7. Les commandes Linux

- 7.1 La commande Linux
- 7.2 La redirection des entrées-sorties
- 7.3 Les tubes de communication et les filtres
- 7.4 Les tâches en arrière-plan
- 7.5 La substitution de commande
- 7.6 Les commandes groupées
- 7.7 Les caractères spéciaux
- 7.8 Les caractères de neutralisation
- 7.9 Exercices

7. Les commandes Linux

7.1 La commande Linux sous Bash

- Syntaxe générale
  - Commande [± option...] [paramètre...]
  - Exemple :
    - rm -ir f1 f2 f3 d1
    - rm –interactive --recursive f1 f2 f3 d1
- Ligne de commandes séquentielles
  - pwd; who; ls
- La commande sur plus d'une ligne
  - ls -l /home/olive/scripts**\<return>**
  - /CREATE-USER/*<return>

7. Les commandes Linux

7.1 La commande Linux sous Bash

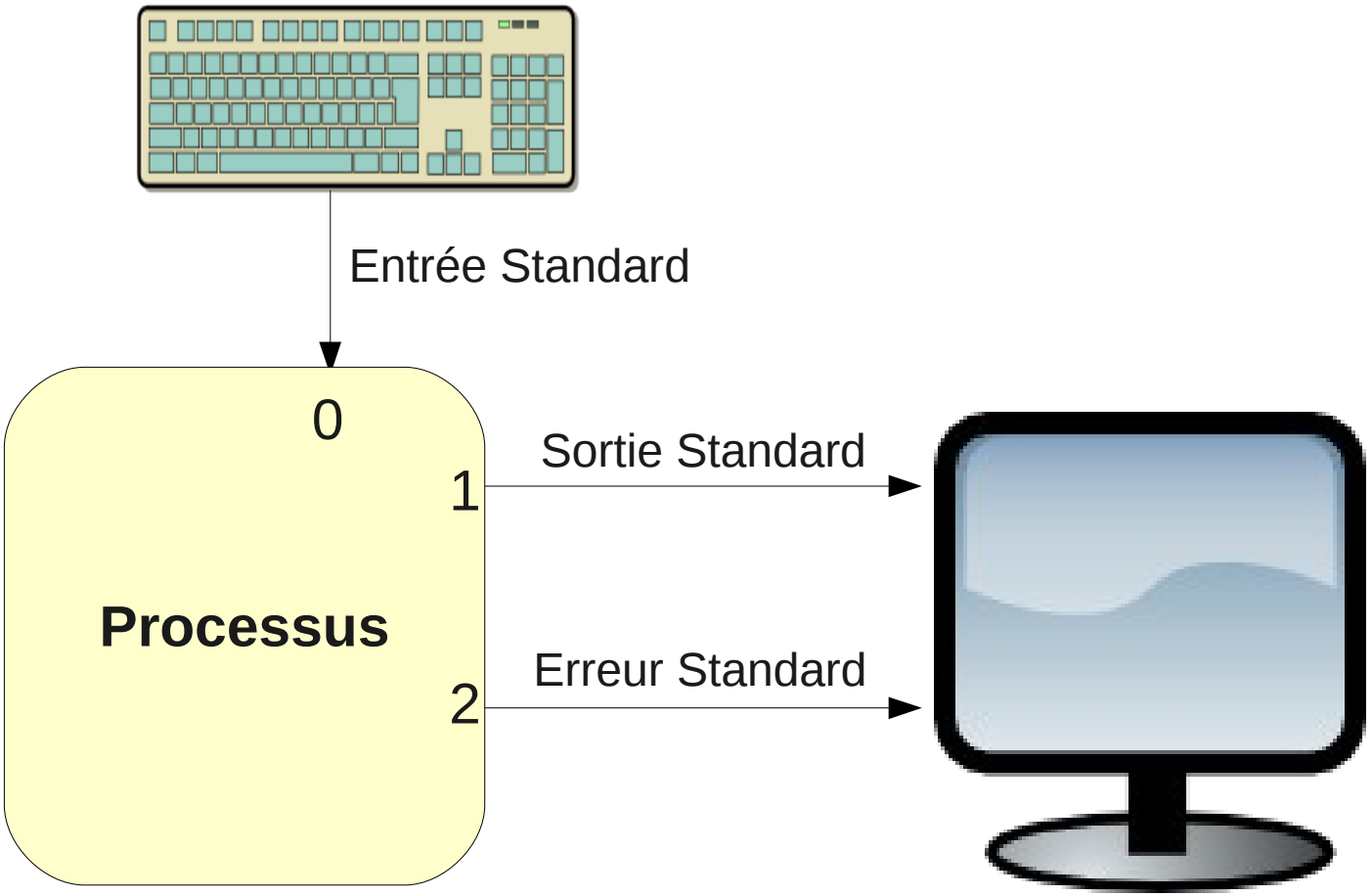
Les séparateurs conditionnels de commande

