

# Qqs cdes – fichiers texte

`$head [options] [fichier]`

ex: `$head -5 fichier`: Affiche les 5 premières lignes

`$tail [options] [fichier]`

`$tail -5 fichier`

`$tail +5 fichier`

`$tail -f fichier`

`$wc [-l] [-c] [fichier]`

Compte les lignes, caracteres

Important: si pas de fichier en argument, les Cdes opèrent sur ce qui est saisi au clavier.

# Qqs commandes – grep

```
$grep [options] regexp [fic(s)]
```

regexp: expression rationnelle, pattern, motif.

```
$grep toto fic1
```

Affiche les lignes de fic1 contenant la chaîne “toto”

```
$grep ^toto fic1 fic2
```

Affiche les lignes de fic1 et fic2 qui commencent par la chaîne toto.

```
$grep toto$ *.c
```

Affiche les lignes de ts les fichiers .c qui se terminent par toto

# Qqs commandes – grep

## Options :

- v: tout sauf
- r: recursivement
- i: case insensitive
- n: affiche num de ligne
- l: nom fichier
- ....

# Qqs commandes – grep

## Expressions rationnelles:

`^`, `$`

`[A-Z]`, `[d-f]`, `[3-8]`

`[aeiouy]`

`^[A-Z]` est différent de `^[A-Z]`

`[:alpha:]`  $\Leftrightarrow$  `A-Za-z`

`[[:alpha:]]`  $\Leftrightarrow$  `[A-Za-z]`

etc ...

# Qqs commandes – cut

Afficher certaines “colonnes”

On parle plutôt de champs et de séparateurs

(**esp** , ; \$ % **x** ...)

`totoxtutuxtixtata`

Si **x** est le séparateur `toto tutu titi tata`

Si **u** est le séparateur `totoxt t xtitixtata`

**\$cut [opts] [fic(s)]**

**-d** delimitateur (séparateur)

**-f** ch1, ch2

**-f** ch1-chn (**-f** ch1- ...)

**-c** idem pour les caractères

`$cut -f1,3 -d: /etc/passwd`

# Qqs commandes – sort

Trier selon certains champs: `$sort [opts]`

`$sort [opts] -k champ -t del [fic(s)]`

`-n` : tri numerique et non alphanumerique

`-r`: reverse

`$sort -n -k 3 -t: /etc/passwd`

# Stdin, stdout, stderr

Chaque commande “communique” au travers de 3 flux:  
**stdin, stdout, stderr**

**Stdout**: Sortie standard.

Où s'affiche le résultat d'une commande: en général sur le terminal.

**Stderr**: Sortie d'erreurs

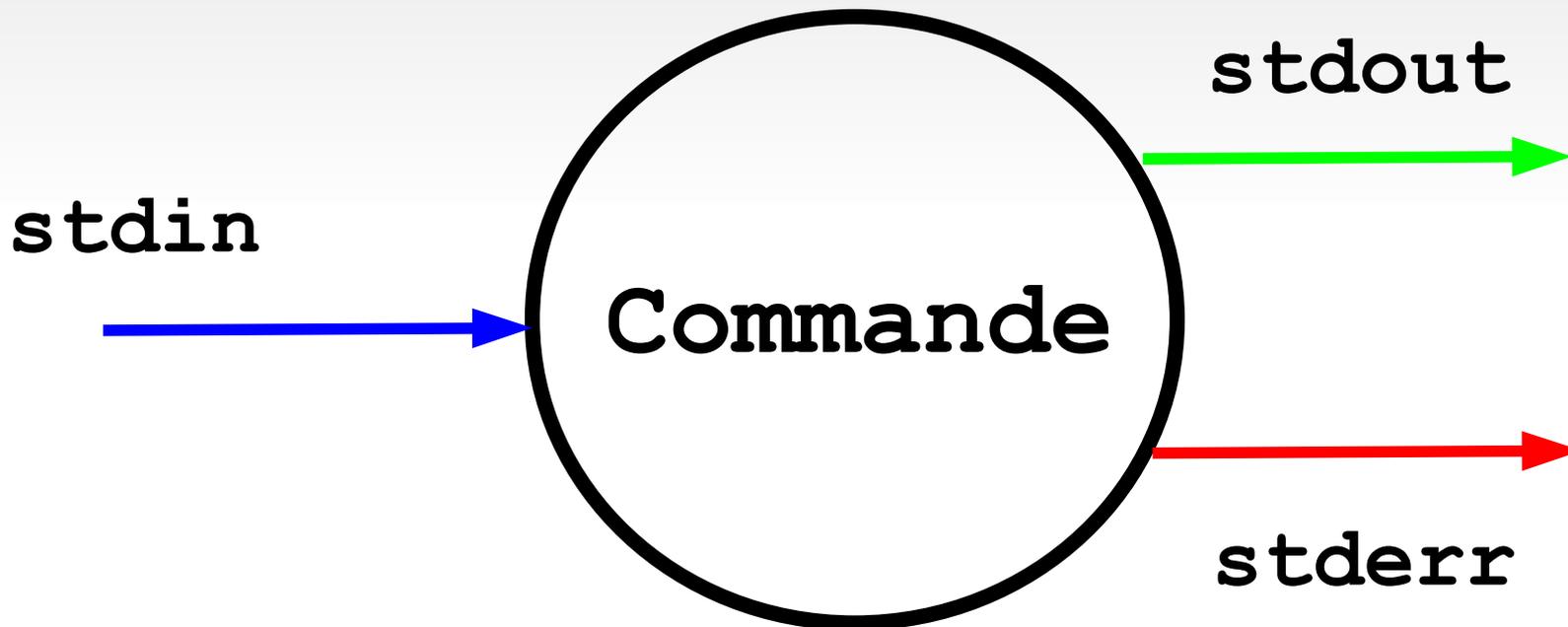
Où s'affichent les erreurs d'une commande: en général sur le terminal aussi.

