

Redirections et pipelines avec Linux

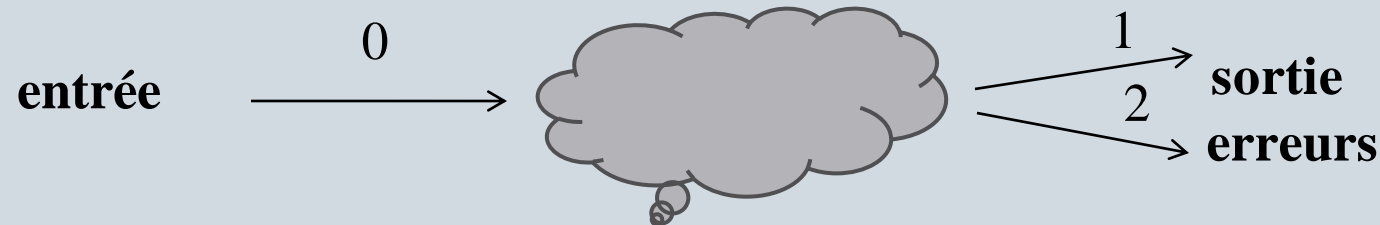


- ... Entrées-sorties de processus
- ... Service « d'assemblage » de commandes

Entrées/Sorties d'un Processus (1)

Les flux standards (stream)

- Un processus lit des données (standard input)
- Un processus écrit un résultat (standard output)
- Un processus écrit des erreurs (standard errors)