- **&&** (ET logique)
- **||** (OU logique)
- Exemple :
- **cd scripts && ls** exécutera 'ls' si 'cd scripts' retourne aucune erreur
- **cd test || mkdir test** exécutera 'mkdir test' si 'cd test' retourne une erreur

7. Les commandes Linux

7.2 La redirection des entrées-sorties

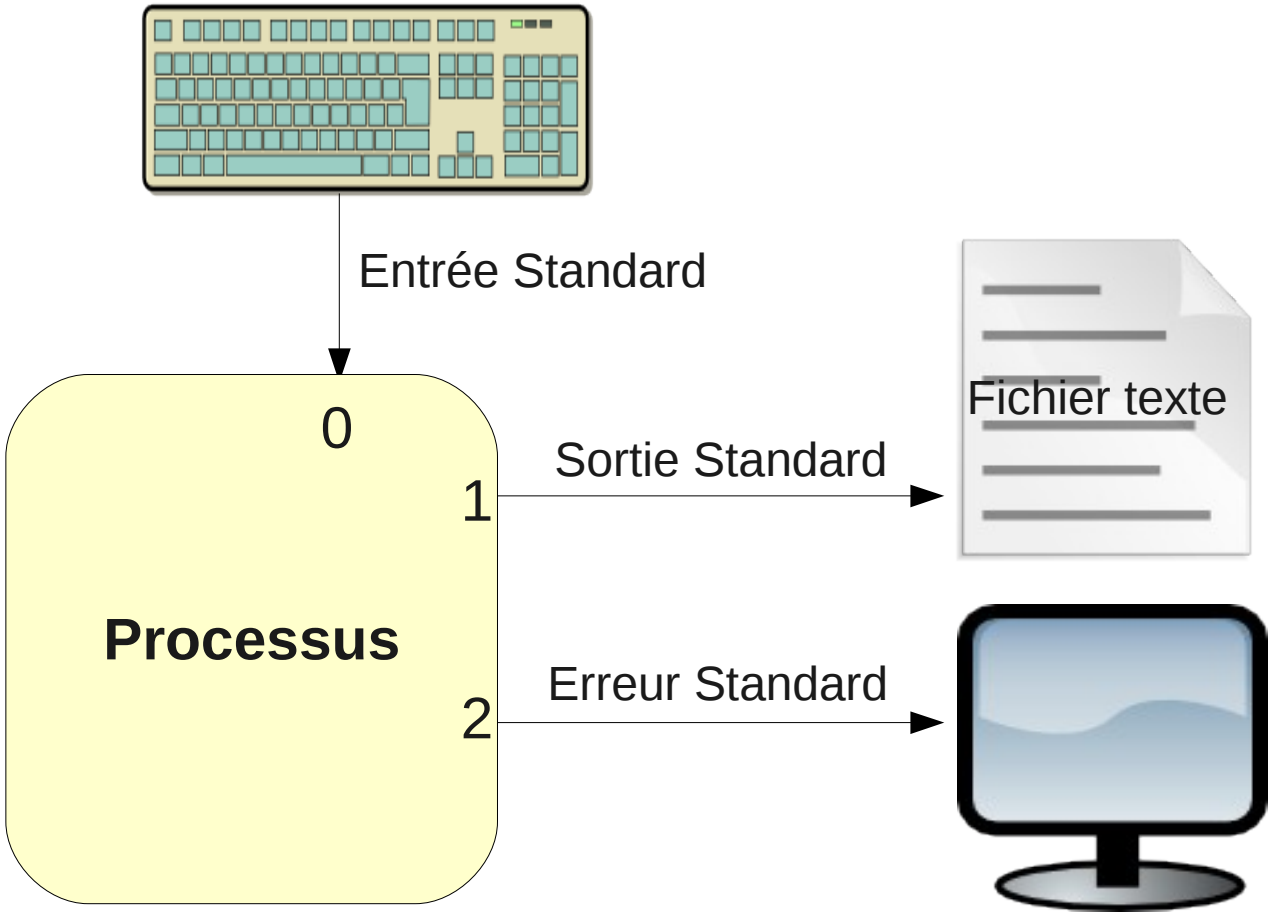
Entrée/sorties standard d'un processus



7. Les commandes Linux

7.2 La redirection des entrées-sorties

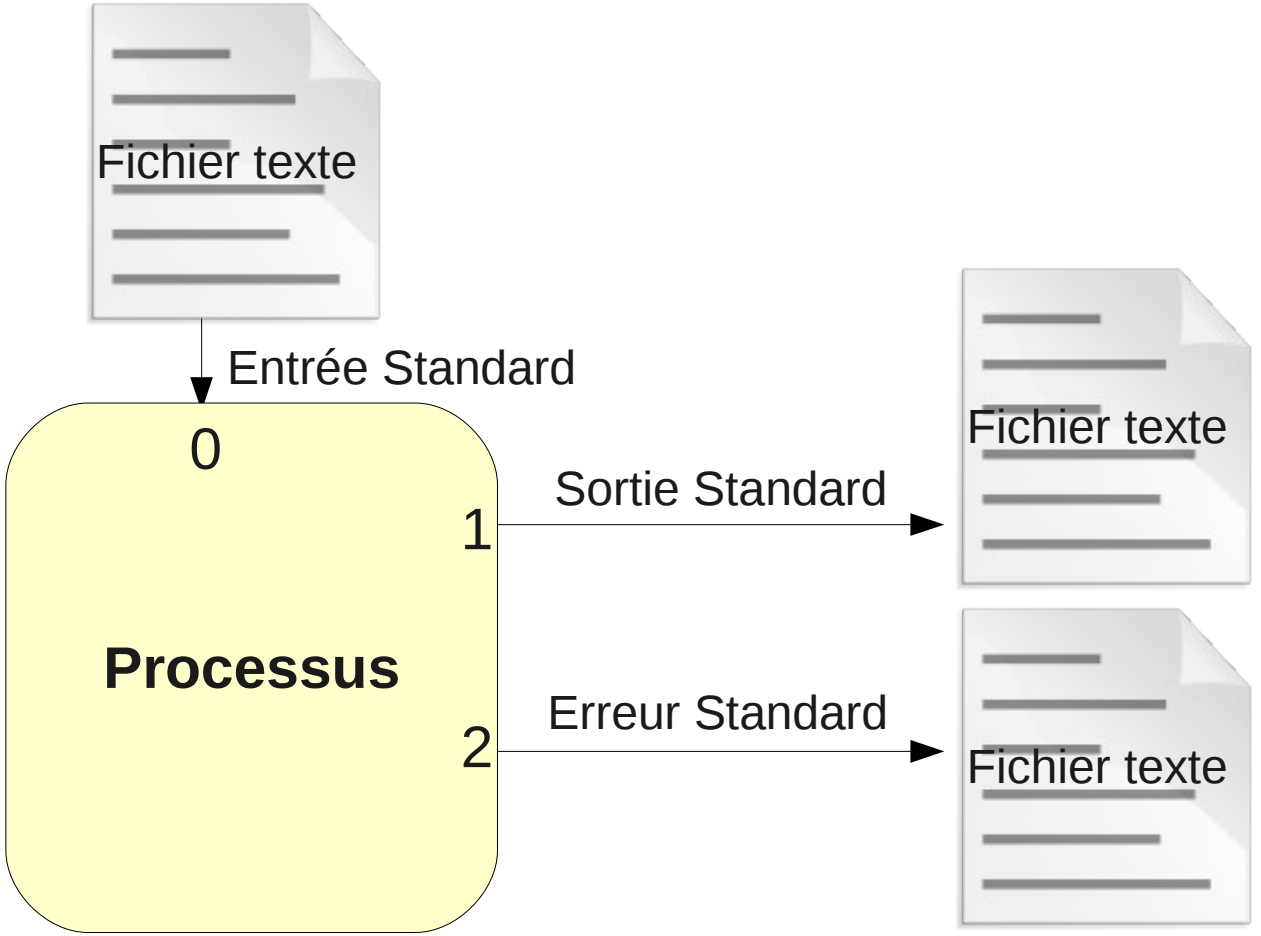
Redirection de la sortie standard vers un fichier