**Stdin**: Entrée standard.

En général le clavier.

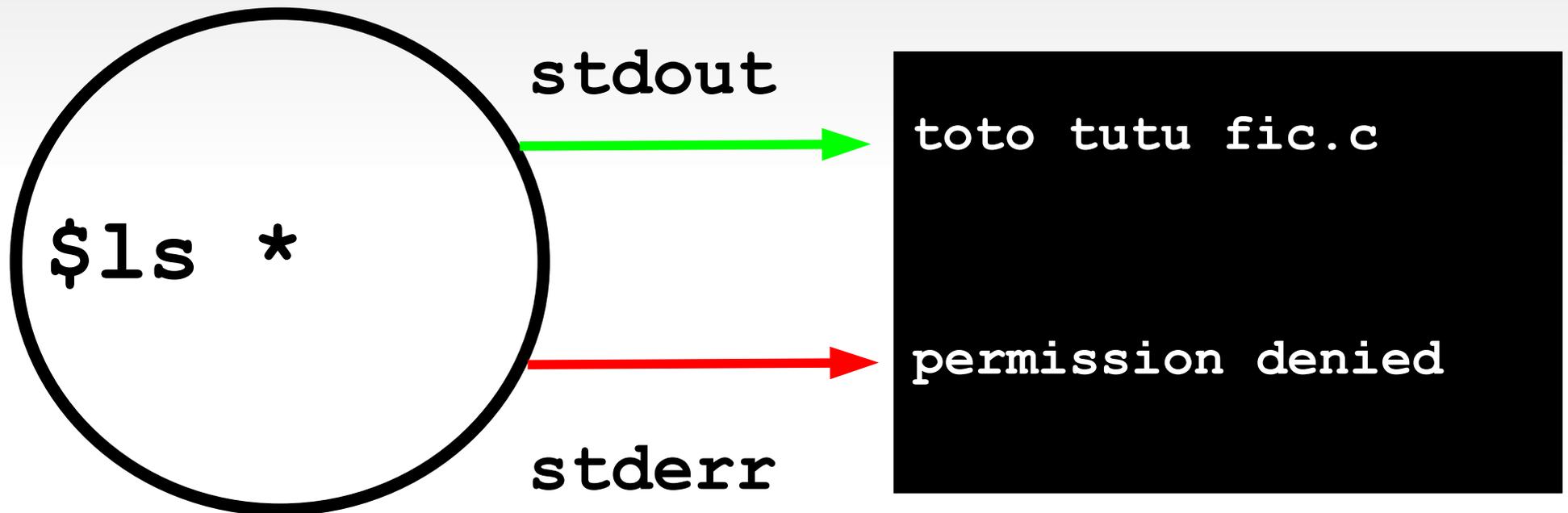
# stdin, stdout, stderr

Représentation



