



# Sujet 103

# Commandes GNU & Unix

# Commandes GNU & Unix

- 103.1 : Travail en ligne de commande (Val. 4)
- 103.2 : Traitement de flux de type texte avec des filtres (Val. 3)
- 103.3 : Gestion élémentaire des fichiers (Val. 4)
- **103.4 : Utilisation des flux, des tubes et des redirections (Val. 4)**
- 103.5 : Création, contrôle et interruption des processus (Val. 4)
- 103.6 : Modification des priorités des processus (Val. 2)
- 103.7 : Recherche dans des fichiers texte avec les expressions rationnelles (Val. 2)
- 103.8 : Édition de fichiers texte avec vi (Val. 3)

# 103.4

## Utilisation des flux, des tubes et des redirections (Val. 4)

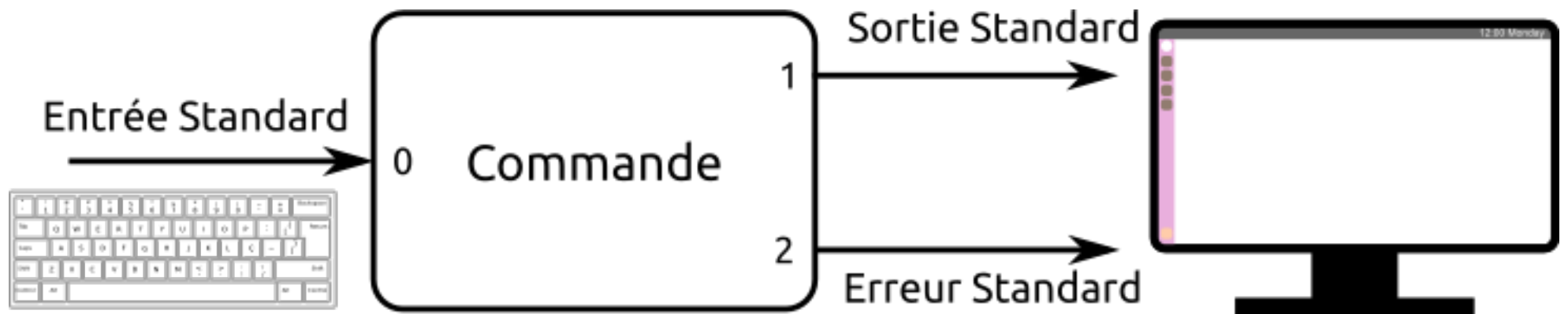
# Utilisation des flux, des tubes et des redirections



- Description : Les candidats doivent être en mesure de **rediriger** des flux et de les associer pour traiter efficacement des données textuelles. Ces tâches incluent les redirections de l'**entrée standard**, de la **sortie standard** et de l'**erreur standard**, la redirection de la sortie d'une commande vers l'entrée d'une autre, l'utilisation de la sortie d'une commande comme paramètres pour une autre commande et l'envoi de la sortie à la fois sur la sortie standard et dans un fichier.
- Termes, fichiers et utilitaires utilisés pour cet objectif :
  - tee
  - xargs

# Flux de données

- GNU/Linux fonctionne avec trois types de flux de données :
  - Entrée standard **stdin** : Descripteur 0 (clavier).
  - Sortie standard **stdout** : Descripteur 1 (écran).
  - Sortie d'erreur standard **stderr** : Descripteur 2 (écran).



# Redirection

- La redirection permet de changer l'entrée, la sortie ou la sortie d'erreur d'une commande.
- La redirection se fait généralement vers ou depuis un fichier.
- La redirection se fait de manière transparente pour les commandes.

# Redirection de sortie (I)

- Le caractère **>** est utilisé pour rediriger la sortie standard.

**commande > chemin**

- Si le fichier de redirection n'existe pas il sera créé. S'il existe, son contenu sera **écrasé. Même si la commande est incorrecte.**
- Le Shell commence d'abord par créer le fichier puis exécute la commande.

# Redirection de sortie (II)

- Les caractères **>>** (double redirection) permettent aussi de rediriger la sortie standard.

**commande >> chemin**

- Si le fichier de redirection n'existe pas il sera créé. S'il existe, la sortie de la commande est ajoutée au contenu du fichier, **sans l'écraser.**



# Redirection de l'entrée (I)

- Le caractère **<** permet de spécifier une autre entrée que l'entrée standard.