7. Les commandes Linux

7.2 La redirection des entrées-sorties

Redirection vers des fichiers



7. Les commandes Linux

7.2 La redirection des entrées-sorties

- Exemples de redirection
  - ls **>** liste.txt
  - ls **>>** liste.txt
  - ls ./toto **2>**erreur.log
  - ls * ./toto **>**liste.txt **2>**erreur/log
  - ls * ./toto **&>**liste.txt
- **cat** et les redirections
  - cat f1 > f2 idem cat <f1 >f2
  - cat f1 f2 f3 > f123
  - cat f2 >> f1

7. Les commandes Linux

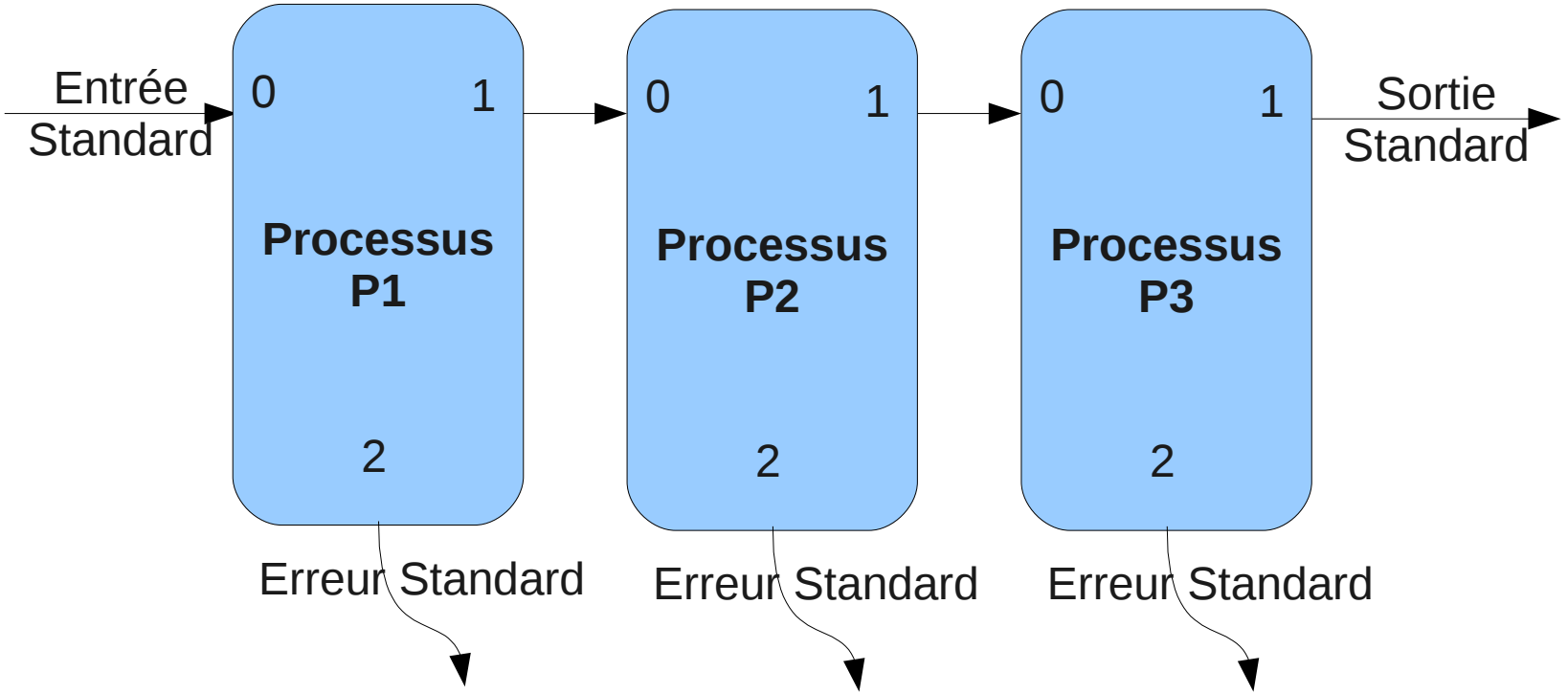
7.2 La redirection des entrées-sorties

- **cat** et les redirections (suite)
  - cat >f1
    - Salut
    - <ctrl-d>
    -
  - cat <<FIN >f1
    - Echo "Bienvenue dans le monde Linux"
    - Echo "A bientôt"
    - FIN

7. Les commandes Linux

7.3 Les tubes de communication (pipe) et les filtres

- Les **tubes** ou **pipes** : flot de données reliant une sortie standard d'une commande vers une entrée standard d'une autre commande. Ci-dessous : **P1 | P2 | P3**.



7. Les commandes Linux

- 7.3 Les tubes de communication (pipe) et les filtres
  - **Exemples** :
    - ls -l | less
