



Administration Linux niveau 1

LPIC-1

Trainer:Younes Kamal Elidrissi



About me

Younes Kamal Elidrissi : Trainer Consultant

kamal_elidrissi@hotmail.com

kamal_elidrissi@befinfo.com

RHCE-CSM: Redhat Certificate of Expertise in Clustering and Storage Management

RHCSE: Redhat Certified System Engineer

RHCSA : Redhat Certified System Administrator

VITAI : VMware IT Academy Instructor

VCP : VMware Certified Professional

VCP-Cloud : VMware Certified Professional Cloud

SCSA: *Sun certified system Administrator solaris10*

SCNA: *Sun certified Network Administrator solaris10*

LPI: *linux Professional institut administrateur niveau 1*

NCLA: *Novel Certified Linux Administrator Suse Enterprise 11*



Linux LPI niveau 1 (lpi-101)

- Module 00 : Historique
- Module 01 : Introduction au système linux
- Module 02 : Installation De Linux (Redhat, Centos,...)
- Module 03 : Commandes de base partie1
- Module 04 : Commandes de base partie2
- Module 05 : Les scripts Shell
- Module 06 : Les utilisateurs et les droits
- Module 07 : Gestion du système de fichier
- Module 08 : Gestion et installation des packages

Linux LPI niveau 1 (lpi-102)

- Module 09 : Noyau linux
- Module 10 : Démarrage, initialisation, arrêt système
- Module 11 : Taches Administratives (crontab, cpio,..)
- Module 12 : L'impression
- Module 13: Configuration Réseaux (tcp/ip,.....)
- Module 14: Services Réseaux (web, Samba, NFS,...)
- Module 15 : Les utilitaires du client linux (ftp,ssh..)
- Module 16 : Élément de sécurité (TCPwrappers, accounts,..)
- Module 17 : L'environnement Graphique (X11, GDM..)



LPI-101

LPIC-1

Trainer: Younes Kamal Elidrissi



Historique suite

- 1969 : Ken Thompson et Dennis Ritchie écrivent une première version du noyau d'un système d'exploitation pour les laboratoires BELL.
- Ils inventent en 1973 un langage d programmation nommé le langage C. Ils ré-écrivent le noyau d'UNIX en C, L'avantage est celui de la portabilité de l'OS.
- L'entreprise ATT qui fournit les premières versions commerciales de cet OS. Diverses entreprises s'intéressent à ce marché, et plusieurs versions apparaissent (Sun et son SunOS, qui deviendra Solaris, IBM et son AIX, HP et son HPUNIX...).
- L'université californienne de Berkeley travaille également sur Unix, et lui apporte des atouts en réseau .
- Les Unix d'aujourd'hui sont les héritiers des versions BSD (berkeley) ou ATT (System V), soit un peu des deux. Sun est de nos jours un acteur majeur d'Unix.

historique

- le système GNU/Linux a vu le jour en 1991 par un étudiant finlandais, Linus Torvalds.
- Linux a su garder, au fil de ses évolutions, l'héritage des tous premiers systèmes Unix.
- Parallèlement à cela, une philosophie nouvelle est apparue concernant le partage des connaissances, protégée par une licence qui garantirait la transparence des fichiers sources et la possibilité de les modifier.

Naissance de linux

1991 : Linus Torvalds (Finlandais) développe un noyau s'inspirant d'unix : linux. Il le met très vite sous licence GPL, rejoint par de nombreux développeurs.



Succès : qualité technique du noyau + nombreuses distributions qui facilitent l'installation du système et des programmes



Module 01:

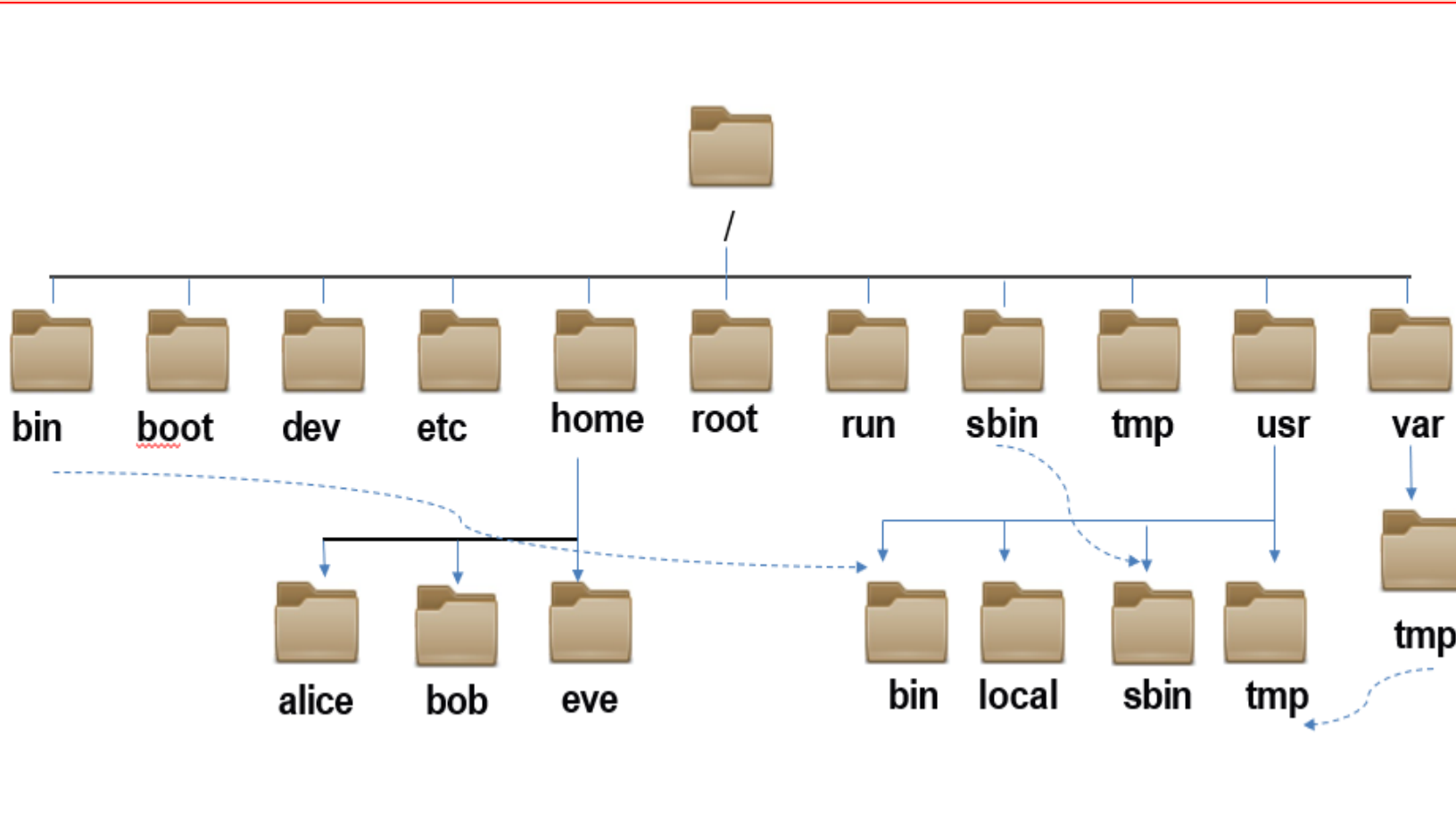
Introduction au système linux

Module 01 : Introduction au système linux

Noyau linux

- Il s'agit de la partie fondamentale d'un système d'exploitation. Le noyau permet de simplifier et sécuriser l'utilisation des différents composants et périphériques de l'ordinateur.
- Il détermine également quel programme doit s'exécuter et pendant combien de temps grâce à une méthode appelée l'ordonnancement.
- Résident en mémoire, se charge au démarrage (boot).
- Le noyau est « minimal », les fonctionnalités plus évoluées sont exécutées en mode « utilisateur »

Le système de fichiers



Le système de fichiers

/	Racine du système, contient les répertoires principaux
/bin	Commandes essentielles communes à tous les utilisateurs
/boot	Fichiers de démarrage du système, contient le noyau
/dev	Points d'entrée des périphériques
/etc	Fichiers de configuration
/home	Contient les répertoires personnels des différents utilisateurs
/root	Répertoire personnel de l'administrateur
/usr	Hiérarchie secondaire, applications, bibliothèques partagées
/proc	Système de fichier virtuel, informations en temps réel

Le Shell : intérêt ?

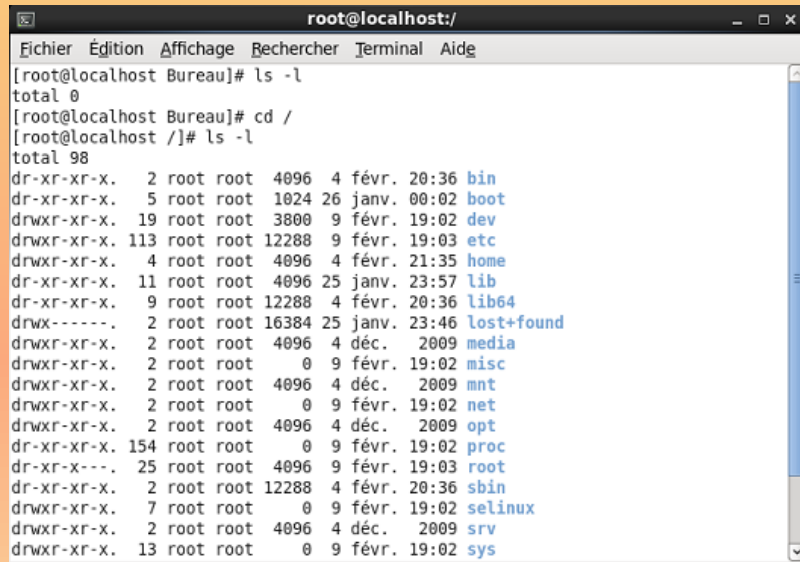
- Interface utilisateur non graphique
 - terminaux texte
 - accès distant
 - Interpréteur de scripts
 - traitement "par lots"
 - automatisation
 - Lancement de tâches multiples
 - tâche combinées (pipes)
 - job control
 - Le choix est vaste
 - sh, bash, ksh, csh, tcsh, ash, zsh, fish,
- On peut exécuter plusieurs shells en parallèle

Le Shell

Nom	Description
bash	(Bourne Again Shell) offre l'édition de la ligne de commande et le rappel des commandes précédentes
csh	(C Shell) développé à Berkeley, compatible avec le shell Bourne. Pas d'édition de la ligne de commande ni d'historique des commandes
ksh	(Korn Shell) offre l'édition de la ligne de commande (touches compatibles Emacs)
sh	le shell original, pas d'édition de la ligne de commande.
tcsh	version améliorée du csh, avec un support de l'édition de la ligne de commande avec correction des commandes tapées
zsh	shell similaire au Korn shell, avec plus de dynamisme lors des affichages et gère la non redondance des commandes.

