

Chapitre 03 : Fichiers et Commandes de base

Dr Mandicou BA

mandicou.ba@esp.sn

<http://www.mandicouba.net>

Diplôme Universitaire de Technologie (DUT, 1^e année)
Diplôme Supérieure de Technologie (DST, 1^e année)
Informatique



ÉCOLE SUPÉRIEURE POLYTECHNIQUE

www.esp.sn



Plan du Chapitre

- 1 Objectifs et Contenu
- 2 Notion de Shell
- 3 Fichiers et Système de Fichiers
 - Notion de fichier et Répertoire
 - L'arborescence des fichiers
- 4 Commandes de base
 - Commandes de manipulation de fichiers et répertoires
 - Les attributs des fichiers
 - Redirections et Méta-Caractères
 - Commande find : Recherche fichiers/répertoires
 - Commandes utilitaires
- 5 Les éditeurs de texte
 - Les éditeurs vi, vim, nano, etc.
- 6 Exercices d'application

Sommaire

- 1 Objectifs et Contenu
- 2 Notion de Shell
- 3 Fichiers et Système de Fichiers
- 4 Commandes de base
- 5 Les éditeurs de texte
- 6 Exercices d'application

Objectifs

- ☞ Connaître les principales caractéristiques du shell
- ☞ Connaître le système de fichier et les principaux fichiers
- ☞ Connaître les principales commandes de base de Linux
- ☞ Éditer un fichier avec les éditeurs vi, vim, nano

Contenu

- ☞ Le shell
- ☞ L'arborescence des fichiers
- ☞ Les attributs de fichiers
- ☞ Les commandes de base
- ☞ Les éditeurs vi, vim et nano

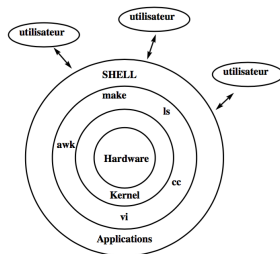
Sommaire

- 1 Objectifs et Contenu
- 2 Notion de Shell**
- 3 Fichiers et Système de Fichiers
- 4 Commandes de base
- 5 Les éditeurs de texte
- 6 Exercices d'application

La théorie

- ☛ Le **shell**, ou **interpréteur de commande** :
 - est le logiciel que vous utilisez à tout instant
- ☛ Il permet d'activer des commandes :
 - soit de manière interactive,
 - soit de manière programmée (sous forme de **scripts** cf. cours second semestre).
- ☛ Sous Linux, le shell usuel est le **bash** (**B**ourne-**A**gain **s**hell).
 - Il est compatible avec le shell ISO (**sh**).
- ☛ Quelques autres shells :
 - ksh, dash, csh, zsh, etc.
- ☛ **Principale difficulté du shell** : usage de multitude de caractères spéciaux (**à venir**)

Positionnement du Shell



Le shell est un programme séparé du noyau qui réalise plusieurs tâches :

- ☛ interprète les commandes tapées au clavier par l'utilisateur et exécute les programmes nécessaires
- ☛ fournit à l'utilisateur un langage de programmation interprété
- ☛ permet de modifier l'environnement de travail
- ☛ réalise les redirections des entrées-sorties

Les tâches fond dans le shell

- ☞ Le caractère spécial **&** permet de lancer une commande (ou un programme) en tâche de fond (**background**)
 - Dès que la commande est lancée : la main est récupérée dans la fenêtre de lancement pour faire autre chose
 - Réduction du nombre de fenêtres actives en même temps et donc la charge du micro-processeur et l'occupation de la mémoire
- ☞ Prudence dans l'utilisation du **&** :
 - 1 une application prévue pour tourner en tâche de fond est généralement multi-fenêtrée et comprend un menu contenant une case *Quit* ou *Exit*.
 - 2 S'il s'agit d'un programme que vous avez écrit, il faut être certain qu'il ne boucle pas car alors vous n'avez plus d'autre possibilité que d'utiliser **ps** puis **kill** pour l'arrêter

Sommaire

- 1 Objectifs et Contenu
- 2 Notion de Shell
- 3 Fichiers et Système de Fichiers**
 - Notion de fichier et Répertoire
 - L'arborescence des fichiers
- 4 Commandes de base
- 5 Les éditeurs de texte
- 6 Exercices d'application

Notion de fichier

Qu'est ce qu'un fichier ?

