

Commandes Shell II.

- A.Belcaid

Table de Matière

1. Basiques vim
2. Manipulation de fichiers et dossiers.
3. Examinât ion de fichiers.
4. Recherche et Tri

Basiques Vim

Key	Description
:w	Sauvegarde fichier
wq	Sauvegarder et quitter.
q!	Quitter sans sauvegarde.
i	Entrer dans le mode <code>insert</code> .
ESc	Revenir en mode normal.
h j k l	se déplacer
u	undo (annuler)
<C-r>	Reprendre

Exercices

“ Terminer les exercices de 1.1 a 1.6 dans **vimtutor**. ”

Si vous n'avez pas cette commande, installer **vim**.

```
sudo apt install vim
```

Partie 2

Commande	Description
mkdir	Création de répertoire.
rmdir	Suppression de répertoire.
cp	Copie de fichiers.
mv	Déplacement de fichiers.
rm	Suppression de fichiers.

Exercice pratique

On se propose d'ecrire un projet **C++** pour tester la recherche binaire dans un tableau. La structure de notre projet est la suivante:

```
.
├── binary_search.cpp
├── binary_search.h
├── binary_search_test.cpp
├── CMakeLists.txt
├── HELP.md
├── README.md
├── test
│   ├── catch.hpp
│   └── tests-main.cpp
```

Questions

1. Créer cette arborescence.
2. Supprimer le fichier `HELP.md`
3. Renommer le fichier `README.md` ad `readMe.md`
4. Lister les fichier par leur ordre de date modification.
5. Déplacer le fichier `binary_search_test.cpp` au répertoire **test**.
6. On vient de changer d'avis, on veut remettre ce fichier a sa place.

Partie 3

Commande	Description
cat	Afficher le contenu d'un fichier.
less	Contenu d'un fichier dans un pager .
more	Similaire a less.
head	Afficher les premières ligne d'un fichier.
tail	Afficher la fin d'un fichier.
cw	Calculer les mots, caractères et taille.

Exercice

Vous travailler dans un projet `Java` et vous laissez des lignes de la forme **TODO** avant chaque fonction ou code **incomplet**.

Les fichiers de ce projet sont stockés dans dossier `projet`.

“ Ecrire une commande pour afficher toutes les tâches dans ce projet. ”

```
cat "TODO" project/*
```

Compilation

Commande	Description
<code>java className.java</code>	Compiler le programme java.
<code>java className</code>	Exécuter le programme.
<code>gcc, python, ruby ...</code>	Applicable a différents langages

- “ Compiler et exécuter le programme HelloWorld.java ”
- “ Écrire un programme avec **vim** puis compiler le! ”
- “ Comment peut-on compiler toutes les classes dans un projet? ”

Partie 4 : Tri et élimination.

Commande	Description
<code>sort</code>	Trier contenu d'un fichier.
<code>uniq</code>	Éliminer les redondances successive!! .

Sort

La commande `sort` est fréquemment utilisée pour **trier** un texte ou **TSV** selon différents ordres. L'ordre peut être soit *lexicographique* ou *numérique**

- Sans flag, elle trie selon un ordre **lexicographique**.
- L'option `-n` force un tri **numérique** selon un champs.
- Le flag `-r` renverse l'ordre.
- le Flag `-k` spécifie un champs dans le cas (csv, tsv).
- le flag `-t` spécifie un **délimiteur**.

Uniq

La commande `uniq` élimine les redondances successives dans un fichier.

- Le flag `-c` afficher aussi le nombre de chaque occurrence.
- Le flag `-d` affiche seulement les redondances.
- Le flag `-u` affiche seulement qui sont uniques.

Exercices

Le fichier `doctors.tsv` contient des entres séparés par **tabulation**.

```
Dr. Rajendra Prasad      January 26, 1950      May 13, 1962
Dr. S. Radhakrishnan    May 13, 1962         May 13, 1967
Dr. Zakir Hussain       May 13, 1967         August 24, 1969
Shri Varahagiri Venkata Giri      August 24, 1969      August 24, 1974
Shri Fakhruddin Ali Ahmed      August 24, 1974      February 11, 1977
Shri Neelam Sanjiva Reddy      July 25, 1977        July 25, 198
```

1. Trier ce fichier selon le nom des docteurs.
2. Afficher le résultat selon un ordre inversé.
3. Trier maintenant selon leur **année** de naissance.