


Chapitre 4 : Redirections

Pour changer la provenance des lectures et la destination des affichages d'une commande

The Debian logo, which is a stylized spiral, is positioned behind the text. Below the spiral, the word "Debian" is written in a simple, sans-serif font.

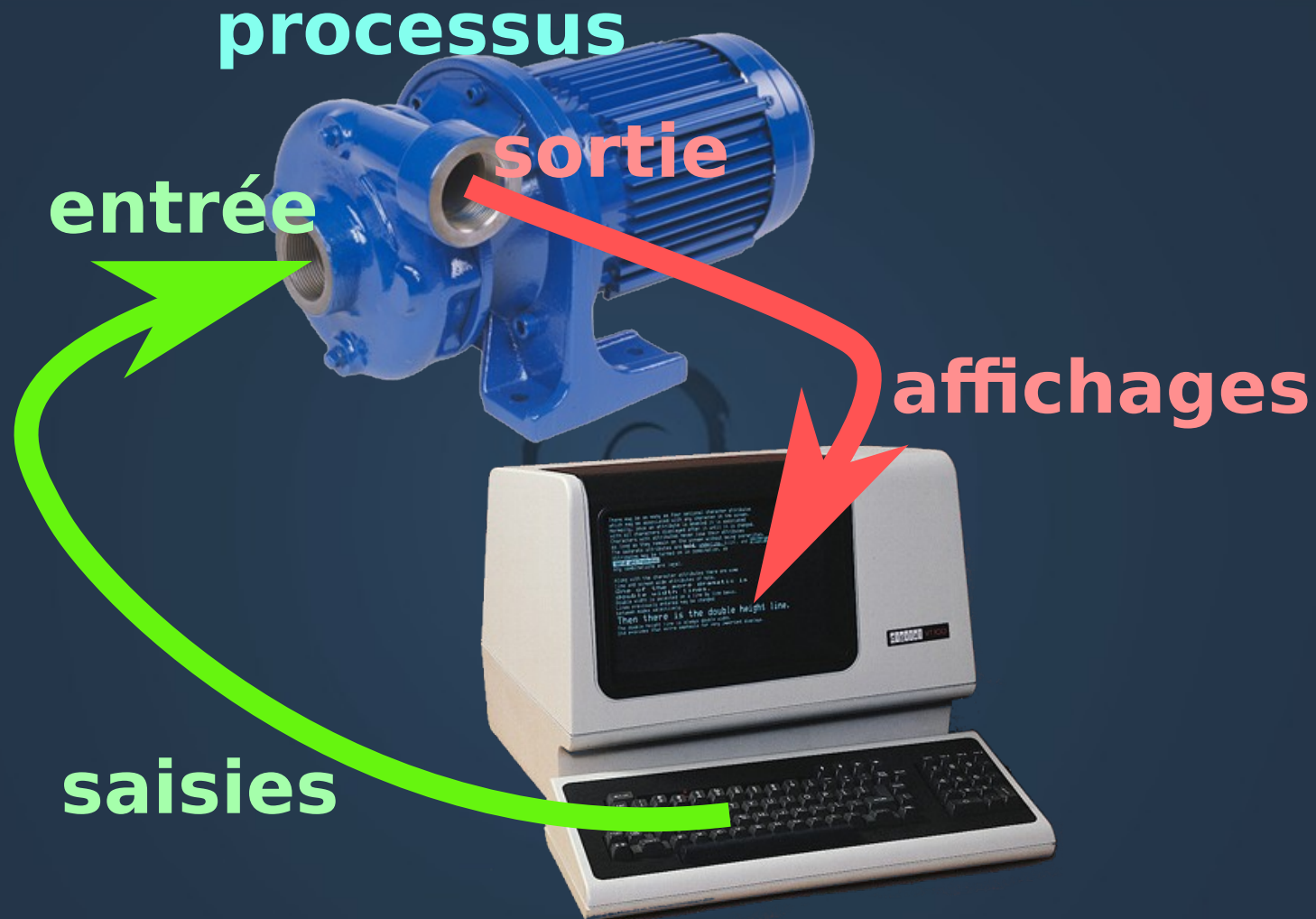
Debian

4.1 – Entrée et sorties d'un processus

Entrée = scanf, gets \leftarrow vient du clavier
Sortie = printf, putchar \Rightarrow va à l'écran

Debian

Entrée et sortie d'un processus



Que se passe-t-il ?

