmp/greptest
# Starting comment
^ This line will be used to demonstrate the use of fgrep
fenestrOS
fenestros
# Another comment
555-5555
f

.fenestros

.fe

#
# End comment
[root@centos8 tmp]# cat /tmp/greptest1
```

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

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

```
[root@centos8 tmp]# diff -u greptest greptest1 > greptest.patch
```

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

```
[root@centos8 tmp]# cat greptest.patch
--- greptest      2021-04-20 05:23:52.710188632 -0400
+++ greptest1     2021-04-20 05:21:55.189882834 -0400
@@ -1,14 +1,7 @@
-# Starting comment
-^ This line will be used to demonstrate the use of fgrep
 fenestrOS
 fenestros
-# Another comment
 555-5555
 f
-
 .fenestros
-
 .fe
-
 £
-# End comment
```

Procédez maintenant à l'application du fichier patch :

```
[root@centos8 tmp]# patch < greptest.patch
patching file greptest
```

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

```
[root@centos8 tmp]# cat greptest
fenestrOS
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.

### 7.11 - 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 :

```
[root@centos8 tmp]# strings /usr/bin/passwd | more
/lib64/ld-linux-x86-64.so.2
libuser.so.1
g_value_get_int64
is_selinux_enabled
_ITM_deregisterTMCloneTable
g_free
g_value_array_get_nth
audit_open
__gmon_start__
g_value_get_string
g_type_check_value_holds
g_value_get_long
freecon
audit_log_acct_message
_ITM_registerTMCloneTable
lu_ent_set_string
lu_ent_get_first_value_strdup
lu_error_free
lu_user_lock
lu_strerror
lu_ent_free
lu_ent_new
lu_user_modify
--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 :

```
[root@centos8 tmp]# strings -t d /usr/bin/passwd | more
 624 /lib64/ld-linux-x86-64.so.2
 2809 libuser.so.1
```

```

2822 g_value_get_int64
2840 is_selinux_enabled
2859 _ITM_deregisterTMCloneTable
2887 g_free
2894 g_value_array_get_nth
2916 audit_open
2927 __gmon_start__
2942 g_value_get_string
2961 g_type_check_value_holds
2986 g_value_get_long
3003 freecon
3011 audit_log_acct_message
3034 _ITM_registerTMCloneTable
3060 lu_ent_set_string
3078 lu_ent_get_first_value_strdup
3108 lu_error_free
3122 lu_user_lock
3135 lu_strerror
3147 lu_ent_free
3159 lu_ent_new
3170 lu_user_modify
--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 :

```
[root@centos8 tmp]# strings -t d -n 15 /usr/bin/passwd | more
624 /lib64/ld-linux-x86-64.so.2
```

```
2822 g_value_get_int64
2840 is_selinux_enabled
2859 _ITM_deregisterTMCloneTable
2894 g_value_array_get_nth
2942 g_value_get_string
2961 g_type_check_value_holds
2986 g_value_get_long
3011 audit_log_acct_message
3034 _ITM_registerTMCloneTable
3060 lu_ent_set_string
3078 lu_ent_get_first_value_strdup
3185 lu_prompt_console
3212 lu_user_lookup_name
3239 lu_ent_set_long
3281 lu_user_removepass
3300 libgobject-2.0.so.0
3320 libglib-2.0.so.0
3379 poptHelpOptions
3435 poptSetOtherOptionHelp
3543 libpam_misc.so.0
3584 audit_log_user_avc_message
3611 libselinux.so.1
```