- ☛ **Fichier** : Une structure (objet) logique qui délimite une zone de stockage de données sur disque
- ☛ Il est caractérisé par un **type** qui dépend :
 - ① de la nature des données qu'il contient
 - ② et par des **attributs** :
 - les droits, date de création, date de modification, etc.
- ☛ **Concrètement** : un fichier est une suite de données (d'octets) sauvegardée sur un support de stockage **permanent** :
 - disque dur, mémoire flash, CD, DVD, etc.
- ☛ Le codage utilisé détermine la nature du fichier :
 - texte, image, son, vidéo, programme exécutable, etc.

Notion de fichier

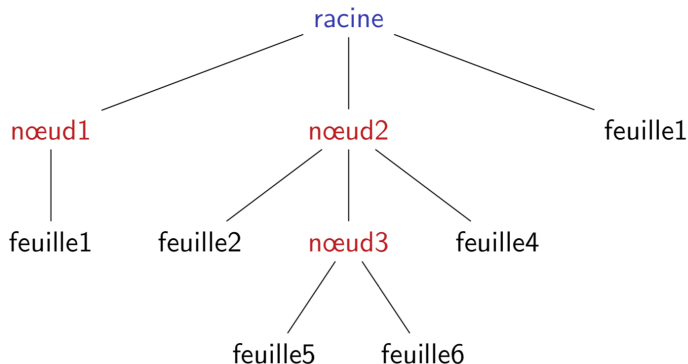
Convention de nommage

- ☛ Nom d'un fichier : chaîne de caractères
 - En pratique : usage des caractères alphabétiques, aux chiffres, ainsi qu'aux caractères `_` , `-`
- ☛ Pratique courante: **nom.extension**
 - L'extension permet à l'utilisateur de se souvenir du type du fichier
 - Exemple d'extension : txt, pdf, gif, png, jpeg, avi, mpeg, etc.
- ☛ Linux fait la différence entre les caractères majuscules et les minuscules
 - Dsti1.txt \neq dsti1.txt

Notion de répertoire

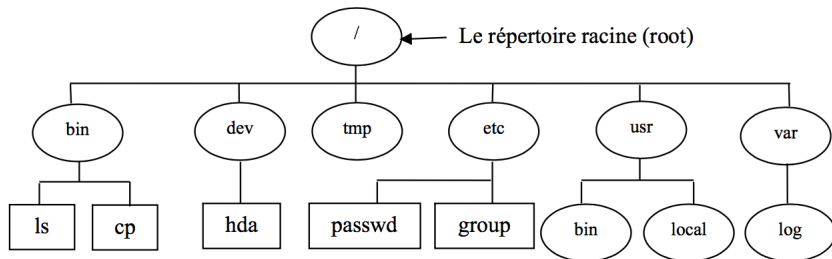
- ☛ Les **répertoires** (ou **dossiers**) sont des fichiers particuliers qui peuvent contenir d'autres fichiers
- ☛ Les règles de nommage vues pour les fichiers restent valides
- ☛ Dans un même répertoire deux fichiers différents ne peuvent pas avoir le même nom
- ☛ Un répertoire peut contenir des fichiers et des sous-répertoires :
 - **Système de fichiers sous forme d'arbre**

Notion de Structure arborescente du système de fichiers



- ☛ **racine** : répertoire racine
- ☛ **nœuds** : répertoires
- ☛ **feuilles** : fichiers