**commande < chemin**

- Les caractères **<<** permettent de lire l'entrée jusqu'à ce que la commande rencontre une certaine chaîne de caractères donnée.

**commande << chaîne**

# Redirection de la sortie d'erreur

- On utilise le descripteur 2 de la sortie standard.  
**commande 2> chemin**
- Si le fichier de redirection n'existe pas il sera créé. S'il existe, son contenu sera **écrasé. Même si la commande est incorrecte.**

# Redirection des deux sorties

- Les caractères **>&** permettent de rediriger la sortie erreur et la sortie standard vers un même fichier.

**commande > chemin 2>&1**

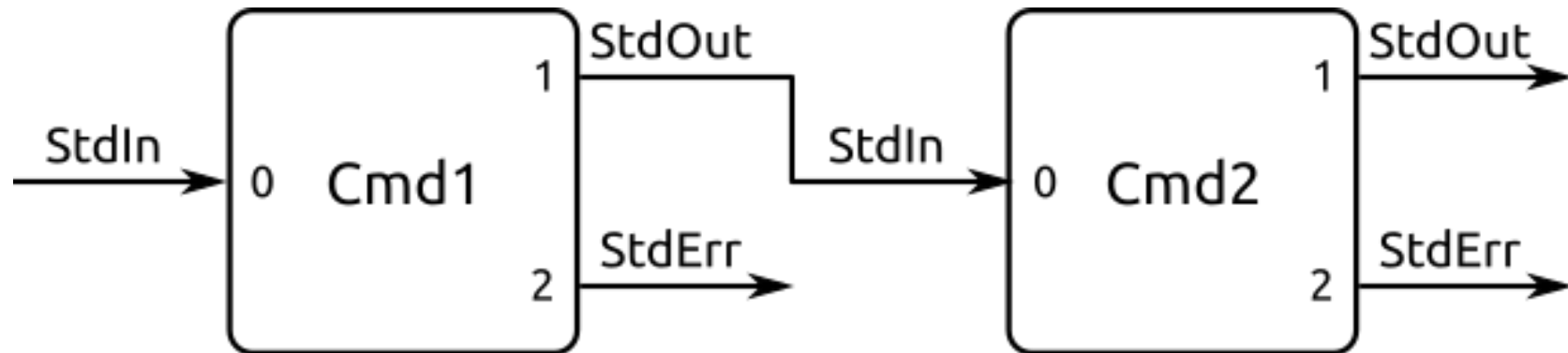
**commande >& chemin**

# Redirection – Examples

- `echo "Bonjour" > message.txt`
- `ls -l > message.txt`
- `wc < message.txt`
- `date >> message.txt`
- `cat < message.txt > fichier.txt`
- `tr "[a-z]" "[A-Z]" << fin`
- `gcc prog.c 2> erreurs.txt`
- `gcc prog.c > erreurs.txt 2>&1`
- `gcc prog.c >& erreurs.txt`

# Tubes

- Le tube (pipe) permet de faire en sorte que la sortie d'une commande (le résultat) devient l'entrée d'une autre.
- Le caractère `|` représente un tube.
- Il est possible d'enchaîner redirections et tubes.



# Tubes – Exemples

- `ls -a /bin > liste.txt ; wc -l < liste.txt`
- `ls -a /bin | wc -l`
- `head /etc/passwd | tail -5 > comptes.txt`
- `cat /etc/passwd | cut -d: -f3 | sort -n`
- `cat /etc/passwd | cut -d: -f7 | sort | uniq  
| wc -l`

# tee

- Dupliquer la sortie standard.
  - Sauvegarder le résultat intermédiaire.
  - Effectuer un traitement supplémentaire.

## **tee [-a] chemin**

- Par défaut si le fichier existe il sera écrasé. L'option **a** permet d'ajouter la sortie à la fin du fichier.
- Exemple :
  - **cut -d: -f1 < /etc/passwd | tee noms.txt | wc -l**

# xargs

- Passer la sortie d'une commande comme arguments à une autre commande.
- Exemples :
  - `cut -d: -f1 < /etc/passwd | sort | xargs echo`
  - `find /tmp -name "*.tmp" | xargs rm -f`
  - `find . -name "*.c" | xargs grep "stdlib.h"`
  - `find . -name "*.c" -print0 | xargs -0 grep "stdlib.h"`