

# R1.04 – Cours 3

## Redirections, filtres

Département Informatique

IUT2, UGA

2022/2023

# Plan du cours

- 1 Documentation en ligne
- 2 Redirection d'entrées/sorties
- 3 Filtres
- 4 Résumé

# Plan du cours

- 1 Documentation en ligne
- 2 Redirection d'entrées/sorties
- 3 Filtres
- 4 Résumé

# Types de documentation

- Documentation intégrée au logiciel
- Pages de manuel (dans `/usr/share/man/`)  
informations minimales sur les logiciels
- Pages info (dans `/usr/share/info/`)  
informations plus détaillées sur les logiciels (mais pas tous)
- Autres docs : fichiers dans `/usr/share/doc/`
  - Texte
  - HTML
  - PostScript (PS)
  - PDF
  - ...

# Documentation dans le *shell*

- Doc intégrée : `ex : COMMANDE --help`  
exemple `rm --help`
- Page de manuel en anglais : `man COMMANDE`
- Page de manuel en français : `man-fr COMMANDE`  
(commande locale à l'IUT2)
- Page info : `info COMMANDE`
- Autres docs
  - `less FICHIER` (pour fichier texte)
  - `zless FICHIER` (pour fichier texte compressé avec gzip)

# Documentation dans un navigateur Web

- Pages de manuel

`http://transit/cgi-bin/man/man2html`

- Pages info

`http://transit/cgi-bin/info2www`

- Autres docs

- dwww

`http://transit/dwww/menu/`

- doc-central

`http://transit/dc/`

# Documentation dans un navigateur graphique

## Logiciel `yelp`

- Pages de manuel
- Pages info
- Logiciels GNOME
- ...

# Exemples de page de manuel : ls

## NAME

ls - list directory contents

## SYNOPSIS

ls [OPTION]... [FILE]...

## DESCRIPTION

List information about the FILES (the current directory by default).  
Sort entries alphabetically if none of -cftuvSUX nor --sort.

Mandatory arguments to long options are mandatory for short options too.

-a, --all  
do not ignore entries starting with .

[...]



# Exemple de page de manuel : ps

## NAME

ps - report a snapshot of the current processes.

## SYNOPSIS

ps [options]

## DESCRIPTION

ps displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use top(1) instead.

[...]

## EXAMPLES

To see every process on the system using standard syntax:

```
ps -e  
ps -ef  
ps -eF  
ps -ely
```

[...]

# Conventions syntaxiques des pages de manuel

Pour spécifier de façon non ambiguë la syntaxe d'une commande, les conventions suivantes sont utilisées :

- Mot entre crochets : [mot]  
 → mot est facultatif  
 il peut donc apparaître 0 ou 1 fois
- Mot avec points de suspension : mot...  
 → mot peut apparaître entre 1 et n fois
- Mot avec crochets et points de suspension : [mot]...  
 → mot peut apparaître entre 0 et n fois
- Mots séparés par une barre verticale : mot1|mot2  
 → on peut taper soit mot1, soit mot2
- Mot en gras et en minuscule : **mot**  
 → mot devra être tapé tel quel
- Mot en majuscules, non gras : MOT  
 → représente un terme général qui devra être instancié

# Plan du cours

- 1 Documentation en ligne
- 2 Redirection d'entrées/sorties
- 3 Filtres
- 4 Résumé

# Entrée et sorties standard d'un processus

- Pourquoi «standard» ?
- Entrée standard
  - exemple : programme Java qui utilise `System.in`
- Sortie standard
  - exemple : programme Java qui utilise `System.out`  
en particulier `System.out.println()`
  - exemple : programme `ls` qui liste des fichiers
- Sortie d'erreur standard
  - exemple : commande `ls -z` qui affiche un message d'erreur  
`ls: invalid option -- 'z'`  
`Try 'ls --help' for more information.`

# Schéma d'un logiciel exécuté dans un terminal

## État initial avant le lancement

- Processus : terminal, shell
- Plomberie initiale entre terminal et shell
  - Connexions : clavier → terminal → shell
  - Connexions : shell → terminal → écran