- scanf , gets et getchar = entrée de données dans le programme ⇒ lecture du clavier
- printf, puts et putchar = sortie de données du programme vers l'extérieur ⇒ écriture de l'écran
- NB : les données sont des caractères, des lignes de texte

Debian

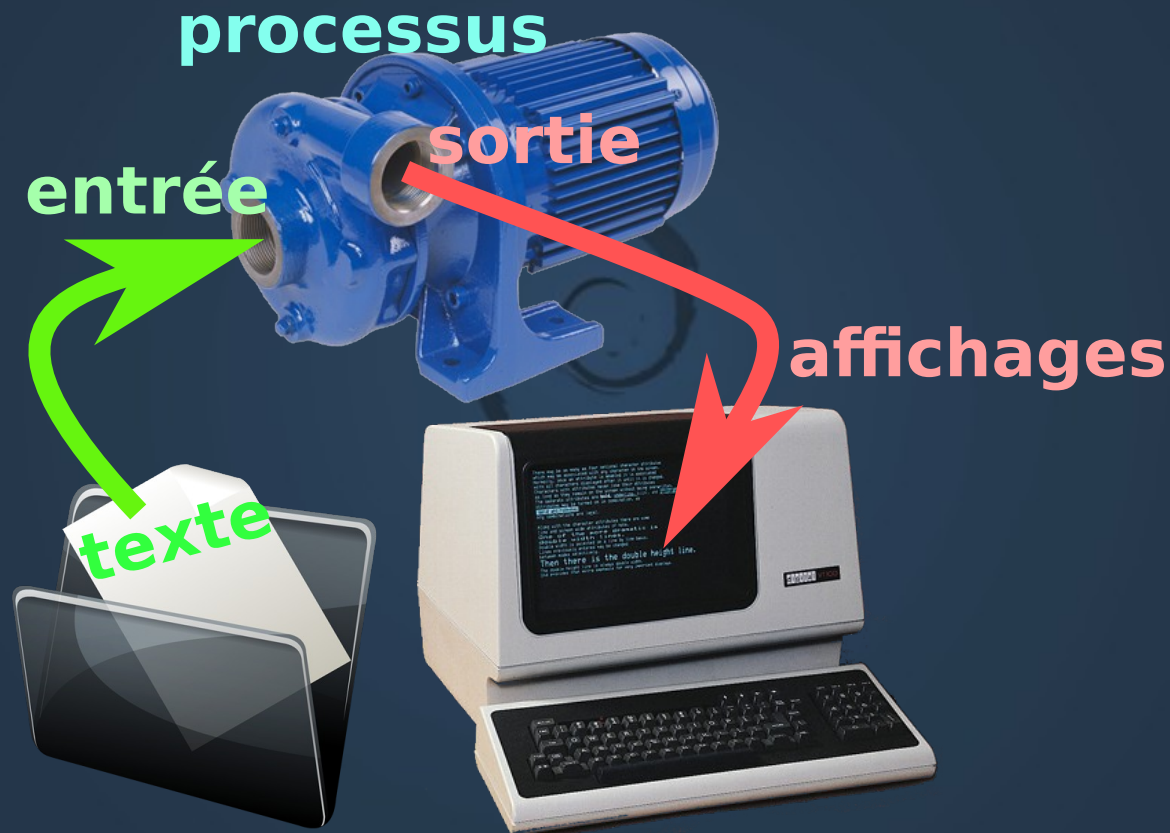
Exemple

```
#include <stdio.h>

int main()
{
    char nom[41];
    printf("À qui ai-je l'honneur ? "); // sortie
    scanf("%s", nom); // entrée
    printf("Salutations %s\n", nom); // sortie
}
```

