

Système d'exploitation

Système d'exploitation - Utilisation

Mickael Rouvier

CERI – Avignon Université
`mickael.rouvier@univ-avignon.fr`





Section 1

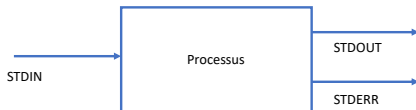
Redirection de flux



Les entrées/sorties des processus

Chaque processus possède 3 flux standards qu'il utilise pour communiquer en général avec l'utilisateur :

- l'**entrée standard** nommée : *stdin* (identifiant du flux : 0) – il s'agit par défaut du clavier
- la **sortie standard** nommée : *stdout* (identifiant du flux : 1) – il s'agit par défaut de l'écran
- la **sortie d'erreur standard** nommée : *stderr* (identifiant du flux : 2) – il s'agit par défaut de l'écran



Sortie standard : stdout

Exemple : programme qui affiche sur la sortie standard (stdout)

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char** argv) {
6
7     cout<<"Exemple de sortie standard (stdout)"<<endl;
8
9     return 0;
10 }
```

Listing 1 – "Sortie standard (stdout)"

Sortie d'erreur : stderr

Exemple : programme qui affiche sur la sortie d'erreur (stderr)

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char** argv) {
6
7     cerr<<"Exemple de sortie d'erreur (stderr)"<<endl;
8
9     return 0;
10 }
```

Listing 2 – "Sortie d'erreur (stderr)"

Entrée standard : stdin

Exemple : programme qui récupère les informations sur l'entrée standard (stdin)

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char** argv) {
6
7     string x;
8     cin>>x;
9
10    return 0;
11 }
```

Listing 3 – "Entrée standard (stdin)"

Rediriger la sortie standard

- **Redirection :**

- Quand on exécute une commande, le shell affiche le résultat sur la console de sortie (l'écran par défaut).
- On peut rediriger cette sortie vers un fichier en utilisant le signe >

```
invite/> ./programme > file.txt
```

- **Concaténation :**

- Au lieu de créer un fichier, il est possible d'ajouter les sorties d'un processus à un fichier existant en utilisant le double signe »

```
invite/> ./programme >> file.txt
```

- **Syntaxe complète :**

- Les signes > peuvent être précédés de l'identifiant du flux à rediriger. Pour la sortie standard, on peut donc utiliser les syntaxes suivantes.

```
invite/> ./programme 1> file.txt  
invite/> ./programme 1>> file.txt
```

Rediriger la sortie d'erreur standard

- **Redirection :**

- La redirection du flux de sortie d'erreur standard utilise les mêmes signes, mais précédés de l'identifiant du flux : 2.

```
invite/> ./programme 2> ls_erreur.txt
```

```
invite/> ./programme 2>> ls_erreur.txt
```


Rediriger l'entrée standard

- **Redirection :**

- Rediriger l'entrée standard permet d'entrer des données provenant d'un fichier au lieu du clavier.

```
invite/> cat < mon_fichier.txt
```

Rediriger un flux vers un autre

- **Redirection :**

- Il est possible de rediriger un flux vers la sortie standard ou la sortie d'erreur en donnant l'identifiant du flux précédé du caractère & à la place du nom de fichier.

```
invite/> ./programme 1>stdout_stderr.txt 2>&1
```

- Le fichier stdout_stderr.txt contient ce qui a été affiché à la fois sur le flux de sortie standard et le flux de sortie d'erreur.

Exemple : calculer la moyenne – Partie 1

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main(int argc, char** argv) {
7      bool loop = true;
8      float mean = 0.0;
9      float number = 0.0;
10
11     cout<<"Entrez vos notes : "<<endl;
12     while(loop) {
13         string xx;
14         cin >> xx;
15
16         if (xx != "exit") {
17             if (stof(xx) < 0.0) {
18                 cerr<<"Note inattendu !"<<endl;
19             }
20             else {
21                 mean += stof(xx);
22                 number += 1.0;
23             }
24         }
25         else {
26             loop = false;
27         }
28     }
29
30     cout<<"Votre moyenn est de : "<<mean/number<<endl;
31     return 0;
32 }
```

Listing 4 – "Exemple de programme en c++ : moyenne.cc"

Exemple : calculer la moyenne – Partie 2

```
invite/> ./moyenne
Entez vos notes :
11
12
-5
Note inattendu !!
13
-1
Votre moyenne : 12
```

