

# **Systemes d'exploitation**

## **Atelier 1: Commandes Unix**

# Affichage et Redirections

- `echo hello`
- `echo " test d'affichage"`

Redirection de l'affichage vers un fichier:

- `echo "je teste l'affichage dans un fichier" > file1.txt`
- `cat file1.txt`

autre méthode:

- `cat > File Input.text`  
`ceci est le texte d'Input`  
`^C`
- `cat <File Input.text >File1.txt`  
*->input depuis FileInput.txt et output vers File1.txt*

# Navigation système

1. **pwd**: print working directory

2. **ls**, **ls -l**: list

application: créer un fichier contenu.txt contenant le résultat de `ls -l`.

3. **cd**: change directory

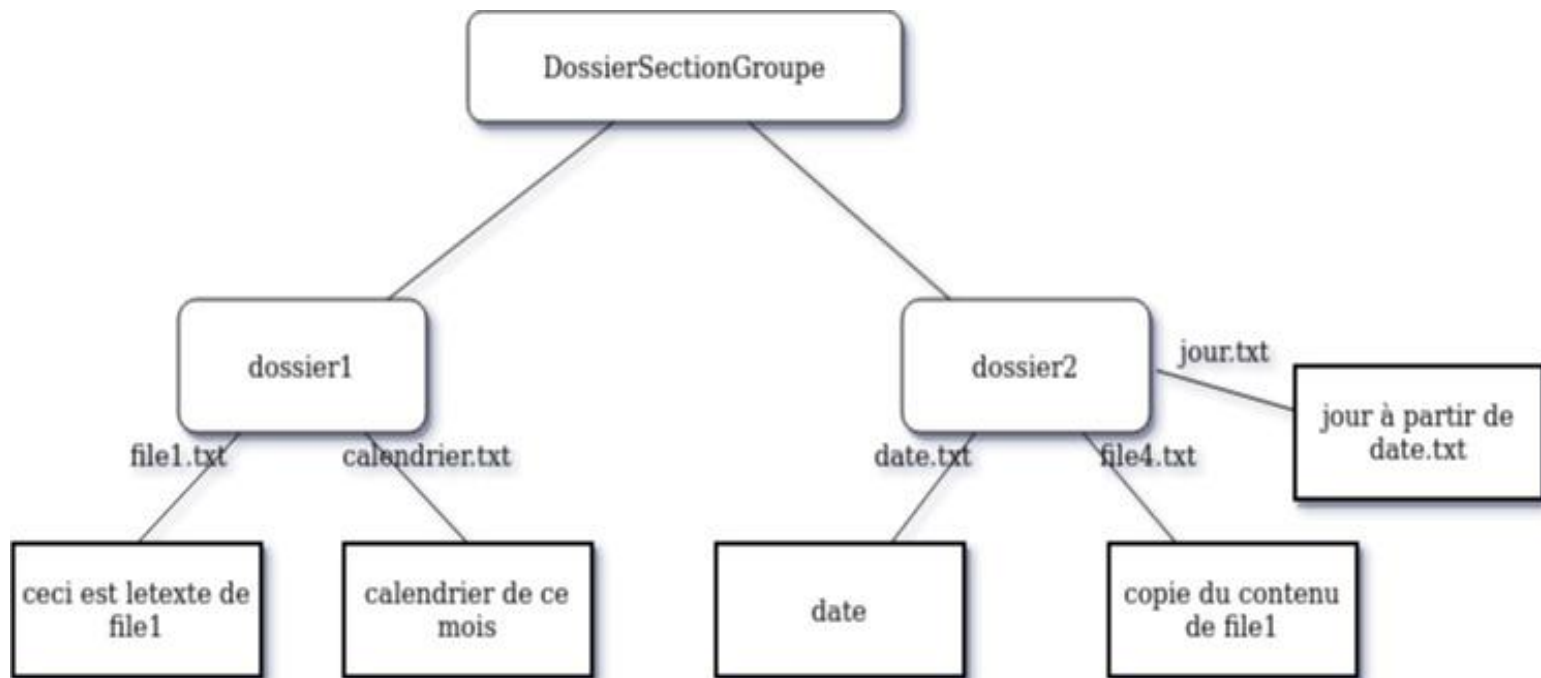
4. **mkdir**, **mkdir -p**. *exemple: mkdir -p chose/truc/machin*: make directory

5. **rm**, **rm -r** (recursive), **cp** source\_file destination,  
**mv** file.txt /path/to/destination/newfile.txt

Question: comment renommer un fichier?

# Application 1

réaliser l'arborescence sous Desktop



## Question bonus

- créer un dossier3 dans lequel créer 20 sous-dossiers: sousdossier1, sousdossier2, ... sousdossier20.
- effacer les 10 premiers sous-dossiers en une seule commande.

# Recherche

- **find**

- \* `find Desktop/` : affiche tout ce qu'il y a dans Desktop
- \* `find -maxdepth 1` : affiche juste le premier niveau
- \* `find -maxdepth 2` : affiche jusqu'au niveau 2
- \* `find -maxdepth 3 -type f` : affiche jusqu'au niveau 3 les types **f**ichiers
- \* `find -maxdepth 2 -type d` : affiche les répertoires.
- \* `find -name "date.txt"` : recherche par nom
- \* `find -type f -size 1024k` : recherche les fichiers de taille = 1024k
- \* `find -type f -size +1024k` : recherche les fichiers de taille **strictement** > 1024k
- \* `find -type f -size -5M` : recherche les fichiers de moins de 5 Mo.

# Application

1. rechercher le fichier jour.txt
2. rechercher tous les fichiers de type txt
3. rechercher tous les fichiers commençant par c.
4. rechercher dans **home** tous les dossiers dont la taille < 5k.
5. rechercher dans **home** les fichiers dont la taille est entre 5 et 20 M.
6. créer un fichier contenant le résultat de la question 3.
7. copier ce fichier dans un dossier résultat crée dans votre Bureau

# Tri dans un fichier

- \* **sort** words.txt: tri des résultats sans changer le fichier d'origine.
- \* **sort -r** words.txt : tri inversé.

exercice:

1. télécharger le fichier **words.txt**.
2. créer un fichier wordsTri.txt avec le contenu trié de words.txt.
3. créer un fichier wordsTriUnique.txt avec le **contenu trié et sans redondance** de words.txt.



# Recherche dans un fichier: grep

- \* **grep** e words.txt : recherche la lettre e dans words.txt
- \* grep e words.txt | wc -l : nombre de lignes qui contiennent e
- \* grep **-c** e words.txt : la même ça donne aussi le nombre de lignes.

## application:

- rechercher dans le fichier words.txt, la lettre a.
- calculer le nombre de lignes contenant 'a' et mettre le résultat dans un fichier resultat.txt.

# Batch Scripting

- `which $SHELL`: permet de savoir où se trouve le shell (`/bin/bash`)
- `nano myscript.sh`
- commencer par `#!/bin/bash`
- la liste des commandes voulues: exemple.

```
mkdir ~/Desktop/myscript
```

```
cd ~/Desktop/magic
```

```
touch file{1..100}
```

```
ls -l ~/Desktop/magic > ~/Desktop/magic.log
```

# Application

- Réaliser un script bash qui crée deux dossiers d1 et d2 dans Documents.
- Créer les fichiers calendrier\_annuel.txt et date.txt dans d1
- Afficher "liste des fichiers dans d1"
- Afficher la liste grâce à la commande ls
- Afficher "copie de fichiers de d1 dans d2"
- copier les fichiers de d1 vers d2