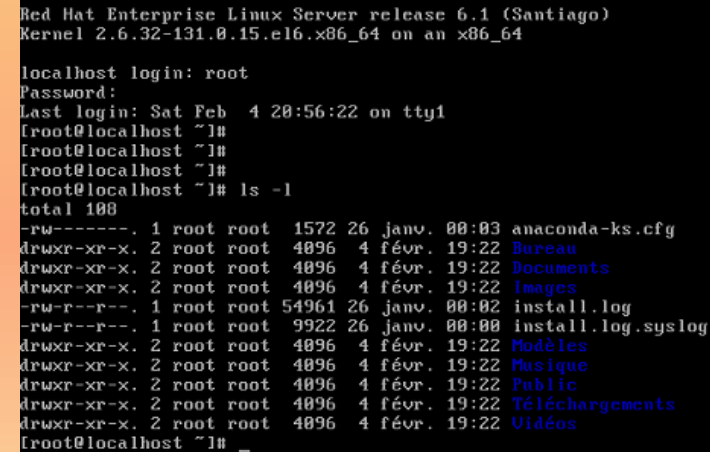
Terminaux

Le terminal est un programme qui ouvre une console dans une interface graphique, il permet de lancer des **commandes**.



```
root@localhost:/  
Fichier Édition Affichage Rechercher Terminal Aide  
[root@localhost Bureau]# ls -l  
total 0  
[root@localhost Bureau]# cd /  
[root@localhost /]# ls -l  
total 98  
dr-xr-xr-x.  2 root root  4096  4 févr. 20:36 bin  
dr-xr-xr-x.  5 root root 1024 26 janv. 00:02 boot  
drwxr-xr-x. 19 root root 3800  9 févr. 19:02 dev  
drwxr-xr-x. 113 root root 12288  9 févr. 19:03 etc  
drwxr-xr-x.  4 root root  4096  4 févr. 21:35 home  
dr-xr-xr-x. 11 root root  4096 25 janv. 23:57 lib  
dr-xr-xr-x.  9 root root 12288  4 févr. 20:36 lib64  
drwx-----.  2 root root 16384 25 janv. 23:46 lost+found  
drwxr-xr-x.  2 root root  4096  4 déc.  2009 media  
drwxr-xr-x.  2 root root    0  9 févr. 19:02 misc  
drwxr-xr-x.  2 root root  4096  4 déc.  2009 mnt  
drwxr-xr-x.  2 root root    0  9 févr. 19:02 net  
drwxr-xr-x.  2 root root  4096  4 déc.  2009 opt  
dr-xr-xr-x. 154 root root    0  9 févr. 19:02 proc  
dr-xr-x---. 25 root root  4096  9 févr. 19:03 root  
dr-xr-xr-x.  2 root root 12288  4 févr. 20:36 sbin  
drwxr-xr-x.  7 root root    0  9 févr. 19:02 selinux  
drwxr-xr-x.  2 root root  4096  4 déc.  2009 srv  
drwxr-xr-x. 13 root root    0  9 févr. 19:02 sys
```

Gnome-terminal



```
Red Hat Enterprise Linux Server release 6.1 (Santiago)  
Kernel 2.6.32-131.0.15.el6.x86_64 on an x86_64  
  
localhost login: root  
Password:  
Last login: Sat Feb  4 20:56:22 on tty1  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]# ls -l  
total 108  
-rw-----.  1 root root 1572 26 janv. 00:03 anaconda-ks.cfg  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Bureau  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Documents  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Images  
-rw-r--r--.  1 root root 54961 26 janv. 00:02 install.log  
-rw-r--r--.  1 root root  9922 26 janv. 00:00 install.log.syslog  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Modèles  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Musique  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Public  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Téléchargements  
drwxr-xr-x.  2 root root  4096  4 févr. 19:22 Vidéos  
[root@localhost ~]# _
```

console

Le Login user et root

/bin/bash est le shell par défaut sur redhat il est lancé quand vous vous loggez depuis un terminal virtuel (**Ctrl-Alt-F1 à Ctrl-Alt-F6**).
ou lorsque vous lancez une session **gnome-terminal** :

- Le compte utilisateur simple *classique*
- Le compte *root* (Administrateur)

Un symbole de l'invite de commande vous permet de déterminer le type de compte que vous utilisez.

Accès simple utilisateur :