    - ls -l | grep "rwxr-xr-x" | less
    - ls -l > f1; grep "rwxr-xr-x" <f1 >f2; less f2; rm f1f2
    - who | wc -l
    - ls | wc -w
    - find . -name scripts | grep coccinella
    - ls *~ | xargs rm
    - find . -name *~ | xargs rm
    - find . -name "*~" -exec rm {} \;

7. Les commandes Linux

7.3 Les tubes de communication (pipe) et les filtres

Utilisation des filtres

- **grep** recherche les occurrences d'une chaîne
- **egrep = grep -E, rgrep = grep -r ou -R, fgrep = grep -F**
- **wc** compte le nbre de caractères(octets), mots et lignes
- **less** affiche l'entrée standard page par page
- **dd** filtre de conversion
- **sed** éditeur en ligne pour filtrer et transformer du texte
- **awk** langage de manipulation de motifs (patterns)
- **sort** filtre de tri de textes et de lignes

7. Les commandes Linux

- 7.3 Les tubes de communication (pipe) et les filtres
  - La commande **xargs**
    - Certaines commandes ne savent pas lire leur entrée standard
      - → **ls** , **rm**, **cp**, **ln**, **mv**
    -
  - **Exemple** :
    - ls | rm aucun sens
    - → ls | xargs rm