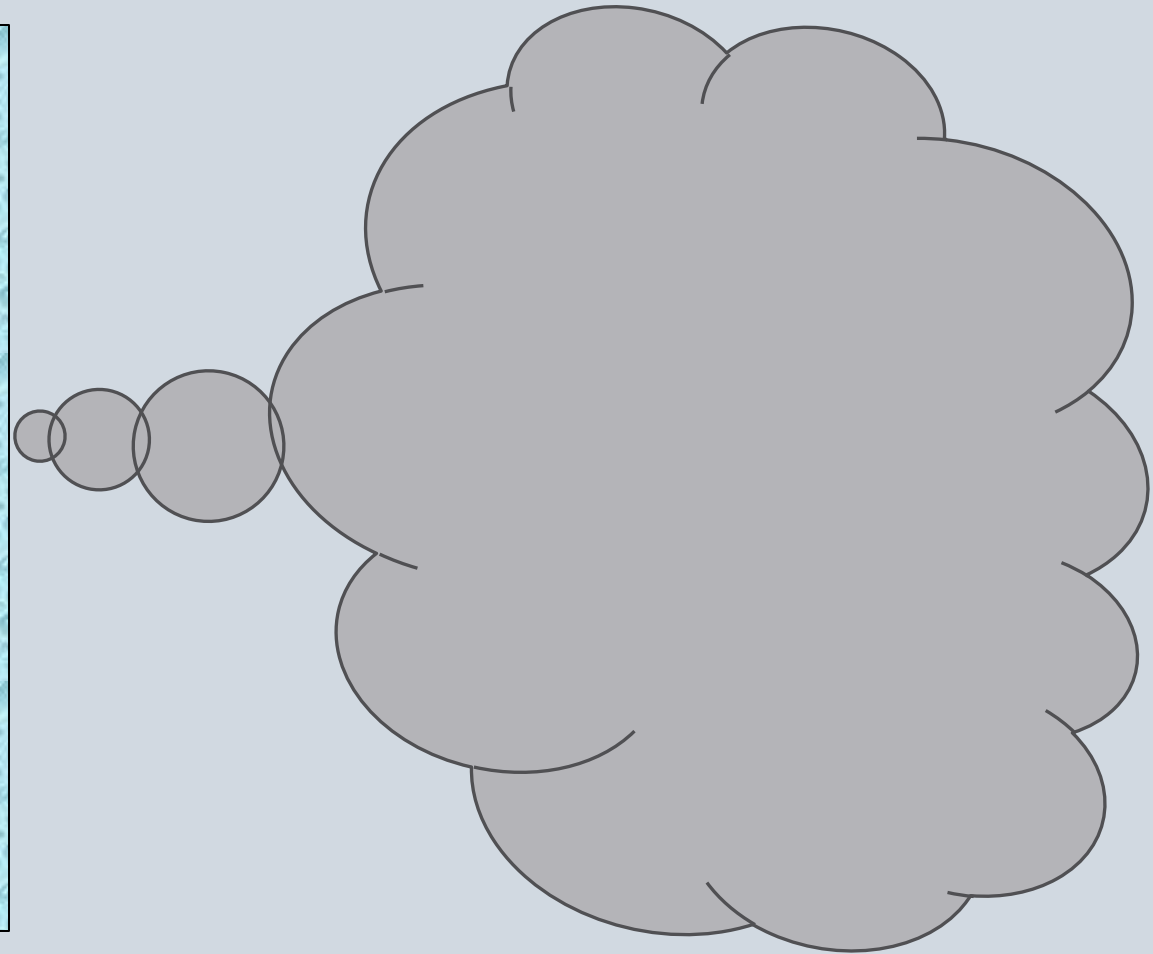


Connexion par défaut des flux aux organes d'entrée/sortie

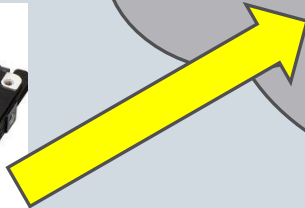
- *stdin* vers terminal (clavier)
- *stdout* vers terminal (affichage terminal)
- *stderr* vers terminal (affichage terminal)

Entrées/Sorties d'un Processus (2)

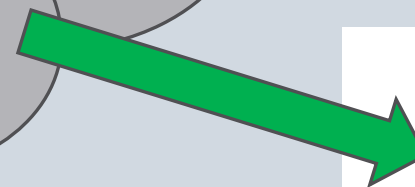
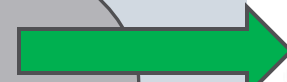
```
#include <stdio.h>
int main ( )           // mon_main.c
{
    fscanf (stdin, "%d", &nombre);
    if (nombre == 0)
        fprintf (stdout, "Tout va bien \n");
    else
        fprintf (stderr, "Tout va mal \n");
    return 0;
}
```



Entrées/Sorties d'un Processus (2)



```
#include <stdio.h>
int main ( )
{
    fscanf (stdin, "%d", &nombre);
    if (nombre == 0)
        fprintf (stdout, "Tout va bien \n");
    else
        fprintf (stderr, "Tout va mal \n");
    return 0;
}
```



>< | >>

> redirige le flux de sortie de la commande pour la placer dans un fichier. Par défaut, si rien n'est précisé, le flux redirigé est la sortie standard, i.e. > est équivalent à 1>. Pour rediriger la sortie d'erreur standard, on utilise 2>.

< redirige le flux d'entrée de la commande pour la prendre dans un fichier,

| redirige la sortie standard de la commande de gauche sur l'entrée standard de la commande de droite,

>> redirige le flux de sortie de la commande pour l'ajouter à la fin d'un fichier existant.

Redirections (1)

Exemples d'exécution

Prompt \$ mon_main < Jeu-essai1.txt

Prompt \$ mon_main < Jeu-essai1.txt > resultat1.txt

Prompt \$ mon_main < Jeu-essai1.txt > resultat1.txt 2> erreur1.txt

Prompt \$ mon_main >> resultat1.txt

Redirections (2)

Exemples d'exécution de commandes standards

Prompt \$ ls > trace

Prompt \$ chmod 755 * 2>trace-erreurs

Prompt \$ mail chignoli@unice.fr < M111_TD2_Dupond.txt

Prompt \$ cat < /etc/passwd

... les caractères du fichier ...

^D

Prompt \$

Redirections (3)

Exemples d'exécution

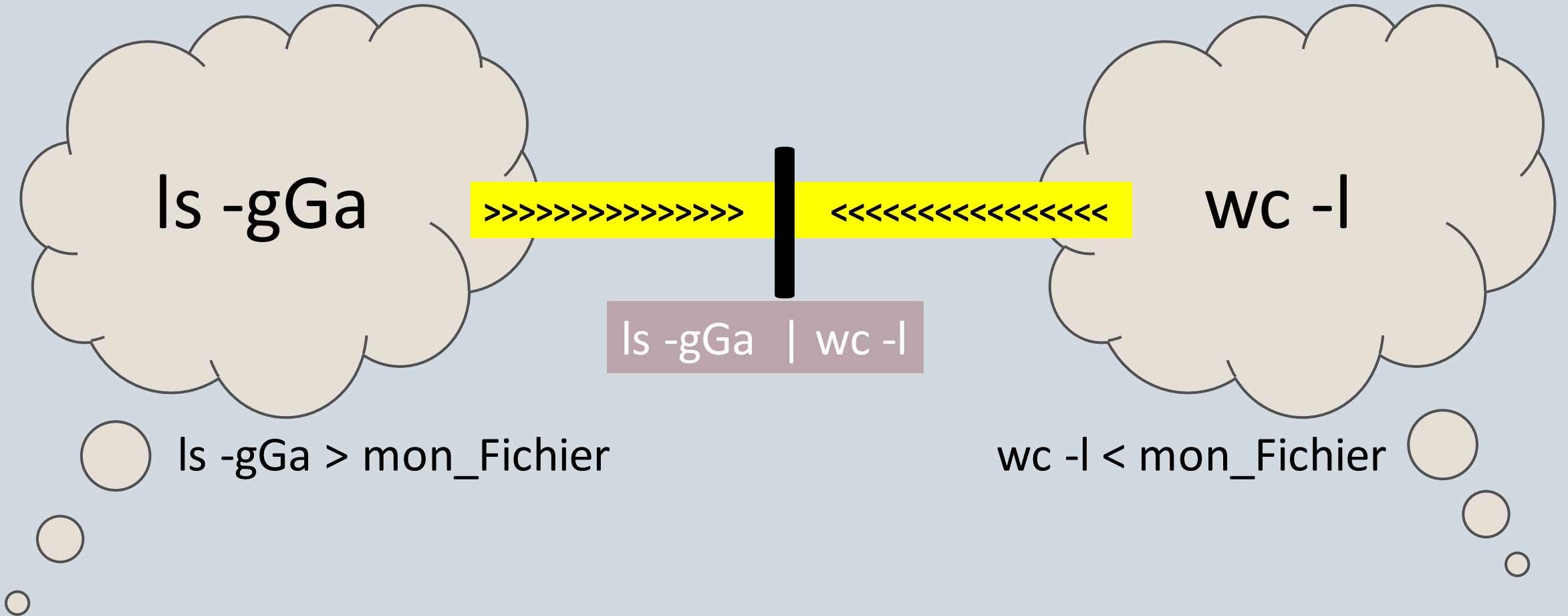
Prompt \$ mon_autre_main 3< Jeu-essai1.txt