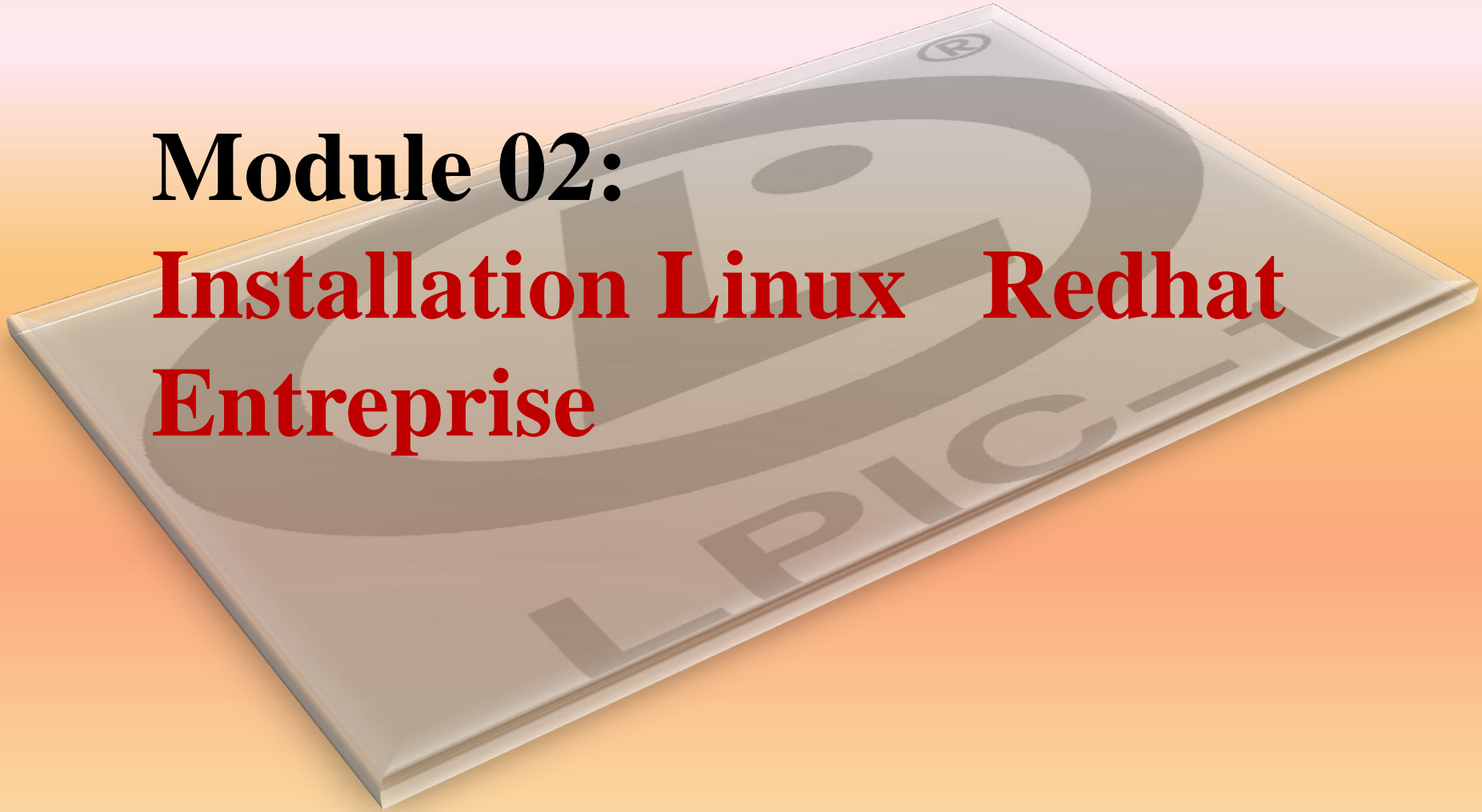
- **user@localhost \$**

Accès root :

- **root@localhost #**

Module 02:

Installation Linux Redhat Enterprise



Votre matériel est-il compatible ?

- La compatibilité matérielle constitue un point essentiel si vous possédez un système plus ancien ou élaboré par vos soins.
- **RedHat Enterprise Linux** est théoriquement compatible avec la plupart des configurations matérielles assemblées en usine au cours des deux dernières années.
- La liste la plus récente de compatibilité du matériel est fournie à l'adresse :

<http://hardware.redhat.com/hcl/>

Avez-vous suffisamment d'espace disque ?

-À moins que vous n'ayez une raison particulière pour agir différemment, nous vous conseillons de créer les partitions suivantes pour les systèmes **x86, AMD64 et Intel® 64** :

- Une partition swap (d'au moins 256 Mo)
- Une partition **/boot/** (100 Mo)
- Une partition **root (3.0 Go - 5.0 Go)** — où se trouve "/" (le répertoire racine).

Une **partition de 3.0 Go** vous permet d'effectuer une installation minimale, alors qu'une **partition root de 5.0 Go** vous permet d'effectuer une installation complète.

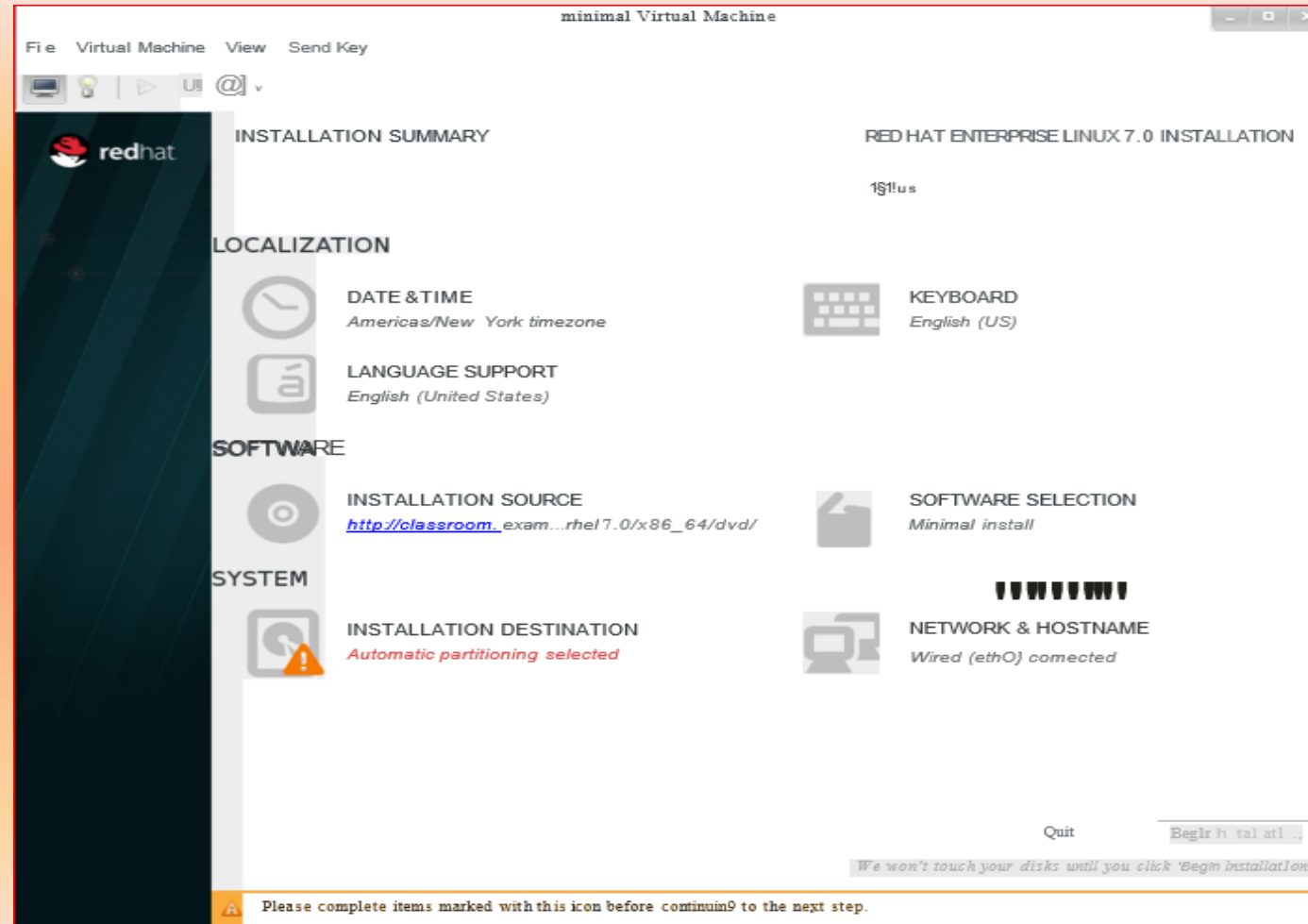
Installation De Base

- Une installation de base ne nécessite que deux partitions, la partition racine / et une partition de **swap** (mémoire virtuelle).
- - **Le kernel Linux 2.6** est notamment plus performant lorsqu'un swap est disponible:
- - il peut ainsi y déplacer des données non fréquemment utilisées et donc gagner en performance disque (puisque les caches disques sont allouées de la mémoire libre).