7. Les commandes Linux

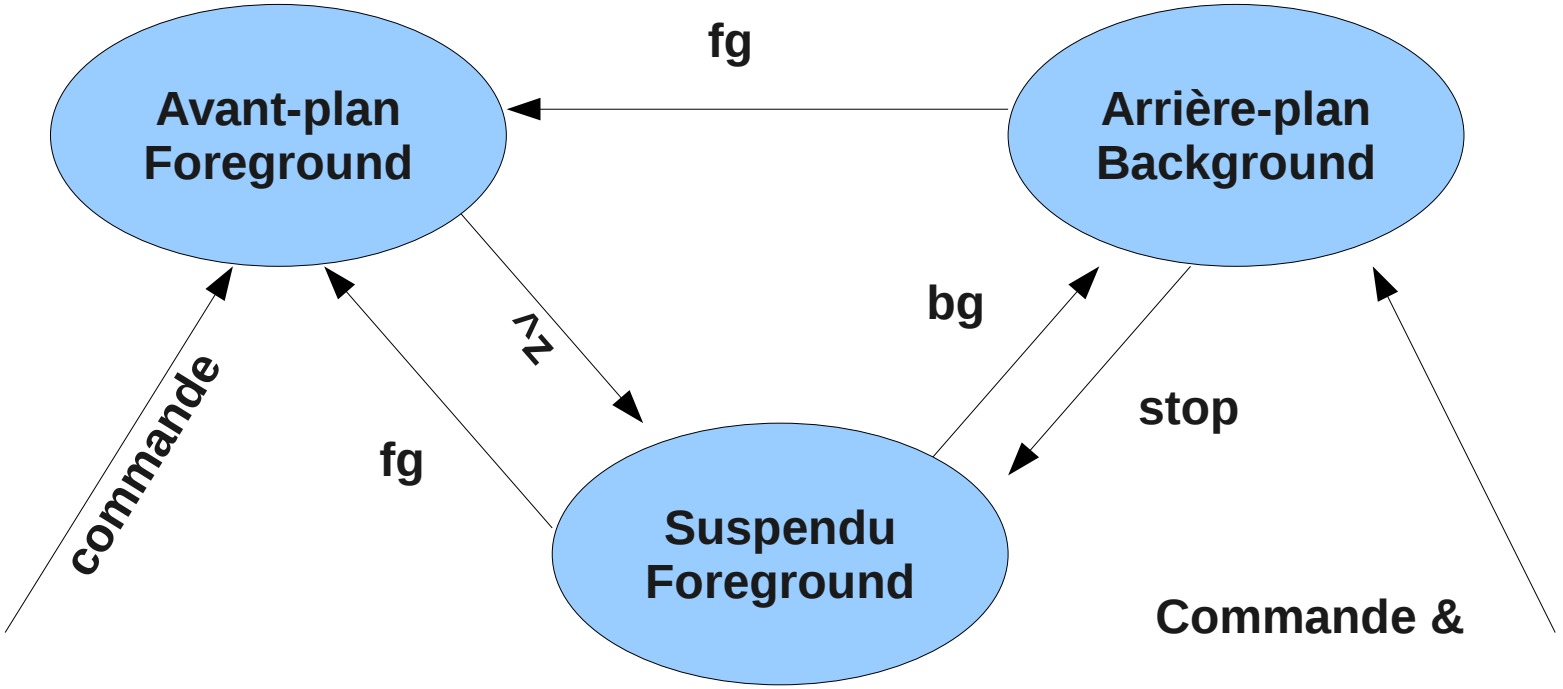
7.4 Tâches en arrière plan

- Permet de lancer une ou plusieurs commandes (longue) et reprendre la main dans la console.
- Valable si la commande ne nécessite pas de dialogue
- **commande&**
- ➔ Pour arrêter la commande
  - Sortir de la session
  - Envoyer un signal au processus: kill -signal PID (%num_job...)
- ➔ Processus non interrompu même si on sort de la session
  - **nohup commande &**

7. Les commandes Linux

7.5 Les différents états d'une tâche (job)

- **Un job est une ligne de commande(s), composé d'un ou plusieurs processus dans le cas d'utilisation de tubes.**



7. Les commandes Linux

7.5 La substitution de commande (backquoting)

- **Utilise le résultat d'une commande comme argument d'une autre commande**
- **`commande`** ou **$(commande)**
- **Exemples** :
  - echo pwd
  - echo 'pwd'
  - echo `pwd`
  - echo **$(**pwd**)**
  - echo "Il y a **$(**who | wc -l**)** utilisateurs connectés"
  - grep -n "motif" **$(**find . -type f print**)**
  - rm -i **$(**find . -mtime +20 -print**)**