--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 :

```
[root@centos8 tmp]# strings -f /bin/* | grep "(c)"
/bin/broadwayd: * (c) Joel Martin (github@martintribe.org), used with permission
/bin/broadwayd: /** @license zlib.js 2012 - imaya [ https://github.com/imaya/zlib.js ] The MIT License
*/(function() {'use strict';var l=void 0,p=this;function q(c,d){var a=c.split("."),b=p;! (a[0]in
b)&&b.execScript&&b.execScript("var "+a[0]);for(var
e;a.length&&(e=a.shift()));!a.length&&d!==l?b[e]=d:b=b[e]?b[e]:b[e]={};var r="undefined"!==typeof
Uint8Array&&"undefined"!==typeof Uint16Array&&"undefined"!==typeof Uint32Array;function u(c){var
```

```

d=c.length,a=0,b=Number.POSITIVE_INFINITY,e,f,g,h,k,m,s,n,t;for(n=0;n<d;++n)c[n]>a&&(a=c[n]),c[n]<b&&(b=c[n]);e=1
<<a;f=new
(r?Uint32Array:Array)(e);g=1;h=0;for(k=2;g<=a;){for(n=0;n<d;++n)if(c[n]===g){m=0;s=h;for(t=0;t<g;++t)m=m<<1|s&1,s
>>=1;for(t=m;t<e;t+=k)f[t]=g<<16|n;++h}++g;h<<=1;k<<=1}return[f,a,b]};function
v(c,d){this.g=[];this.h=32768;this.c=this.f=this.d=this.k=0;this.input=r?new
Uint8Array(c):c;this.l=!1;this.i=w;this.p=!1;if(d||!(d={}))d.index&&(this.d=d.index),d.bufferSize&&(this.h=d.buff
erSize),d.bufferType&&(this.i=d.bufferType),d.resize&&(this.p=d.resize);switch(this.i){case
x:this.a=32768;this.b=new (r?Uint8Array:Array)(32768+this.h+258);break;case w:this.a=0;this.b=new
(r?Uint8Array:Array)(this.h);this.e=this.u;this.m=this.r;this.j=this.s;break;default:throw Error("invalid inflate
mode");
/bin/broadwayd: v.prototype.t=function(){for(;!this.l;){var c=y(this,3);c&1&&(this.l!=0);c>>>=1;switch(c){case
0:var d=this.input,a=this.d,b=this.b,e=this.a,f=l,g=l,h=l,k=b.length,m=l;this.c=this.f=0;f=d[a++];if(f==l)throw
Error("invalid uncompressed block header: LEN (first byte)");g=f;f=d[a++];if(f==l)throw Error("invalid
uncompressed block header: LEN (second byte)");g|=f<<8;f=d[a++];if(f==l)throw Error("invalid uncompressed block
header: NLEN (first byte)");h=f;f=d[a++];if(f==l)throw Error("invalid uncompressed block header: NLEN (second
byte)");h|=_
/bin/broadwayd: function B(c){function d(a,c,b){var d,f,e,g;for(g=0;g<a;)switch(d=S(this,c),d){case
16:for(e=3+y(this,2);e--;)b[g++]=f;break;case 17:for(e=3+y(this,3);e--;)b[g++]=0;f=0;break;case
18:for(e=11+y(this,7);e--;)b[g++]=0;f=0;break;default:f=b[g++]=d}return b}var
a=y(c,5)+257,b=y(c,5)+1,e=y(c,4)+4,f=new
(r?Uint8Array:Array)(D.length),g,h,k,m;for(m=0;m<e;++m)f[D[m]]=y(c,3);g=u(f);h=new (r?Uint8Array:Array)(a);k=new
(r?Uint8Array:Array)(b);c.j(u(d.call(c,a,g,h)),u(d.call(c,b,g,k)))}
/bin/broadwayd: v.prototype.e=function(){var c=new
(r?Uint8Array:Array)(this.a-32768),d=this.a-32768,a,b,e=this.b;if(r)c.set(e.subarray(32768,c.length));else{a=0;fo
r(b=c.length;a<b;++a)c[a]=e[a+32768]}this.g.push(c);this.k+=c.length;if(r)e.set(e.subarray(d,d+32768));else
for(a=0;32768>a;++a)e[a]=e[d+a];this.a=32768;return e};
/bin/broadwayd: v.prototype.u=function(c){var
d,a=this.input.length/this.d+1|0,b,e,f,g=this.input,h=this.b;c&&("number"==typeof c.o&&(a=c.o),"number"==typeof
c.q&&(a+=c.q));2>a?(b=(g.length-
this.d)/this.n[2],f=258*(b/2)|0,e=f<h.length?h.length+f:h.length<<1):e=h.length*a;r?(d=new
Uint8Array(e),d.set(h)):d=h;return this.b=d};
/bin/btrace: # Copyright (c) 2005 Silicon Graphics, Inc.
/bin/chcat:           if len(c) > 0 and (c[0] == "+" or c[0] == "-"):
/bin/chcat:           if len(c) > 0 and c[0] == "+":