Exemple : calculer la moyenne – Partie 3

```
invite/> cat note.txt
```

```
11
```

```
12
```

```
-5
```

```
13
```

```
-1
```

```
invite/> ./moyenne < note.txt
```

```
Entez vos notes :
```

```
11
```

```
12
```

```
-5
```

```
Note inattendu !!
```

```
13
```

```
-1
```

```
Votre moyenne : 12
```

Exemple : calculer la moyenne – Partie 4

```
invite/> ./moyenne < note.txt > stdout.txt  
Note inattendu !!
```

```
invite/> cat stdout.txt  
Entez vos notes :  
11  
12  
-5  
13  
-1  
Votre moyenne : 12
```

Exemple : calculer la moyenne – Partie 5

```
invite/> ./moyenne < note.txt > stdout.txt 2> stderr.txt
```

```
invite/> cat stderr.txt  
Note inattendu !!
```

Pipe : chaîne les commandes – Partie 1

Problématique : Comment afficher uniquement les prénoms triés par ordre alphabétique ?

```
invite/> cat note_etudiant.txt  
Emma;18  
Alice;12  
Jade;5  
Léna;19  
Camille;18
```

Solution 1 : Consiste à utiliser des fichiers temporaires

```
invite/> cut -f1 -d";" note_etudiant.txt > note_etudiant_cut.txt  
invite/> sort note_etudiant_cut.txt  
Alice  
Camille  
Emma  
Jade  
Léna  
invite/> rm note_etudiant_cut.txt
```


Pipe : chaîne les commandes – Partie 1

note_etudiant.txt → `cut -f1 -d";"` → note_etudiant_cut.txt

note_etudiant_cut.txt → `sort` → note_etudiant_sort.txt

note_etudiant.txt → `cut -f1 -d";"` → `sort` → note_etudiant_sort.txt

Pipe : chaîne les commandes – Partie 2

Solution 2 : Utiliser les pipe

```
invite/> cut -f1 -d";" note_etudiant.txt | sort  
Alice  
Camille  
Emma  
Jade  
Léna
```

Le pipe est un mécanisme qui permet de rediriger la sortie (stdout) d'un programme vers l'entrée (stdin) d'un autre programme

```
invite/> cat note_etudiant.txt | cut -f1 -d";" | sort  
Alice  
Camille  
Emma  
Jade  
Léna
```

Pipe : chaîne les commandes – Partie 3

Soit une matrice carré d'ordre 4 :

```
invite/> cat matrix.txt  
01;02;03;04  
04;05;06;07  
08;09;10;11  
12;13;14;15
```

Afficher la première ligne du fichier

```
invite/> cat matrix.txt | head -n 1
```

Afficher la dernière ligne du fichier

```
invite/> cat matrix.txt | tail -n 1
```

Afficher la troisième ligne du fichier

```
invite/> cat matrix.txt | head -n 3 | tail -n 1
```

Afficher l'élément $A_{3,3}$

```
invite/> cat matrix.txt | head -n 3 | tail -n 1 | cut -f3 -d";"
```

Pipe : exemples d'utilisation

Des exemples d'utilisation de pipe avec la commande history :

```
invite/> history | tail -n 3
invite/> history | grep red
invite/> history | grep red | grep blue
invite/> history | grep red | more
invite/> history | more
```

Des exemples d'utilisation de pipe avec la commande ls :

```
invite/> ls -lrth | tail -n 3
invite/> ls -lrth | head -n 5
invite/> ls -lrth | more
```

Des exemples d'utilisation de pipe avec des fichiers :

```
invite/> cat fichier.txt | tail -n 2 | head -n1
invite/> cat fichier.txt | cut -f4 -d";" | head -n 3 | tail -n 1
```

La commande **tr** permet de modifier le contenu de l'entrée standard :

Soit une matrice carré d'ordre 4 :

```
invite/> cat matrix.txt  
01;02;03;04  
04;05;06;07  
08;09;10;11  
12;13;14;15
```

Remplacer les ; par des - :

```
invite/> cat matrix.txt | tr ";" "-"  
01-02-03-04  
04-05-06-07  
08-09-10-11  
12-13-14-15
```

La commande **tr** permet de modifier le contenu de l'entrée standard :

Soit une matrice carré d'ordre 4 :

```
invite/> cat fichier.txt  
toto  
tata  
titi  
tutu
```

Remplacer les ; par des - :

```
invite/> cat fichier.txt | tr [:lower:] [:upper:]  
TOTO  
TATA  
TITI  
TUTU
```