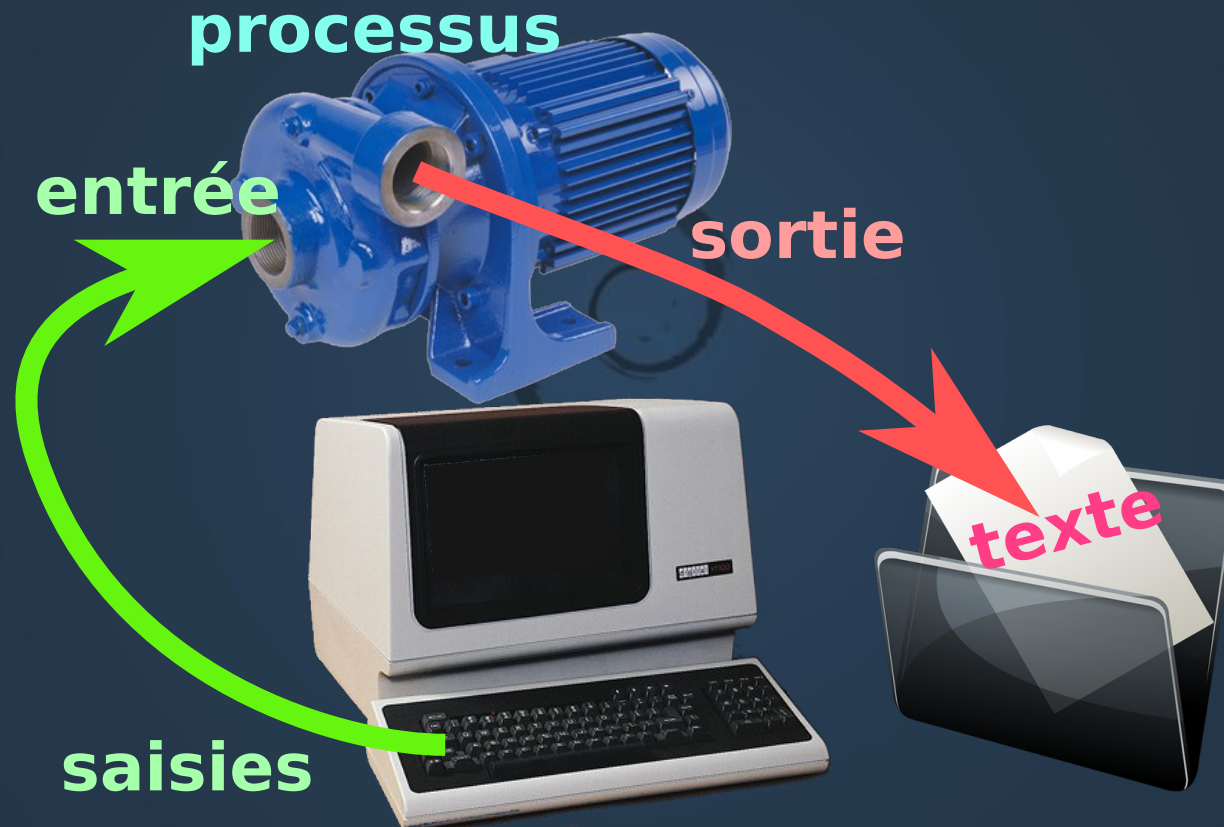
Redirection de l'entrée

- On peut changer la provenance des données



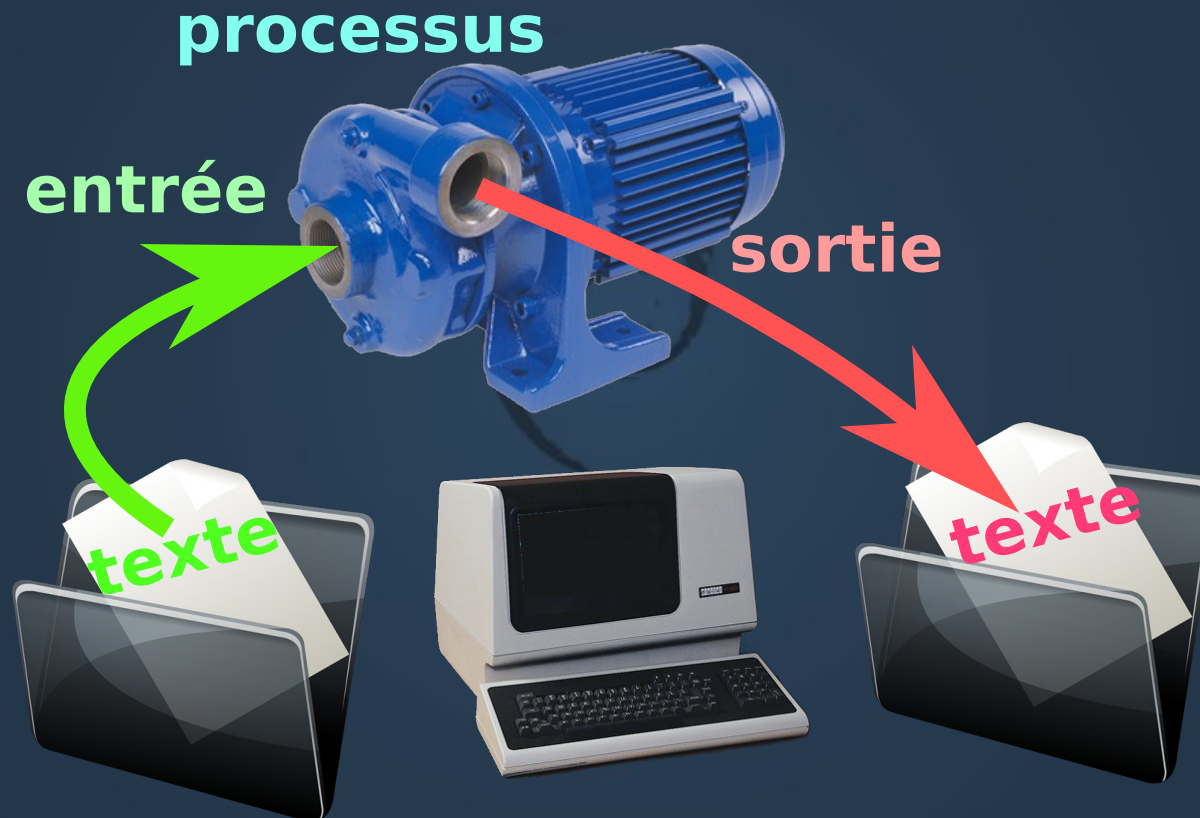
Redirection de la sortie

- On peut changer la destination des affichages



Redirection des deux

- Changer l'entrée et changer la sortie :



Vocabulaire

- L'entrée s'appelle **stdin** : entrée standard, (standard input)
 - Elle est affectée au clavier *par défaut*
- La sortie s'appelle **stdout** : sortie standard (output)
 - Elle est affectée *par défaut* à l'écran
- *Par défaut* = en absence d'une directive lors de la saisie de la commande

Directives de redirection

- Pour changer l'entrée :

commande < fichier

- Pour changer la sortie, deux manières :

commande > fichier crée ou écrase le fichier

commande >> fichier crée ou complète le fichier

- Pour changer les deux :