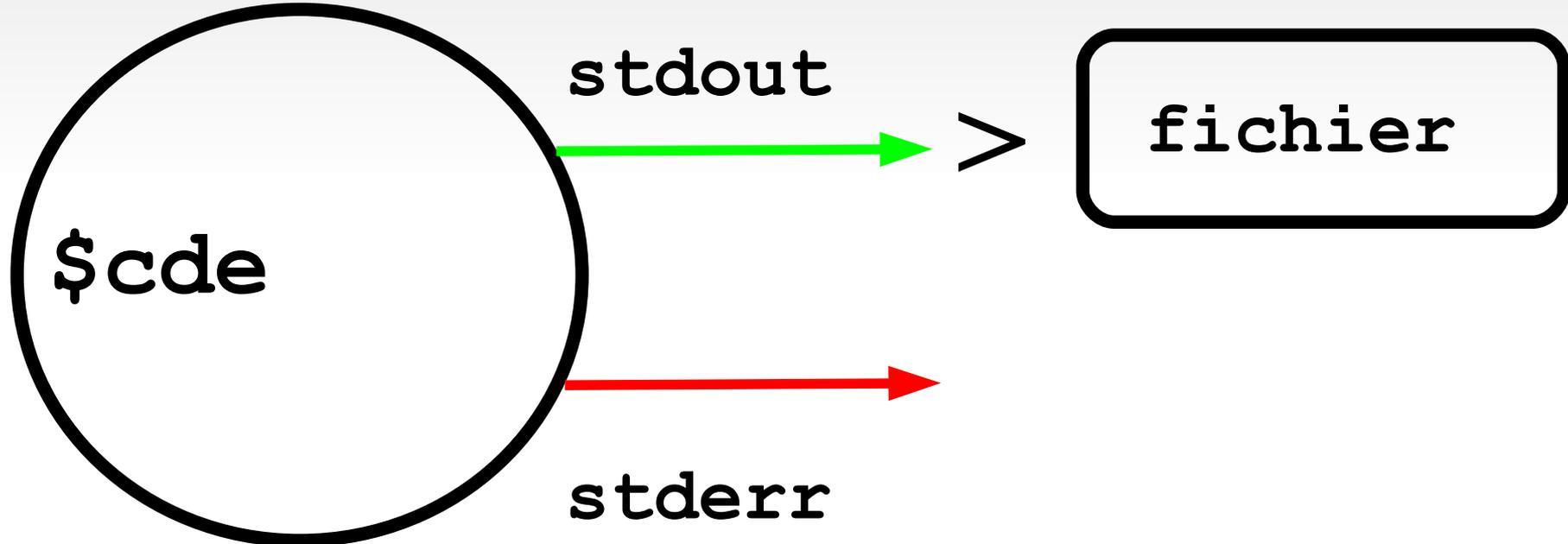
# stdout, stderr

`$cde`



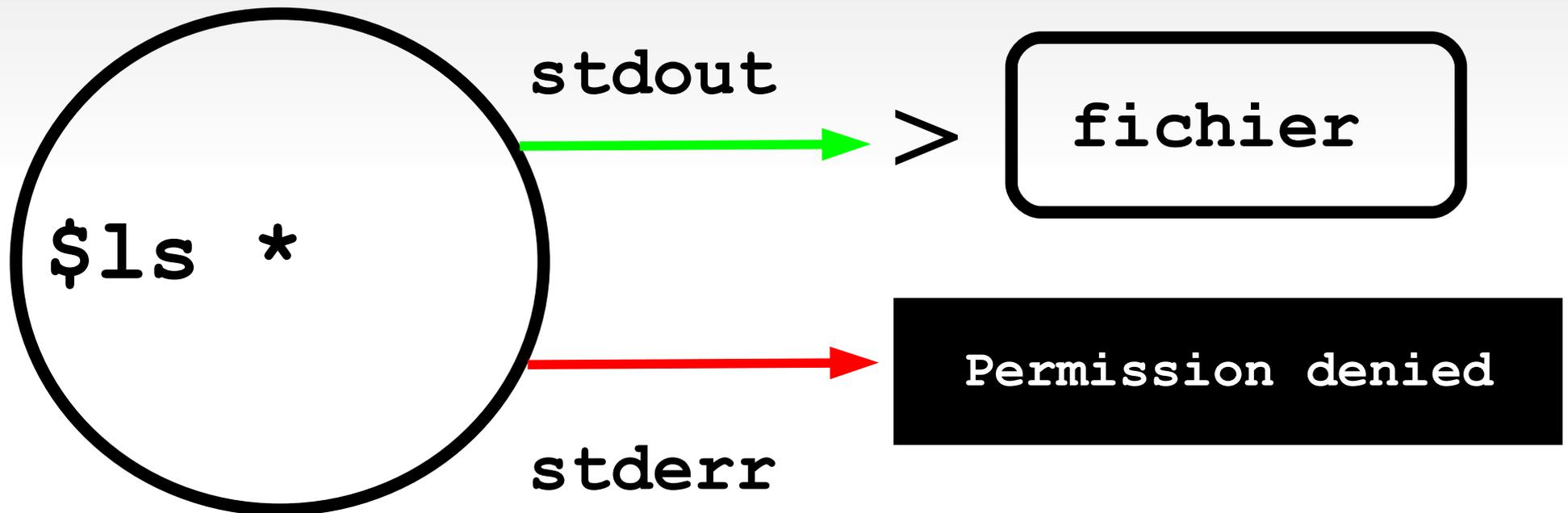
# redirection stdout

```
$ cde > fichier
```