7. Les commandes Linux

- 7.6 Les commandes groupées
  - **(cmd1; cmd2; ....;cmdn)**
  - Cet ensemble est exécuté par un nouveau processus shell
  - **Exemple** :
  - cd; cd projet; rm *.o; pwd =>/home/olive/projet
  - cd; **(cd projet; rm *.o)**; pwd =>/home/olive
    - **Considérée comme une seule commande au niveau de la redirection des entrées-sorties.**
    - **Peut être lancée en arrière plan.**

7. Les commandes Linux

- 7.7 Les caractères spéciaux générateurs de noms de fichiers (ou métacaractères)
  - Désigne un ensemble d'objets
    - Noms de fichiers,
    - Chaînes de caractères (**expressions régulières** ou **regexp**).
  - Utilisés par
    - les commandes d'édition (vi, ex, sed,...),
    - Les filtres (grep, egrep, awk, sort...).

7. Les commandes Linux

- 7.7 Les caractères spéciaux générateurs de noms de fichiers (ou métacaractères)
  - ***** toutes chaîne de caractères
  - **?** un caractère quelconque
  - **[...]** liste de caractères par ordre alphabétique si séparés par -
  - **[!...]** liste de caractères à exclure
  - **Exemples** :
  - a[!a-z]s
  - a[a-z0-9A-Z]s

7. Les commandes Linux

7.8 Les caractères de neutralisation

- Il est nécessaire dans certains cas d'empêcher le shell d'interpréter les caractères spéciaux : **<**, **>**, *****, **?**, **!**, **<return>**, ...
- **** (backslash)
- **'** (quote)
- **"** (double-quotes) mais certains métacaractères sont interprétés
- **Exemple** :
- touch 'f*?1' ou touch f*\?1
- echo 'je suis $PWD' et echo "je suis $PWD"

7. Les commandes Linux

7.9 Exercices

- 1. Créer un fichier de nom -i et le supprimer
- 2. Alias permettant de lister page par page et dans l'ordre alphabétique l'ensemble des variables d'environnement.
- 3. Créer dans un répertoire rep1 les fichiers suivants : *fich1, fich2, fich11, fich12, fich1a, ficha1, .fich1, .fich2, toto, afich*.
  - Lister les fichiers :
    - 1/ dont les noms commencent par fich,
    - 2/ dont les noms commencent par fich suivi d'un seul caractère,
    - 3/ dont les noms commencent par fich suivi d'un chiffre,
    - 4/ dont les noms commencent par .,
    - 5/ dont les noms ne commencent pas par f,
    - 6/ dont les noms contiennent fich

7. Les commandes Linux

7.9 Exercices solutions

- 1. Créer un fichier de nom -i et le supprimer (cat)
  - cat >-i
  - rm -i et rm \-i (ne fonctionnent pas)
  - rm ./-i
  -
- 2. Alias permettant de lister page par page et dans l'ordre alphabétique l'ensemble des variables d'environnement.
  - alias env=''alias|env|sort''

7. Les commandes Linux

- 7.9 Exercices solutions
  - 3. Créer dans un répertoire rep1 les fichiers suivants : *fich1, fich2, fich11, fich12, fich1a, ficha1, .fich1, .fich2, toto, afich*.
    - **mkdir rep1; cd rep1; touch *fich1, fich2, fich11, fich12, fich1a, ficha1, .fich1, .fich2, toto, afich***
    - Lister les fichiers :
      - 1/ dont les noms commencent par fich,
      - **ls fich***
      - 2/ dont les noms commencent par fich suivi d'un seul caractère,
      - **ls fich?**

7. Les commandes Linux

- 7.9 Exercices solutions
  - 3. Créer dans un répertoire rep1 les fichiers suivants : *fich1, fich2, fich11, fich12, fich1a, ficha1, .fich1, .fich2, toto, afich*.
    - 3/ dont les noms commencent par fich suivi d'un chiffre,
    - **ls fich[0-9]**
    - 4/ dont les noms commencent par .,
    - **ls .??* (ne pas prendre . Et ..)**
    - 5/ dont les noms ne commencent pas par f,
    - **ls [!f]***
    - 6/ dont les noms contiennent fich.
    - **ls *fich***