L'arborescence de fichiers Linux



Les principaux répertoires

<code>/bin/</code>	Le répertoire contenant les principales commandes (ls, cp...)
<code>/boot/</code>	Le répertoire contenant les fichiers de démarrage (vmlinuz...)
<code>/dev/</code>	Le répertoire contenant les fichiers périphériques (console, hda...)
<code>/etc/</code>	Le répertoire contenant la configuration du système (passwd...)
<code>/home/</code>	Le répertoire contenant les répertoires de connexion des utilisateurs
<code>/lib/</code>	Le répertoire contenant les bibliothèques
<code>/mnt/</code>	Répertoire permettant des montages temporaires de FS
<code>/opt/</code>	Répertoire contenant des applications (alternative à /usr)
<code>/proc/</code>	Répertoire d'interface avec le noyau

Les principaux répertoires

<code>/root/</code>	Répertoire de l'administrateur
<code>/sbin/</code>	Répertoire contenant les commandes d'administration
<code>/tmp/</code>	Répertoire contenant les fichiers temporaires
<code>/usr/</code>	Répertoire contenant les applications, les bibliothèques des utilisateurs
<code>/usr/bin/</code>	Répertoire contenant d'autres commandes
<code>/usr/local/</code>	Répertoire contenant les logiciels libres en Open Source
<code>/var/</code>	Répertoire contenant des fichiers résultant de l'exploitation
<code>/var/log/</code>	Répertoire contenant les journaux de bord

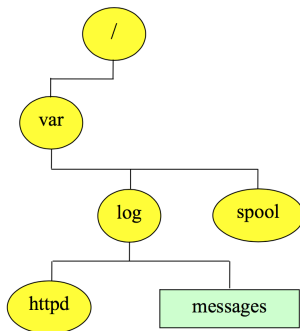
Chemins

- ☛ Un chemin indique, pour une application, l'emplacement d'un fichier dans l'arborescence
- ☛ **Chemin absolu** : part de la racine et commence par « / » :
 - **/var/log/messages** : le fichier messages est dans le répertoire log, qui lui-même est dans le répertoire var, qui lui-même est directement sous la racine.
- ☛ **Chemin relatif** : indique l'emplacement d'un fichier à partir du répertoire courant :
 - **messages** : Le fichier /var/log/messages **si le répertoire courant est** /var/log/. Même chose pour **./messages**
 - **../messages** : Le fichier /var/log/messages **si l'on est dans le répertoire** /var/log/httpd/.

Chemins

A retenir

- ☛ Dans ces chemins, « . » et « .. » représentent respectivement :
 - 1 le répertoire courant
 - 2 et le répertoire père (le répertoire hiérarchiquement supérieur)



Sommaire

1 Objectifs et Contenu

2 Notion de Shell

3 Fichiers et Système de Fichiers

4 **Commandes de base**

- Commandes de manipulation de fichiers et répertoires
- Les attributs des fichiers
- Redirections et Méta-Caractères
- Commande find : Recherche fichiers/répertoires
- Commandes utilitaires

5 Les éditeurs de texte

6 Exercices d'application

Commandes de navigation

- ☞ **pwd** : (Print Working Directory) affiche le nom du répertoire courant
- ☞ **cd** : (Change Directory) : la navigation d'un répertoire à un autre s'effectue avec la commande **cd** suivie du nom du répertoire
 - **cd /** : permet de se retrouver à la racine du disque.
 - **cd ~** ou **cd** : permet d'accéder directement au répertoire de l'utilisateur.
 - **cd /var/log/** : aller dans le répertoire /var/log
 - **cd ..** : remonter dans le répertoire parent à partir de l'emplacement actuel
 - **cd -** : Permet de revenir au répertoire précédent

Commandes d'affichage

☛ **ls** (list) : permet d'afficher le contenu d'un répertoire

☛ Quelques exemples d'utilisation :

- **ls -l** : afficher les informations de manière détaillée
- **ls -a** : afficher les fichiers cachés
- **ls -h** : afficher la taille des fichiers de façon lisible
- **ls -t** trier les fichiers par date du plus récent au plus ancien
- **ls -r** : tri inversé
- **ls -S** trier par taille décroissante
- **ls -la** afficher tous les fichiers y compris les fichiers cachés
- **ls -lhS** : afficher les informations des fichiers, avec des tailles lisibles le tout ordonné du plus grand au plus petit.