commande < fichierE > fichiers

commande < fichierE >> fichiers

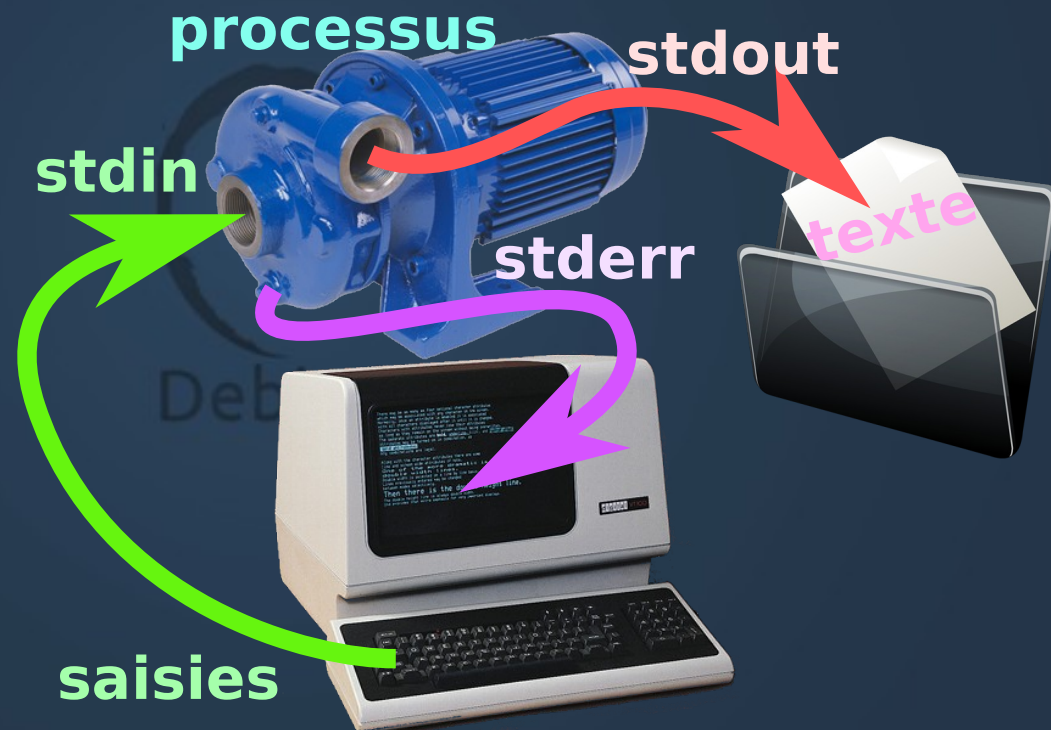
Exemples

- `ls > listing`
- `wc -l < listing`
- `shuf < listing > listing_pas_trié`
- `date >> listing_pas_trié`
- `tail -n 1 < listing_pas_trié`

Debian

stdin, stdout et stderr

- En réalité, les commandes ont une entrée (**stdin**) et deux sorties (**stdout** et **stderr**), la seconde sert pour les messages d'informations ou d'erreurs.