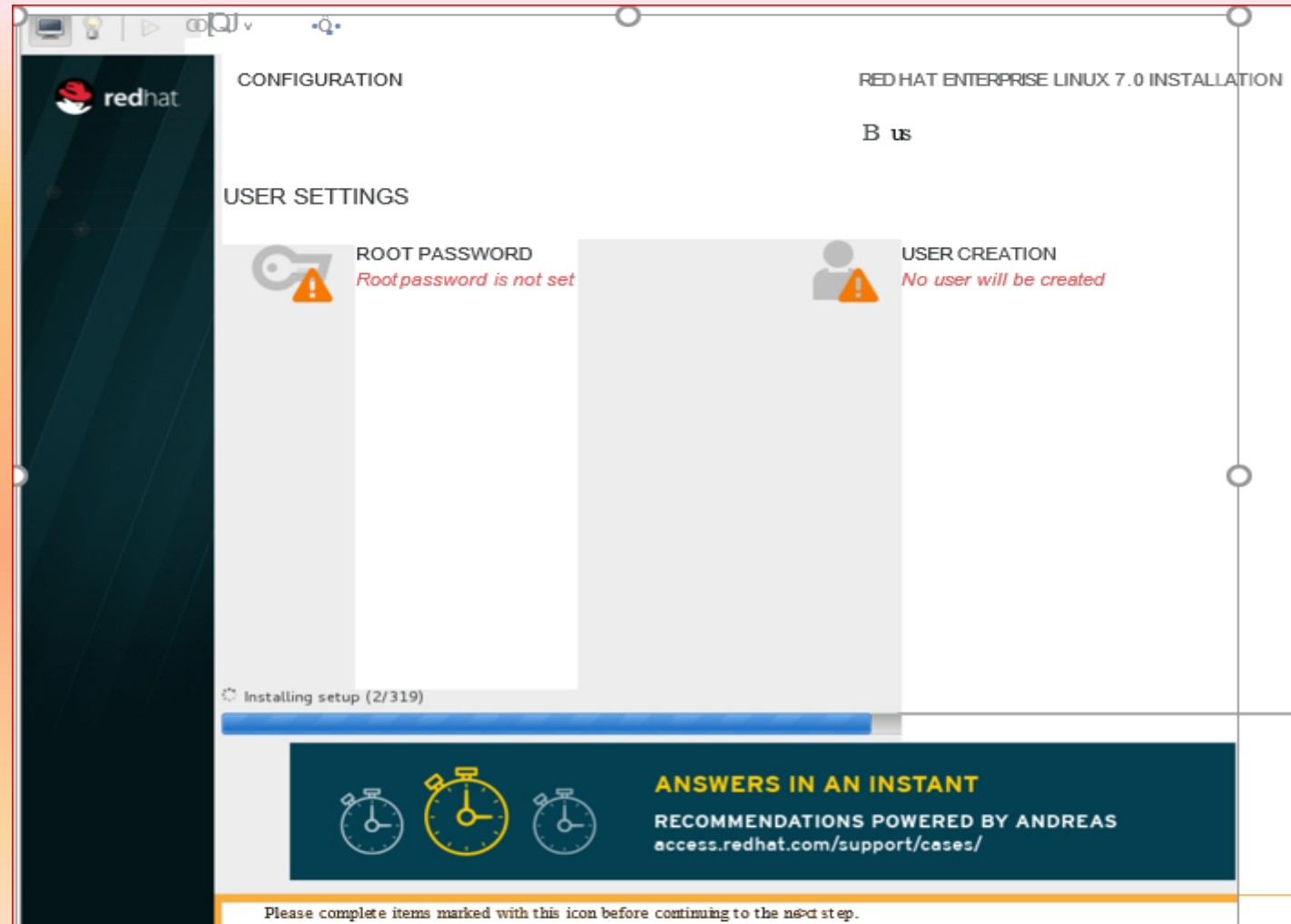
Installation Personnalisée

- On peut également subdiviser le système en plusieurs partitions si désiré. Les raisons d'un tel partitionnement plus fin sont :
- **Maintenir le démarrage** sous la limite de 1024 cylindres (disques IDE et anciens BIOS) :
- Petit **/** séparé de **/usr**, ou carrément partition de démarrage **/boot**.
- **Performance** : les opérations I/O de lecture et écriture sur disque
- **Utilisation de plusieurs FS** : systèmes de fichiers différents (p.ex. : ext4, xfs....)