Création et copie de fichiers

Création de fichier

☞ Commande `touch`

- `touch monfichier`
- si le fichier existe alors mise à jour la date de dernière modification
- sinon crée un fichier vide avec le nom *monfichier*

Copie de fichier

☞ Commande `cp` : (copy)

- `cp ancien nouveau` : copier le fichier *ancien* dans *nouveau*
- `cp -r AncienDossier NouveauDossier` : copier d'une manière récursive (option -r) le répertoire *AncienDossier* vers le répertoire *NouveauDossier*

Renommage/Déplacement et Suppression de fichiers

Renommage/Déplacement de fichier

☛ Commande **mv** : (move)

- **mv *ancien nouveau*** : déplacer le fichier *ancien* dans *nouveau*, en le renommant si *nouveau* désigne un fichier

Suppression de fichier

☛ Commande **rm** : (remove)

- **rm *fichier*** : effacer (supprimer) le fichier *fichier*
- **rm -r *repertoire*** : effacer (supprimer) le répertoire *repertoire*, et récursivement (option -r) tout son contenu.

Création et suppression de répertoires

Création de répertoire

- ☞ Commande `mkdir` : (make directory)
 - `mkdir repertoire` : créer le répertoire *repertoire*

Suppression de répertoire

- ☞ Commande `rmdir` : (remove directory)
 - `rmdir repertoire` : effacer le répertoire **vide** *repertoire*

Les principaux attributs d'un fichier

- ☞ Le type du fichier (cf. ci-après)
- ☞ Les droits
- ☞ Le propriétaire
- ☞ Le groupe
- ☞ La taille
- ☞ Les dates de dernière modification, de création et de dernier accès

Les types d'un fichier

- Ordinaire (-)
- Répertoire (d)
- Périphérique (c, b)
- Lien symbolique (l)
- Fichier IPC, tube et socket (p, s)

- Un fichier ordinaire est composé de zéro à plusieurs octets.
- Un fichier lien symbolique permet d'accéder à un autre fichier
 - **ls** montre le nom du fichier lié apparaît derrière une flèche
- Les fichiers IPC (Inter Process Communication) permettent la communication entre applications
 - les fichiers sockets (appelés socket Unix) peuvent remplacer les sockets réseau (socket TCP) lorsque le client et le serveur se trouvent sur le même poste.

Focus : la commande ls (1/2)

```
$ ls -l le_fichier
-rw-r--r-- 1 root bin 153 Jul 23 20:21 le_fichier
```

droits

propriétaire

groupe

taille

Date et heure de dernière modification

chemin

type {
- : ordinaire
d : répertoire

Focus : la commande ls (2/2)

```
[root@linux1 ex]# ls -l
total 8
-rw-r--r--  1 root bin   136 Feb 27 12:06 le_fichier
lrwxrwxrwx  1 root root   11 Feb 27 12:07 passwd -> /etc/passwd
crw-r--r--  1 root root  4, 2 Feb 27 12:08 perif
brw-r--r--  1 root root  4, 2 Feb 27 12:08 perif2
drwxr-xr-x  2 root root 4096 Feb 27 12:06 repertoire
srwxr-xr-x  1 root root    0 Apr 11 10:56 socket
prw-r--r--  1 root root    0 Feb 27 12:08 tube
```

Les Redirections

Les redirections

- ☞ **> fichier** : Redirige la sortie standard (STDOUT) d'une commande dans un fichier
- ☞ **>> fichier** : Idem, mais l'écriture se fait en fin de fichier au lieu d'écraser le fichier
- ☞ **< fichier** : Redirige l'entrée standard (STDIN) d'une commande à partir d'un fichier
- ☞ **cmd1 | cmd2** :

Méta-Caractères

Les jockets (Méta-Caractères)

- ☛ * : Une suite quelconque de caractères quelconques dans un nom de fichier.
- ☛ ? : Un caractère quelconque dans un nom de fichier.
- ☛ [abc] : un caractère quelconque parmi a, b ou c
- ☛ [a – d] : un caractère quelconque dans la plage allant du caractère *a* au caractère *d*
- ☛ {*mot1*, *mot2*, *mot3*} : prend comme valeur chaque élément de l'ensemble, donc d'abord *mot1*, ensuite *mot2* et finalement *mot3*
- ☛ {1..4} : spécifie l'ensemble d'éléments qui vont de 1 à 4