Join

La commande **join** permet de fusionner les lignes de deux fichiers ayant un champ commun :

```
invite/> cat janvier2012  
Alimentation 50.00  
Eau 25.00  
Electricite 123.50  
Loyer 456.90  
Assurances 234.00
```

```
invite/> cat fevrier2012  
Alimentation 67.00  
Eau 34.00  
Electricite 156.00  
Loyer 456.90  
Assurances 225.00
```

La commande **join** permet de fusionner les lignes de deux fichiers ayant un champ commun :

```
invite/> join -1 1 -2 1 -d" " janvier2012 fevrier2012
Alimentation 50.00 67.00
Eau 25.00 34.00
Electricite 123.50 156.00
Loyer 456.90 456.90
Assurances 234.00 225.00
```


La commande **join** permet de fusionner les lignes de deux fichiers ayant un champ commun :

```
invite/> cat fevrier2012 | join -1 1 -2 1 -d" " janvier2012 -  
Alimentation 50.00 67.00  
Eau 25.00 34.00  
Electricite 123.50 156.00  
Loyer 456.90 456.90  
Assurances 234.00 225.00
```

Section 2

Compression des données



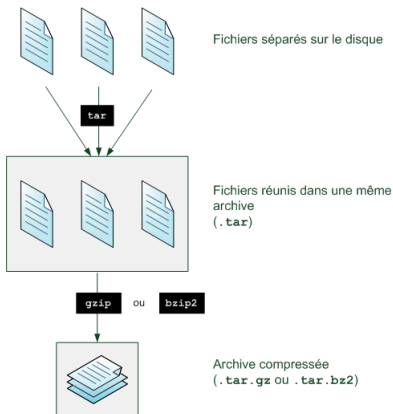
Compression

- **Définition** : compresser des données, c'est réduire la place occupée par cet ensemble de données.
 - **Compression sans perte** : Une compression est dite sans perte si les données après décompression sont identiques aux données originelles (exemple : texte)
 - **Compression avec perte** : Une compression est dite avec perte si les données après décompression sont différentes des données originelles (exemple : audio, image, vidéo)

Compression sans perte

- **Codage par répétition** : Toute suite de bits ou de caractères identiques est remplacée par un couple (nombre d'occurrences ; bit ou caractère répété).
- **Codage entropique** : Le codage entropique utilise des statistiques sur la source pour construire un code (ex. morse et la lettre e codé sur un point)
- **Codage par dictionnaire** : procèdent par la recherche de similitudes entre le texte à compresser et un ensemble de chaînes contenues dans une structure de données appelée "dictionnaire"

Compression sous Linux



- **Assembler** des fichiers dans une archive (programme tar)
- **Compresser** l'archive (programme gzip ou bzip2)

Créer une archive

Pour créer une archive :

```
invite/> tar -cvf nom_archive.tar nom_dossier
```

Options :

- **-c** : créer une archive
- **-v** : afficher le détail des opérations
- **-f** : l'archive du fichier

Créer une archive

On peut indiquer des dossiers ou des fichiers ou des fichiers et des dossiers :

```
invite/> tar -cvf nom_archive.tar nom_dossier
```

```
invite/> tar -cvf archive.tar fichier1 fichier2 fichier3
```

Afficher le contenu d'une archive sans l'extraire

Pour afficher le contenu d'une archive :

```
invite/> tar -tf tutoriels.tar  
file1  
file2  
file3
```

Options :

- **-t** : liste le contenu de l'archive
- **-f** : l'archive du fichier

Ajouter un fichier dans l'archive

Ajoute un fichier dans une archive :

```
invite/> tar -rvf tutoriels.tar fichier_supplementaire.tuto
```

Options :

- **-r** : les nouvelles entrées sont ajoutés dans l'archive
- **-v** : afficher le détail des opérations
- **-f** : l'archive du fichier

Extraire les fichiers de l'archive

Extraire les fichiers de l'archive :

```
invite/> tar -xvf tutoriels.tar
```

Options :

- **-x** : extraire les données de l'archive
- **-v** : afficher le détail des opérations
- **-f** : l'archive du fichier

Compresser et décompresser une archive

Compresser :

```
invite/> gzip tutoriels.tar  
invite/> bzip2 tutoriels.tar
```

Attention : gzip va créer un fichier tutoriels.tar.gz

Décompresser :

```
invite/> gunzip tutoriels.tar.gz  
invite/> bunzip2 tutoriels.tar.bz2
```

Archiver et compresser en même temps

Compresser :

```
invite/> tar -zcvf tutoriels.tar.gz tutoriels/
```

Décompresser :

```
invite/> tar -zxvf tutoriels.tar.gz
```