Installation De Base



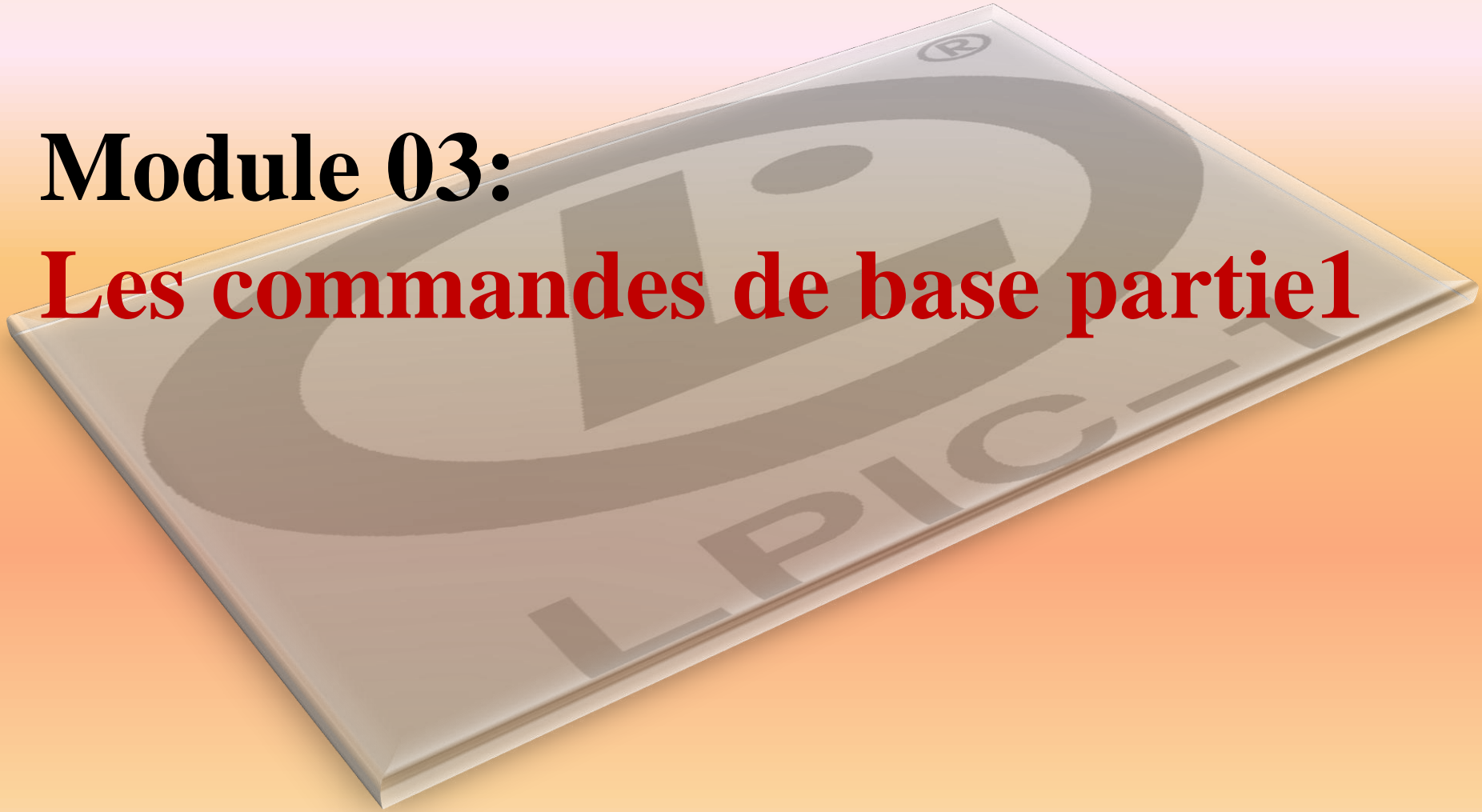
Installation De Base



LAB

Module 03:

Les commandes de base partie1



La Ligne De commande

- Toutes les lignes de commande ont cette syntaxe:

command **[options]** **[arguments]**

Les lignes de commande peuvent être assemblées dans un fichier pour former un script.

- **Par exemple :**

-Pour afficher les information du système

uname -a Afficher toutes les informations décrites ci-dessus

Afficher le type (matériel) de machine.

Afficher le nom d'hôte de la machine sur le réseau.

Afficher le numéro de version du système d'exploitation

Les pages de manuel (Man pages)

- Les manuels en ligne décrivent la plupart des commandes utilisées dans votre système. **man [command]**

Exemples:

```
[root@test ] /# man mkdir
[root@test ] /# man cal
```

- Pour rechercher un mot clé dans les pages de manuel, il faut utiliser l'option k

```
[root@test ] /# man -k compress
[root@test ] /# apropos compress
```

- L'emplacement des pages de manuel peut être modifié avec la variable MANPATH. Pour afficher le contenu de MANPATH.

```
[root@test ] /# echo $MANPATH
/usr/local/man:/usr/share/man:/usr/X11R6/man
```

Aide : info, help, apropos, whatis

- **info commande**

info est constitué d'un ensemble de pages réparties en plusieurs niveaux, à travers lesquelles il est possible de naviguer de diverses façons.

- **help Commande**

affiche le manuel d'une commande interne (builtin)

- **apropos sujet**

affiche les pages de man correspondant au sujet

- **whatis Commande**

affiche une information succincte sur la commande

Balade dans le «fs»

- Se déplacer dans le filesystem

cd chemin (chemin relatif) cd /chemin (chemin absolu)

- Si vous êtes en administrateur système la commande par **cd ~** vous placera dans le répertoire **/root**.
- Dans le cas où je suis (je suis connecté en tant qu'utilisateur *user1*) je vais automatiquement me retrouver dans le répertoire de l'utilisateur *user1* qui se trouve dans **/home/user1**.
- Les répertoires des utilisateurs sont tous sous **/home**.

Balade dans le «fs»

- pour connaître le chemin du répertoire où l'on se trouve est d'utiliser la commande **pwd** (print working directory) :

pwd

- Pour changer de répertoire courant :

cd *répertoire*

- Le symbole (**~**) représente votre répertoire d'accueil
- Le symbole (**..**) représente le répertoire parent.
- Le symbole (**.**) représente le répertoire courant.

commandes de base : ls

ls option argument

-a : affiche tous les fichiers (y compris ceux commençant par '.')