Méta-Caractères : exemples d'utilisation

🖱 `ls -l *.pdf`

- affiche seulement les fichiers ayant l'extension pdf

🖱 `rm td[1-3].pdf`

- efface les fichiers tp1.pdf, tp2.pdf et tp3.pdf

🖱 `touch Rapport-DSTI1-01,02.pdf`

- met à jours la date (ou crée s'ils n'existent pas) les fichiers Rapport-DSTI1-0.pdf et Rapport-DSTI1-0.pdf

🖱 `cp /tmp/?DSTI2018.data .`

- copie dans le répertoire courant (.) tous les fichiers qui sont dans le répertoire **/tmp** dont le nom est composé d'une chaîne de neuf caractères qui se termine par **DSTI2018** et qui ont l'extension **data**

Commande find

Structure et rôle de la commande **find**

🖱 **find** *repertoire critere action*

- **chercher** dans la sous-arborescence de *repertoire* les fichiers satisfaisant *critere* et executer *action* sur chaque fichier retrouvé

Exemple d'utilisation de **find**

🖱 **find** ~-name "*.pdf" -print

- Afficher tous les fichiers ayant l'extension pdf qui se trouvent dans la sous-arborescence enracinée dans le répertoire personnel de l'utilisateur.

La commande find : les critères

- ☞ **-name nom** : rechercher selon le nom de l'objet.
- ☞ **-size [+|-]n[c|k|b]** : Recherche les fichiers dont la taille est de **n**.
 - + : plus grand, - : plus petit, rien : exactement, c : octet, k : kibi octet, b : bloc de 512 o (par défaut)
- ☞ **-mtime [+|-]n** : Recherche les fichiers modifiés depuis les n derniers jours
 - : + : plus grand, - : plus petit, rien : exactement, n = nombre de jours
- ☞ **-atime [+|-]n** : Recherche les fichiers accédés depuis les n derniers jours.
- ☞ **-ctime [+|-]n** : Recherche les fichiers accédés depuis les n derniers jours.
- ☞ **-type [f|d|l]** : rechercher selon le type (f : fichier, d : répertoire et l : lien)

La commande find : les critères

- ☛ **-newer fichier** : rechercher les fichiers plus récents que le fichier donné
- ☛ **-empty** : Recherche tous les fichiers vides (de longueur zéro) et tous les répertoires vides (qui ne contiennent aucun fichier).
- ☛ **-perm droit** : rechercher selon les droits associés à l'objet.
- ☛ **-user uid** : rechercher selon l'identité du propriétaire
- ☛ **-group gid** : rechercher selon l'identité du groupe
- ☛ **-nouser** : rechercher les objets sans utilisateur
- ☛ **-nogroup** : rechercher les objets sans groupe

La commande find : les critères composés

- ☞ **-a** : ET logique
- ☞ **-o** : OU logique
- ☞ **!** : NON logique
- ☞ **(...)** : parenthèses

Exemples

- ☞ **find /home -size 0 -a -user dsti1 -print** : afficher les noms des fichiers dans l'arborescence /home, de taille 0 octets appartenant à l'utilisateur dsti1.
- ☞ **find / -user dsti1 -o -user duti1 -print** : afficher les fichiers appartenant à l'utilisateur dsti1 ou à l'utilisateur duti1

La commande find : les actions

Action : Explications

- ☛ **-fprint f** : Enregistre le nom du fichier avec son chemin complet dans le fichier nommé f.
- ☛ **-ls** : Affiche le nom du fichier sous la forme de ls
- ☛ **-print** : Affiche les noms des fichiers
- ☛ **-exec C** : Exécute la commande **C** en substituant le nom du fichier en traitement à **{}** dans l'énoncé de la commande.