Prompt \$ mon_autre_main 3< Jeu-essai1.txt 4> resultat1.txt

Prompt \$ mon_autre_main 4>> resultat1.txt

Pipeline (1)

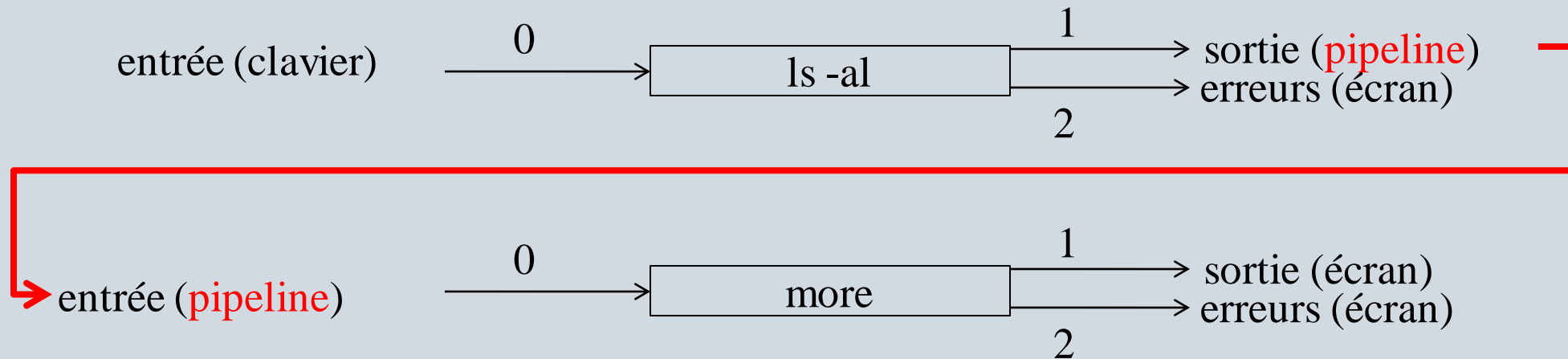
Ou tube ... ou aboutement ...



Pipeline (2)

Exemple :

Prompt \$ **ls -al | more**



`ls -al 2> /dev/null | more`

redirige les erreurs du processus ls vers null (trou noir)

`ls -al | more 2> /dev/null`

redirige les erreurs du processus more vers null

Exemples

```
cat *.txt | sort | uniq > fichier-resultat
```

Trie la sortie de tous les fichiers .txt et supprime les lignes # dupliquées, pour finalement enregistrer les résultats dans # « fichier-résultat ».

```
cat < toto > titi 2> tutu
```

Prend en entrée le fichier toto, met le résultat

Sur le fichier titi et la sortie erreur sur tutu

```
vagrant@vagrant:~/Documents$ cat > titi
J'écris ces lignes dans le fichier titi à l'aide de la commande cat.
Le fichier est ainsi créé.
^C
vagrant@vagrant:~/Documents$ cat titi
J'écris ces lignes dans le fichier titi à l'aide de la commande cat.
Le fichier est ainsi créé.
vagrant@vagrant:~/Documents$ cat << FIN > titi
> cat va écrire dans le fichier titi et écraser ce qu'il y avait avant
> Jusqu'à ce que j'écrive le mot
> FIN
vagrant@vagrant:~/Documents$ cat titi
cat va écrire dans le fichier titi et écraser ce qu'il y avait avant
Jusqu'à ce que j'écrive le mot
vagrant@vagrant:~/Documents$ █
```