-l : listing étendu

-R : récursif

-S : tri par taille

-t : tri par date de modification

-1 : affichage sur une colonne

– exemples

ls *.txt

ls /etc

ls /etc/host*

ls /etc/rc[13].d

ls laR ~

commandes de base : {mk,rm}dir

mkdir [p] répertoire1 répertoire2 ...

crée les répertoires répertoire1, répertoire2, ...

l'option -p per et destination (fichier ou répertoire)

– options

-p : crée les répertoires supérieurs si nécessaire

– exemples

mkdir \$HOME/documents

mkdir test \$HOME/images_iso

mkdir p \$HOME/documents/personnel/{photos,factures}

commandes de base : {mk,rm}dir

rmdir répertoire1 répertoire2

supprime les répertoires répertoire1, répertoire2, ...

ces répertoires doivent être vides pour pouvoir être supprimés
(utiliser **rm -rf** sinon)

– exemples

rmdir \$HOME/documents

rmdir test \$HOME/images_iso

rmdir \$HOME/documents/personnel/{photos,factures,}

commandes de base : cp & mv

cp source destination

copie la source (fichier ou répertoire) vers la destination (fichier ou répertoire)

– options

-p : préserve les droits

-R : récursif

-f : force l'écrasement de la destination

– exemples

cp *.txt /tmp

cp test.txt toast.txt

cp -Rf /home /var/backup

commandes de base : cp & mv

mv source destination

- déplace la source (fichier ou répertoire) vers la destination (fichier ou répertoire).
- permet aussi de renommer un fichier

– exemples

```
mv *.txt /tmp
```

```
mv test.txt toast.txt
```

commandes de base : rm

rm **argument**

Supprime le fichier ou répertoire argument

– options

-R : récursif

-f : force la suppression

– exemples

rm -rf /myrep1

rm toast.txt

Fait Attention !!!!!!!!!!! rm -rf /*

commandes de base : touch

touch fichier

Crée le fichier s'il n'existe pas, ou met la date de modification du fichier à l'heure courant s'il existe.

La commande '>' et 'echo' permet aussi de créer un fichier

– exemples

```
touch toast.txt
```

```
>file1.txt
```

```
echo > file2.txt
```

commandes de base : cat, less

cat fichier1 fichier2 ...

affiche le contenu de fichier1 fichier2 ... sur la sortie standard

si cat est appelé sans arguments, la source est l'entrée standard

– exemple

cat /var/log/messages

less fichier1 fichier2 ...

comme cat, affiche le contenu de fichier1 fichier2 ... sur la sortie standard mais effectue un arrêt à chaque page si less est appelé sans arguments, la source est l'entrée standard

– exemple

less /etc/passwd

commandes de base : more, tee, wc

more fichier

a l'avantage d'afficher le fichier page par page. Pour passer d'une page à l'autre, tapez sur la touche **ESPACE**

– exemple : `more /var/log/messages`

tee fichier

duplique l'entrée standard vers la sortie standard et dans un fichier

– exemple : `vmstat 1 | tee toto`

wc option fichier

compte le nombre de lignes (-l), bytes (-c), mots (-w) dans fichier (ou sur STDIN si aucun fichier n'est spécifié)

– exemple: `wc -l /etc/passwd`

commandes de base : head & tail

head [nX] fichier1

affiche les X premières lignes de fichier1 sur la sortie standard

– exemple `head -n1 /etc/passwd`

tail [nX] fichier1

affiche les X dernières lignes de fichier1 sur la sortie standard

si tail est appelé sans arguments, la source est l'entrée standard et le nombre de lignes est 10

– exemple

`tail -n5 /var/log/syslog`

`tail -f /var/log/syslog`

commandes de base : head & tail

head [nX] fichier1

affiche les X premières lignes de fichier1 sur la sortie standard

– exemple `head -n1 /etc/passwd`

tail [nX] fichier1

affiche les X dernières lignes de fichier1 sur la sortie standard

si tail est appelé sans arguments, la source est l'entrée standard et le nombre de lignes est 10

– exemple

`tail -n5 /var/log/syslog`

`tail -f /var/log/syslog`

commandes de base : head & tail

Une combinaison des deux permet d'afficher la nième ligne d'un fichier :

head -n10 /etc/passwd | tail -n1

affiche la 9ème (10-1) ligne de /etc/passwd

head -n20 /etc/group | tail -n3

affiche les lignes 17 à 20 (20-3 -> 20) lignes de /etc/group

commandes de base : cut

- La commande cut permet d'afficher des zones spécifiques d'un fichier.

- Par exemple :

cut -c1 /etc/passwd

affichera la première colonne du fichier /etc/passwd. Il existe d'autres spécifications

- On peut également spécifier un *séparateur de champs* avec l'option *-d*. Par exemple :

cut -d: -f6 /etc/passwd

- affichera le 6^{eme} champ du fichier /etc/passwd, dont le séparateur de champs est le caractère double point (":").

commandes de base : sort

- La commande **sort** trie les lignes d'un ou plusieurs fichiers de texte. Par défaut, le tri se fait suivant l'ordre lexicographique.
- Un certain nombre d'options sont fournies pour modifier l'ordre du tri :
 - **-n** pour tri numérique,
 - **-r** pour tri inversé,
 - **-k x** pour tri à partir du champ x,
 - **-t c** pour utiliser le séparateur de champs
- Quelques exemples de la commande sort :

```
ls -l | sort -n -k5
```

```
sort -n -k3 /etc/passwd
```

commandes de base : uniq

Ce filtre élimine les lignes dupliquées depuis un fichier trié.
On le voit souvent dans un tube combiné avec un **sort**

```
cat liste-1 liste-2 liste-3 | sort | uniq > liste.finale
```

Dans cet exemple

la 1^{er} commande cat Concatène les fichiers liste1 list2 et list3

la 2em commande sort les trie,

la 3em commande uniq efface les lignes doubles,

et enfin > écrit le résultat dans un fichier de sortie.

la commande sed

sed est un éditeur de ligne non interactif, il lit les lignes d'un fichier une à une leur applique un certain nombre de commandes d'édition et renvoie les lignes résultantes sur la sortie standard.

- Il ne modifie pas le fichier traité, il écrit tout sur la sortie standard.