Exemples

- ☛ **find . -empty -ls** : cherche tous les fichiers vides et les énumère
- ☛ **find / -name "XF86Config" -print** : cherche le fichier XF86Config dans toute l'arborescence depuis la racine
- ☛ **find . -name '*.Z' -print** : spécifie au critère -name des méta-caractères (mettre entre apostrophes pour éviter que le shell lui-même les interprète)

La commande find : les actions

Focus sur -exec Commande

- ☛ **-exec Commande** : Permet de lancer pour chaque fichier un autre programme (**Commande**)
 - i.e. effectue **Commande** sur chaque fichier trouvé par find
- ☛ La séquence de commandes se termine par un **\;**
 - le « ; » est échappé pour être certain que le shell le passe de façon littérale à find
- ☛ Si **COMMANDE** contient **{}**, alors find substitue le chemin complet du fichier en cours à « **{}** »
- ☛ Lorsque find rencontre un -exec, il considère tout ce qui suit comme une ligne de commandes, jusqu'au caractère **;**

La commande find : les actions, focus sur -exec Commande

Exemples

☛ `find / -name "*.rc" -exec echo "J'ai trouvé" \;`

- Affiche « j'ai trouvé » pour chaque un fichier se terminant par *.rc

☛ `find / -name "*.rc" -exec echo "J'ai trouvé" {} \;`

- Identique au précédent mais affiche aussi le fichier concerné

☛ `find / -name "*.rc" -exec echo "J'ai trouvé" \; -print`

- exécute -print comme la prochaine commande à appliquer au fichier « *.rc » après un saut de ligne
- applique par défaut l'opérateur booléen -a et le fichier

☛ `find . (-name a.out -o -name '*.o') -atime +7 -exec rm \;`

- combine des critères et utilise l'action -exec avec la commande rm
- efface tous les fichiers **a.out** et tous ceux portant l'extension **.o** qui n'ont pas été accédés depuis plus de **7** jours

Manuel en ligne

👉 **man** *cmd* : affiche page à page le chapitre du manuel sur la commande *cmd*

Manipulation de fichiers textes

- 🖱 **echo chaîne** : affiche **chaîne** sur la sortie standard (l'écran)
- 🖱 **more fichier** : affiche le contenu de **fichier** page par page. Le passage à la page suivante se fait en appuyant sur espace
- 🖱 **cat fichier** : affiche le contenu de **fichier** sur la sortie standard
- 🖱 **wc [options] fichier** : affiche le nombre de caractères (-c), de mots (-w) ou de lignes (-l) dans fichier
- 🖱 **head -n fichier** : affiche les premiers **n** lignes de **fichier**
- 🖱 **tail -n fichier** : affiche les dernières **n** lignes de **fichier**

Les filtres textes

- ☛ **sort fichier** : trie les lignes de **fichier** (voir les options avec la page man)
- ☛ **diff fich1 fich2** : affiche les lignes de **fich1** et **fich2** qui sont **différentes**.
- ☛ **uniq fichier** : remplacer des lignes consécutives identiques par une seule ligne (supprimer la duplication)
- ☛ **tr str1 str2** : remplacer (convertir une chaîne de caractères) chaque occurrence de **str1** par **str2**.
- ☛ **cut options fichier** : permet d'afficher des zones spécifiques d'un fichier
 - **cut -c1 /etc/passwd** : affichera la première colonne du fichier /etc/passwd.
 - **cut -d: -f6 /etc/passwd** : affichera le 6eme champ du fichier /etc/passwd, dont le séparateur de champs est le caractère double point (":").
 - spécifie un séparateur de champs avec l'option -d
 - voir les options avec la page man

Filtre : La commande grep

☛ **grep motif fichier** : afficher les lignes dans fichier qui contiennent motif.