```

```
/bin/chcat:           if len(c) > 0 and c[0] == "-":  
/bin/gprof: @(#) Copyright (c) 1983 Regents of the University of California.  
/bin/lsusb.py: # Copyright (c) 2009 Kurt Garloff <garloff@suse.de>  
/bin/lsusb.py: # Copyright (c) 2013 Kurt Garloff <kurt@garloff.de>  
/bin/pinentry: # Copyright (c) 2006 SUSE LINUX Products GmbH, Nuernberg, Germany.  
/bin/pinentry: # Copyright (c) 2009 Fedora Project  
/bin/pinentry: # Copyright (c) 2014-2015 Red Hat  
/bin/pkgconf: Copyright (c) 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018  
/bin/pkg-config: Copyright (c) 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018  
/bin/pod2usage: # Copyright (c) 1996-2000 by Bradford Appleton. All rights reserved.  
/bin/pod2usage: # Copyright (c) 2001-2016 by Marek Rouchal.  
/bin/qemu-img: Copyright (c) 2003-2019 Fabrice Bellard and the QEMU Project developers  
/bin/qemu-img: Copyright (c) 2003-2019 Fabrice Bellard and the QEMU Project developers  
/bin/qemu-io: Copyright (c) 2003-2019 Fabrice Bellard and the QEMU Project developers  
/bin/qemu-nbd: Copyright (c) 2003-2019 Fabrice Bellard and the QEMU Project developers  
/bin/qemu-pr-helper: Copyright (c) 2003-2019 Fabrice Bellard and the QEMU Project developers  
/bin/rescan-scsi-bus.sh: # (c) 1998--2010 Kurt Garloff <kurt@garloff.de>, GNU GPL v2 or v3  
/bin/rescan-scsi-bus.sh: # (c) 2006--2018 Hannes Reinecke, GNU GPL v2 or later  
/bin/rngtest: Copyright (c) 2004 by Henrique de Moraes Holschuh  
/bin/screen: Copyright (c) 2015-2017 Juergen Weigert, Alexander Naumov, Amadeusz Slawinski  
/bin/screen: Copyright (c) 2010-2014 Juergen Weigert, Sadrul Habib Chowdhury  
/bin/screen: Copyright (c) 2008-2009 Juergen Weigert, Michael Schroeder, Micah Cowan, Sadrul Habib Chowdhury  
/bin/screen: Copyright (c) 1993-2007 Juergen Weigert, Michael Schroeder  
/bin/screen: Copyright (c) 1987 Oliver Laumann  
/bin/scsi-rescan: # (c) 1998--2010 Kurt Garloff <kurt@garloff.de>, GNU GPL v2 or v3  
/bin/scsi-rescan: # (c) 2006--2018 Hannes Reinecke, GNU GPL v2 or later  
/bin/sg_test_rwbuf: (c) Douglas Gilbert, Kurt Garloff, 2000-2007, GNU GPL  
/bin/slabinfo: slabinfo 4/15/2011. (c) 2007 sgi/(c) 2011 Linux Foundation.  
/bin/ssh-copy-id: # Copyright (c) 1999-2016 Philip Hands <phil@hands.com>  
/bin/strace: Copyright (c) 1991-%s The strace developers <%s>.  
/bin/strace-log-merge: # Copyright (c) 2012-2019 The strace developers.  
/bin/systemd-analyze: hashmap_update(*uid_refs, UID_TO_PTR(uid), UINT32_TO_PTR(c)) >= 0  
/bin/tree: $Version: $ tree v1.7.0 (c) 1996 - 2014 by Steve Baker, Thomas Moore, Francesc Rocher, Florian Sesser,  
Kyosuke Tokoro $
```

```
/bin/usb-devices: # Copyright (c) 2009 Greg Kroah-Hartman <greg@kroah.com>
/bin/usb-devices: # Copyright (c) 2009 Randy Dunlap <rdunlap@xenotime.net>
/bin/usb-devices: # Copyright (c) 2009 Frans Pop <elendil@planet.nl>
/bin/vdo: # Copyright (c) 2020 Red Hat, Inc.
/bin/vdo-by-dev: # Copyright (c) 2020 Red Hat, Inc.
/bin/vdostats: # Copyright (c) 2020 Red Hat, Inc.
/bin/zip: Copyright (c) 1990-2008 Info-ZIP. All rights reserved.
/bin/zip: Copyright (c) 1990-2008 Info-ZIP - Type '%s "-L"' for software license.
/bin/zip: bzip2 code and library copyright (c) Julian      (See the bzip2 license for t
/bin/zipcloak: Copyright (c) 1990-2008 Info-ZIP. All rights reserved.
/bin/zipcloak: Copyright (c) 1990-2008 Info-ZIP - Type '%s "-L"' for software license.
/bin/zipnote: Copyright (c) 1990-2008 Info-ZIP. All rights reserved.
/bin/zipnote: Copyright (c) 1990-2008 Info-ZIP - Type '%s "-L"' for software license.
/bin/zipsplit: Copyright (c) 1990-2008 Info-ZIP. All rights reserved.
/bin/zipsplit: Copyright (c) 1990-2008 Info-ZIP - Type '%s "-L"' for software license.
```