## Lancement du logiciel en avant plan

- Processus : terminal, shell, logiciel
- Plomberie après le lancement
  - Connexions : clavier → terminal → logiciel
  - Connexions : logiciel → terminal → écran

# Redirections vers un fichier

- Exemple de cas d'utilisation :  
commande `ls -l` trop longue  
pour faire un copier/coller dans un mail
- Exemple de commande :  
`ls -l *.mkv > ma-liste-de-films`
- Redirection de la sortie standard  
`COMMANDE > FICHIER`
- Redirection de l'erreur standard  
`COMMANDE 2> FICHIER`
- Redirection de la sortie **et** de l'erreur standard  
`COMMANDE &> FICHIER`
- Redirection de l'entrée standard  
`COMMANDE < FICHIER`  
exemple d'utilisation : tests répétitifs d'un programme Java  
qui demande des entrées

# La commande `cat`

- Rappel : afficher le contenu d'un fichier  
`cat ma-liste`
- La commande est plus générale
- Elle permet de **concaténer** plusieurs fichiers
- Exemple  
`cat ma-liste*`

# Redirection avec ajout à la fin

- Si le fichier existe déjà (ou pas)  
`COMMANDE >> FICHIER`
- Exemple d'utilisation :  
depuis un 2ème répertoire :  
`ls -l >> ~/ma-liste-de-films`
- Autre exemple :  
`cat fichier1.txt > gros-fichier.txt`  
(équivalent à une commande `cp`)  
`cat fichier2.txt >> gros-fichier.txt`
- Façon plus simple de procéder :  
`cat *.txt > gros-fichier.txt`



# Redirection vers un autre processus : tubes

- Forme générale :

```
COMMANDE1 | COMMANDE2
```

- Exemple :

```
cat ma-liste-de-films | sort
```

- Forme plus courte :

```
sort ma-liste-de-films
```

- Combinaison avec redirection :

```
cat ma-liste-de-films | sort > ma-liste-de-films-triée
```

- Exécution par le *shell* avec 2 processus et un tube

# Plan du cours

- 1 Documentation en ligne
- 2 Redirection d'entrées/sorties
- 3 Filtres**
- 4 Résumé

# Définition

- Les filtres sont des commandes utiles en sortie d'un tube
- Il existe des filtres pour
  - trier : `sort`
  - extraire des lignes : `grep`, `head`, `tail`
  - extraire des colonnes : `cut`
  - compter : `wc`
  - ...

# Exemples de tris avec `sort`

- On part de 2 listes de films (ou plus)
- On crée une liste unique triée

```
cat ma-liste-de-films* | sort
```

# Exemples de recherches avec grep

- Rechercher dans un fichier texte avec grep  
Avec ou sans tube
  - `cat Avare.txt | grep MOT`
  - `grep MOT Avare.txt`
- Si ce que l'on cherche contient des caractères spéciaux du *shell*
  - `cat Avare.txt | grep 'CHAINE'`
  - `grep 'CHAINE' Avare.txt`
- Rechercher un fichier dans un répertoire  
`ls -l | grep toto`  
Remarque : autre façon de faire : `ls -l *toto*`
- Rechercher un processus  
`ps aux | grep toto`

# Exemples de comptages avec `wc`

- `wc` vient de « *word count* »
- Compter dans un fichier texte  
`wc Avare.txt`  
→ nombre de lignes, mots, caractères
- Compter tous les processus  
`ps aux | wc -l`
- Compter un nombre de fichiers  
`ls -l | wc -l`

# Tubes et filtres multiples

- On peut enchaîner plusieurs filtres
- Exemple : compter le nombre d'occurrences d'un mot dans un fichier
  - `cat Avare.txt | grep père | wc -l`
  - `grep père Avare.txt | wc -l`
- Exemple : compter mes processus
  - `ps aux | grep NOM_DE_LOGIN | wc -l`
- Exemple : combien ai-je de fichiers dont le nom contient «toto»
  - `ls -l | grep toto | wc -l`

# Plan du cours

- 1 Documentation en ligne
- 2 Redirection d'entrées/sorties
- 3 Filtres
- 4 Résumé**



# Résumé

- Je sais trouver les options des commandes dans les docs
- Je sais combiner des commandes avec des tubes et des filtres