☛ Quelques options (voir man grep pour une liste complète)

- *-num* : affiche *num* lignes avant et après la ligne où figure le *motif*
- *-B num* : affiche *num* lignes avant la ligne où figure le *motif*
- *-A num* : affiche *num* lignes après la ligne où figure le *motif*
- *-n* : affiche *n* numéro de ligne
- *-c* : affiche le nombre d'occurrence du *motif* dans le fichier
- *-i* : ignore les différences entre minuscules et majuscules
- *-v* : cherche les lignes qui ne contiennent pas le motif

☛ Variations sur le motif

- 1 grep '^[aA]' fich : cherche les lignes qui commencent par a ou A dans le fichier fich.
- 2 grep '[aA]\$' fich : cherche les lignes qui se terminent par a ou A dans le fichier fich.

Sommaire

- 1 Objectifs et Contenu
- 2 Notion de Shell
- 3 Fichiers et Système de Fichiers
- 4 Commandes de base
- 5 Les éditeurs de texte**
 - Les éditeurs vi, vim, nano, etc.
- 6 Exercices d'application

Les éditeurs disponibles sous Linux

Les éditeurs vi et vim

- ☛ L'éditeur le plus usuel sous Linux est vim.
- ☛ Il est compatible avec vi, l'éditeur standard (ISO) des systèmes Unix.
- ☛ C'est un éditeur en mode texte.
- ☛ La connaissance d'un éditeur de ce type est indispensable si l'on travaille sur un serveur
 - le mode graphique y est généralement absent

Les autres éditeurs : nano, emacs

- ☛ La plupart des distributions Linux offrent également l'éditeur **nano**,
 - qui est très simple à utiliser
- ☛ L'éditeur **emacs** qui est très puissant mais réservé aux développeurs.
- ☛ Sous les systèmes possédant une interface graphique, on peut utiliser **gedit** ou **kedit**

Les commandes fondamentales de vi

←↓↑→

ou h, j, k, l

Déplacement dans les quatre directions.

i

Insertion avant le curseur.

a

Insertion après le curseur.

J

Jointure de la ligne courante et de la ligne suivante.

x

Suppression du caractère courant.

dd

Suppression de la ligne courante.

u

Annule la dernière modification.

:q!

Abandon.

:wq

Sauvegarde et sortie de l'éditeur.

Sommaire

- 1 Objectifs et Contenu
- 2 Notion de Shell
- 3 Fichiers et Système de Fichiers
- 4 Commandes de base
- 5 Les éditeurs de texte
- 6 Exercices d'application**

Utilisation des principales commandes (1/3)

a) Lister les fichiers du répertoire courant.

```
$ ls
```

b) Lister les fichiers du répertoire courant avec leurs attributs.

```
$ ls -l
```

c) Lister des fichiers du répertoire /tmp.

```
$ ls -l /tmp
```

d) Idem, mais liste également les fichiers cachés (qui commencent par « . »).

```
$ ls -la /tmp
```

Utilisation des principales commandes (2/3)

e) Copier un fichier, la copie est présente dans le répertoire courant (cp src dst).

```
$ cp /etc/passwd password
```

d) Copier des fichiers dans un répertoire (cp src ... rep_dst).

```
$ cp /etc/issue /etc/group /tmp
```

e) Se déplacer dans le répertoire /tmp, afficher le répertoire courant et détruire les fichiers précédemment copiés. Enfin, revenir dans son répertoire de connexion.

```
$ cd /tmp
```

```
$ pwd
```

```
/tmp
```


Utilisation des principales commandes (3/3)

```
$ rm issue group
```

```
$ cd
```

f) Renommer un fichier, déplacer un fichier dans /tmp.

```
$ mv password pass
```

```
$ mv pass /tmp
```

g) Afficher le type de contenu d'un fichier.

```
$ file /bin/bash
```

h) Créer un répertoire dans le répertoire courant.

```
$ mkdir sous_rep
```

i) Afficher page par page un fichier, le résultat d'une commande.

```
$ more /etc/services
```

```
$ ls -l /etc | more
```

Chapitre 03 : Fichiers et Commandes de base

Dr Mandicou BA

mandicou.ba@esp.sn

<http://www.mandicouba.net>

Diplôme Universitaire de Technologie (DUT, 1^e année)

Diplôme Supérieure de Technologie (DST, 1^e année)

Informatique



ECOLE SUPERIEURE POLYTECHNIQUE

www.esp.sn