**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.

## 7.12 - 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 les deux fichiers.

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

```
[root@centos8 tmp]# comm /etc/passwd /root/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
libstoragemgmt:x:996:993:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
cockpit-ws:x:995:991:User for cockpit-ws:/nonexisting:/sbin/nologin
sssd:x:994:990:User for sssd:/:/sbin/nologin
setroubleshoot:x:993:989::/var/lib/setroubleshoot:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:992:988::/var/lib/chrony:/sbin/nologin
```

```
tcpdump:x:72:72::/:sbin/nologin
trainee10:x:1000:1000:trainee:/home/trainee:/bin/bash
comm: file 2 is not in sorted order
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
Linux est super!
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
comm: file 1 is not in sorted order
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

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

```
[root@centos8 tmp]# comm -12 /etc/passwd /root/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
```

```
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
libstoragemgmt:x:996:993:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
cockpit-ws:x:995:991:User for cockpit-ws:/nonexisting:/sbin/nologin
sssd:x:994:990:User for sssd:/:/sbin/nologin
setroubleshoot:x:993:989::/var/lib/setroubleshoot:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:992:988::/var/lib/chrony:/sbin/nologin
comm: file 2 is not in sorted order
comm: file 1 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.

#### 7.13 - 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 :

```
[root@centos8 tmp]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

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

```
[root@centos8 tmp]# head -n 15 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
```

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

```
[root@centos8 tmp]# head -c 150 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7[root@centos8 tmp]#
```

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

```
[root@centos8 tmp]# head -c -150 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
libstoragemgmt:x:996:993:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
cockpit-ws:x:995:991:User for cockpit-ws:/nonexisting:/sbin/nologin
sssd:x:994:990:User for sssd:/sbin/nologin
setroubleshoot:x:993:989::/var/lib/setroubleshoot:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
```

```
chrony:x:992:988::/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd us[root@centos8 tmp]#
```

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

```
[root@centos8 tmp]# head -c 1b /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow [root@centos8 tmp]#
```

```
[root@centos8 tmp]# head -c 512 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow [root@centos8 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.

#### 7.14 - 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 :

```
[root@centos8 tmp]# tail /etc/passwd
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

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

```
root@centos8 tmp]# tail -n 15 /etc/passwd
sssd:x:994:990:User for sssd:/sbin/nologin
setroubleshoot:x:993:989::/var/lib/setroubleshoot:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:992:988::/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

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

```
[root@centos8 tmp]# tail -c 150 /etc/passwd
```

```
er:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

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

```
[root@centos8 tmp]# tail -c +150 /etc/passwd
7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
libstoragemgmt:x:996:993:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
cockpit-ws:x:995:991:User for cockpit-ws:/nonexisting:/sbin/nologin
sssd:x:994:990:User for sssd:/sbin/nologin
setroubleshoot:x:993:989:/var/lib/setroubleshoot:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:992:988:/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
trainee:x:1000:1000:trainee:/home/trainee:/bin/bash
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
```

```
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

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

```
[root@centos8 tmp]# tail -c 1b /etc/passwd
instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