- la syntaxe n'est franchement pas très conviviale, mais il permet de réaliser des commandes complexes sur des gros fichiers.

La syntaxe de sed est la suivante:

sed -e 'programme sed' fichier-a-traiter

Ou

Sed -f fichier-programme fichier-a-traiter

la commande sed

La fonction de substitution **s** permet de changer la première ou toutes les occurrences d'une chaîne par une autre. La syntaxe est la suivante:

sed "s/toto/TOTO/" fichier

va changer la première occurrence de la chaîne toto par TOTO (la première chaîne toto rencontrée dans le texte uniquement)

sed "s/toto/TOTO/3" fichier

va changer la troisième occurrence de la chaîne toto par TOTO

Dans même ligne ex:totototo.....**toto**.....

(la troisième chaîne toto rencontrée dans le texte uniquement)

la commande split

La commande split permet de découper un fichier en plusieurs plus petits unités. Ses options sont :

- **b** *n* (Bytes) découpage par blocs de *n* octets
- ou
- **l** *n* (Lignes) découpage par blocs de *n* lignes

Syntaxe : `split fichier`

Exemple :

split -b 135000 vacances.mpeg

commandes de base : grep & egrep

grep options patron fichier

liste les fichiers contenant une chaîne correspondant à un patron.

– options

-v : inverse le comportement de grep (n'affiche que les lignes qui ne correspondent pas)

-i : insensible à la casse

-R : récursif

– patrons d'expression régulières

. n'importe quel caractère

* le caractère précédent 0 ou plusieurs fois

+ le caractère précédent 1 fois au moins

[az] un caractère en minuscule

[azAZ] une lettre

[09] un chiffre

^/\$ le début/la fin de ligne

| séparateur pour spécifier de multiples expressions (ou logique)

commandes de base : grep & egrep

- **Ex1:-** pour chercher tous les lignes qui contiennent le mot root

Dans le fichier /etc/group

grep root /etc/group

- **Ex2:** pour chercher tous les lignes qui ne contiennent pas le mot root dans le fichier /etc/group

grep -v root /etc/group

- **Ex3:-** pour chercher les noms des fichiers qui contient le mot root

grep -l root group passwd hosts

- **Ex4:** pour compter le nombre de lignes qui contiennent le mot root dans le fichier /etc/group

grep -c root /etc/group

commandes de base : grep & egrep

- Matcher une ligne commençant par « foo » :

```
egrep "^foo.*"
```

- Matcher une ligne commençant par « foo » ou commençant par « bar » :

```
egrep "^(foo|bar) "
```

- Matcher une ligne commençant par « foo » et se terminant par « bar » :

```
egrep "^foo.*bar$"
```

commandes de base : find

- La commande find permet de rechercher des fichiers au sein de l'arborescence du système de fichiers à l'aide de critères et donne la possibilité d'agir sur les résultats retournés.
- **find chemin critères options**
- La commande find étant réursive, il suffit d'indiquer un répertoire de base pour que toute l'arborescence depuis ce répertoire soit développée.

L'option de base est **-print** (souvent implicite sur la plupart des Unix) qui permet d'afficher sur écran les résultats.

find

commandes de base : find

- Critères de recherche :

-name : `find . -name "fic*" -print`

-type : permet une sélection par type de fichier `find / -type d`

-size : permet de préciser la taille des fichiers recherchés

`find -size +100k`

-mtime : recherche sur la date de dernière modification

`find . -mtime -1`

-perm : permet d'effectuer des recherches sur les autorisations d'accès ex: `find -type d -perm -111`

-inum : -inum permet une recherche par numéro d'inode et

`find / -inum 13456`

commandes de base : find

- **Commande**
- Outre l'option **-print** on trouve d'autres options permettant d'effectuer une action sur les fichiers trouvés.
- ls**: affiche des informations détaillées sur les fichiers trouvés
- exec** : La commande exécutée par **-exec** doit se terminer par un caractère spécial doit s'écrire **** pour ne pas être interprété par le shell.

Pour passer comme paramètre pour la commande le fichier trouvé par find, il faut écrire **{ }** (substitution du fichier).

Exemple pour effacer tous les fichiers finissant par « .mp3 » :

```
find . -type f -name "*.mp3" -exec rm -f { } \;
```

commandes de base : **whereis**, **which**, **locate**

- La commande **whereis** recherche dans les chemins de fichiers binaires, du manuel et des sources les fichiers correspondant aux critères fournis.

whereis date

- La commande **which** recherche une commande dans le PATH (chemin des exécutables) et vous fournit la première qu'elle trouve

which date

- La commande **locate** recherche un fichier selon le modèle donné dans une base de données de fichiers construite par la commande **updatedb**.

updatedb

locate toto

commandes de base : ln

Les liens sont des fichiers spéciaux permettant d'associer plusieurs noms (liens) à un seul et même fichier

On distingue deux types de liens :

- **Les liens symboliques** représentant des pointeurs virtuels (raccourcis) vers des fichiers réels. En cas de suppression du lien symbolique le fichier pointé n'est pas supprimé.
- **Les liens physiques** (aussi appelées liens durs ou en anglais **hardlinks**) représentent un nom alternatif pour un fichier

Commandes De Base : ln

Les liens sont des fichiers spéciaux permettant d'associer plusieurs noms (liens) à un seul et même fichier

On distingue deux types de liens :

- **Les liens symboliques** représentant des pointeurs virtuels (raccourcis) vers des fichiers réels. En cas de suppression du lien symbolique le fichier pointé n'est pas supprimé.
- **Les liens physiques** (aussi appelées liens durs ou en anglais **(hardlinks)**) représentent un nom alternatif pour un fichier
 - - Pour créer un lien symbolique :
 - **# ln -s source_file target_file**
 - Les liens physiques sont créées à l'aide de la commande ln (sans l'option -n) selon la syntaxe suivante :
 - **# ln nom-du-fichier-reel nom-du-lien-physique**

LAB