Exemple en C

```
#include <stdio.h>

int main()
{
    printf("Hello\n");           // → stdout
    fprintf(stderr, "Salut\n");  // → stderr
    fprintf(stdout, "Ciao\n");   // → stdout
}
```

Directive pour rediriger stderr

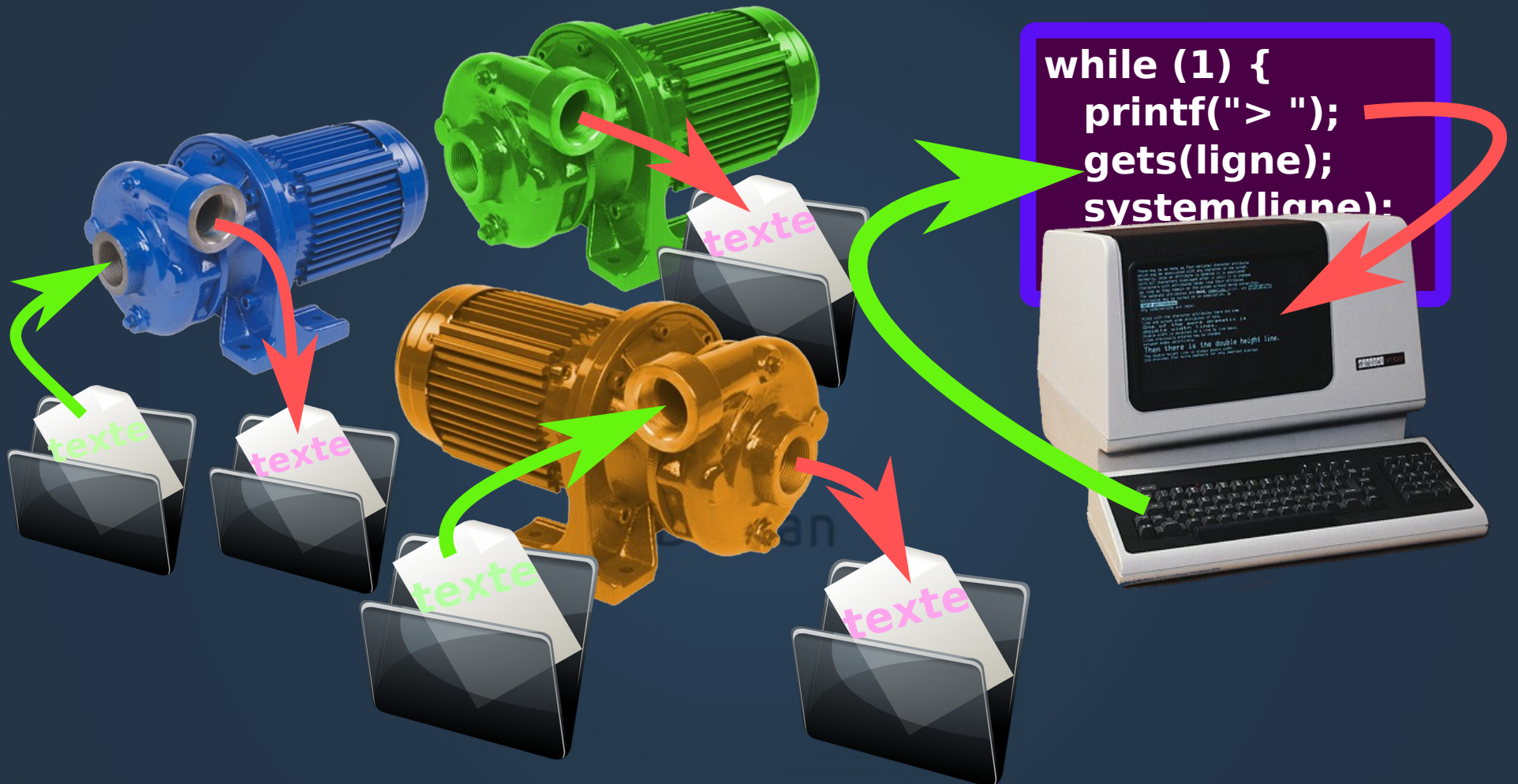
- En général, on ne redirige pas stderr afin de toujours voir les messages d'erreur sur la console
- Pour rediriger stdout et stderr ensemble vers un même fichier :