```
[root@centos8 tmp]# tail -c 512 /etc/passwd
instances:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
gluster:x:989:985:GlusterFS daemons:/run/gluster:/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
saslauth:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
dnsmasq:x:983:983:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

Les multiplicateurs les plus utilisés sont :

Multiplicateur	Valeur en octets
b	512

Multiplicateur	Valeur en octets
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 au mesure que le fichier est mis à jour :

```
[root@centos8 tmp]# tail -f /var/log/messages
Apr 20 06:27:53 centos8 systemd[1]: Started dnf makecache.
Apr 20 07:28:29 centos8 systemd[1]: Starting dnf makecache...
Apr 20 07:28:29 centos8 dnf[12423]: Metadata cache refreshed recently.
Apr 20 07:28:29 centos8 systemd[1]: dnf-makecache.service: Succeeded.
Apr 20 07:28:29 centos8 systemd[1]: Started dnf makecache.
Apr 20 07:50:35 centos8 systemd[1]: Started /usr/bin/systemctl start man-db-cache-update.
Apr 20 07:50:35 centos8 systemd[1]: Starting man-db-cache-update.service...
Apr 20 07:50:36 centos8 systemd[1]: man-db-cache-update.service: Succeeded.
Apr 20 07:50:36 centos8 systemd[1]: Started man-db-cache-update.service.
Apr 20 07:50:36 centos8 systemd[1]: run-r85917a32bc86476980c271609ba457fb.service: Succeeded.
^C
```

## Options de la commande



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

## LAB #8 - Utiliser les commandes ifconfig, grep, tr et cut pour isoler l'adresse IPv4

```
[root@centos8 tmp]# ifconfig ens18
ens18: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.45 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::86b6:8d39:cab2:d84d prefixlen 64 scopeid 0x20<link>
            ether 4e:b1:31:bd:5d:b2 txqueuelen 1000 (Ethernet)
            RX packets 11473 bytes 24023891 (22.9 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 9521 bytes 1744650 (1.6 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@centos8 tmp]# ifconfig ens18 | grep "inet"
    inet 10.0.2.45 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::86b6:8d39:cab2:d84d prefixlen 64 scopeid 0x20<link>

[root@centos8 tmp]# ifconfig ens18 | grep "inet" | grep -v "inet6"
    inet 10.0.2.45 netmask 255.255.255.0 broadcast 10.0.2.255

[root@centos8 tmp]# ifconfig ens18 | grep "inet" | grep -v "inet6" | tr -s " " ":" 
:inet:10.0.2.45:netmask:255.255.255.0:broadcast:10.0.2.255

[root@centos8 tmp]# ifconfig ens18 | grep "inet" | grep -v "inet6" | tr -s " " ":" | cut -d: -f3
10.0.2.45
```



**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.

## LAB #9 - Utiliser les commandes ip, grep, awk et sed pour isoler l'adresse IPv4

```
[root@centos8 tmp]# ip addr show ens18
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 4e:b1:31:bd:5d:b2 brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.45/24 brd 10.0.2.255 scope global noprefixroute ens18
            valid_lft forever preferred_lft forever
        inet6 fe80::86b6:8d39:cab2:d84d/64 scope link noprefixroute
            valid_lft forever preferred_lft forever

[root@centos8 tmp]# ip addr show ens18 | grep "inet"
    inet 10.0.2.45/24 brd 10.0.2.255 scope global noprefixroute ens18
    inet6 fe80::86b6:8d39:cab2:d84d/64 scope link noprefixroute

[root@centos8 tmp]# ip addr show ens18 | grep "inet" | grep -v "inet6"
    inet 10.0.2.45/24 brd 10.0.2.255 scope global noprefixroute ens18

[root@centos8 tmp]# ip addr show ens18 | grep "inet" | grep -v "inet6" | awk '{ print $2; }'
10.0.2.45/24

[root@centos8 tmp]# ip addr show ens18 | grep "inet" | grep -v "inet6" | awk '{ print $2; }' | sed 's/\.*$//'
10.0.2.45
```

---

Copyright © 2023 Hugh Norris.

From:

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



Permanent link:

<https://www.ittraining.team/doku.php?id=elearning:workbooks:centos:8:lcf800:l701>

Last update: **2023/07/14 10:43**