commande &> fichier

NB: ne pas confondre avec le & de la mise en arrière-plan

Retour sur les jobs

Jobs avec E/S redirigées + Shell



Redirection et jobs

- Pour lancer un job avec redirection :

commande < **fichierE** > **fichiers** &


De cette manière, le processus ne sera pas bloqué sur une lecture de stdin et ne polluera pas l'écran avec ses affichages

- On peut en plus rediriger stderr :

commande < **fichierE** &> **fichiers** &

4.2 – Tubes entre processus

Pour coupler deux commandes

The Debian logo, which is a stylized spiral, is positioned behind the text 'Pour coupler deux commandes'.

Debian

Idée

- On voudrait compter les processus qui tournent actuellement sur le système :
 - commande `ps -edf` donne la liste
 - commande `wc -l` pour compter les lignes d'une liste
- Idée n°1 :
 - `ps -edf > liste`
 - `wc -l < liste`
 - `rm liste`

Ça fait passer par un **fichier intermédiaire...**

Meilleure solution : un tube

processus 1

processus 2

TUBE

affichages

saisies



Explication

- La sortie du premier processus n'est plus redirigée vers un fichier, mais vers l'entrée du 2e processus, sorte de composition de fonctions
résultat = travail2(travail1(entrée))
- **Un tube est une connexion sans fichier intermédiaire, entre deux processus**
 - Extrêmement efficace en interne
- Ici, pour compter les processus, il faut coupler `ps -edf` avec `wc -l`

Directive pour créer un tube

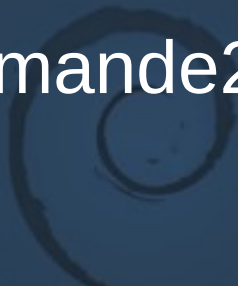
- Pour créer un tube :

commande1 | *commande2*

Remarque : la commande2 est appelée « **filtre** »

- Exemples :

ps -edf | wc -l

The Debian logo, a stylized spiral, is positioned behind the word "Debian".

Debian

Quelques tubes utiles

- Améliorer l'affichage des commandes :
 - `ls -l | more`
 - `ps -edf | more`
- Faire des petits traitements de données :
 - `ps -edf | egrep sonPID`
 - Affiche le processus ayant ce PID
 - `ps -edf | sort`
 - Trie la liste des processus sur le nom (au lieu du PID)
- Faire des gros traitements de données...

Tubes complexes

- On peut composer une **commande complexe**, avec **plusieurs tubes et des redirections**
 - *cmde1 | cmde2 | cmde3 | cmde4...*
 - *cmde1 < fichier1 | cmde2 | cmde3 > fichier2*
 - **Attention, les redirections ne peuvent être que sur la première et la dernière commande ; ces deux exemples sont incorrects :**
 - *commande1 | commande2 < fichier*
 - *commande1 > fichier | commande2*

Exemples d'application des tubes

- Fichiers de données, ex : résultats MotoGP
 - Afficher le 1er coureur français

```
cat MotoGP2012 | grep :FR: | head -n 1
```

- Suite logicielle PPM

- Pivoter et redimensionner une image :

```
anytopnm Tourner.jpg | pnmrotate -90 |  
pnmscale 0.25 | display
```