# redirection stdout

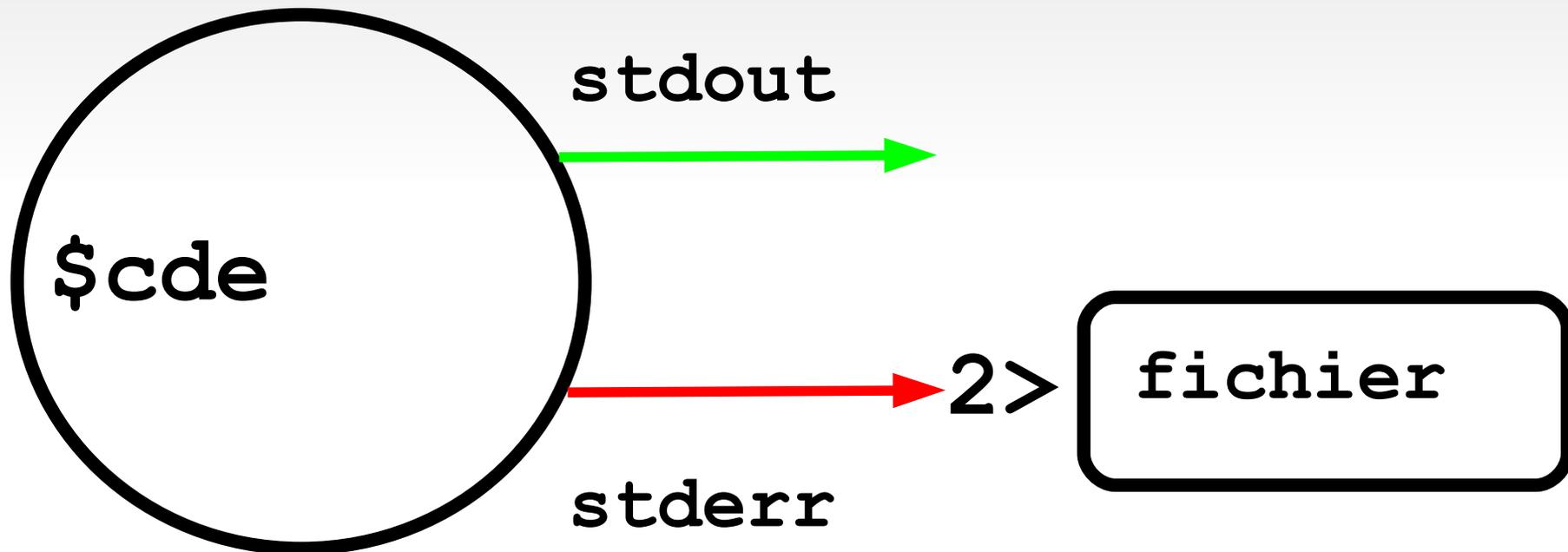
```
$ ls * > fichier
```



Remarque: `>>` concatène la sortie standard à un fichier existant

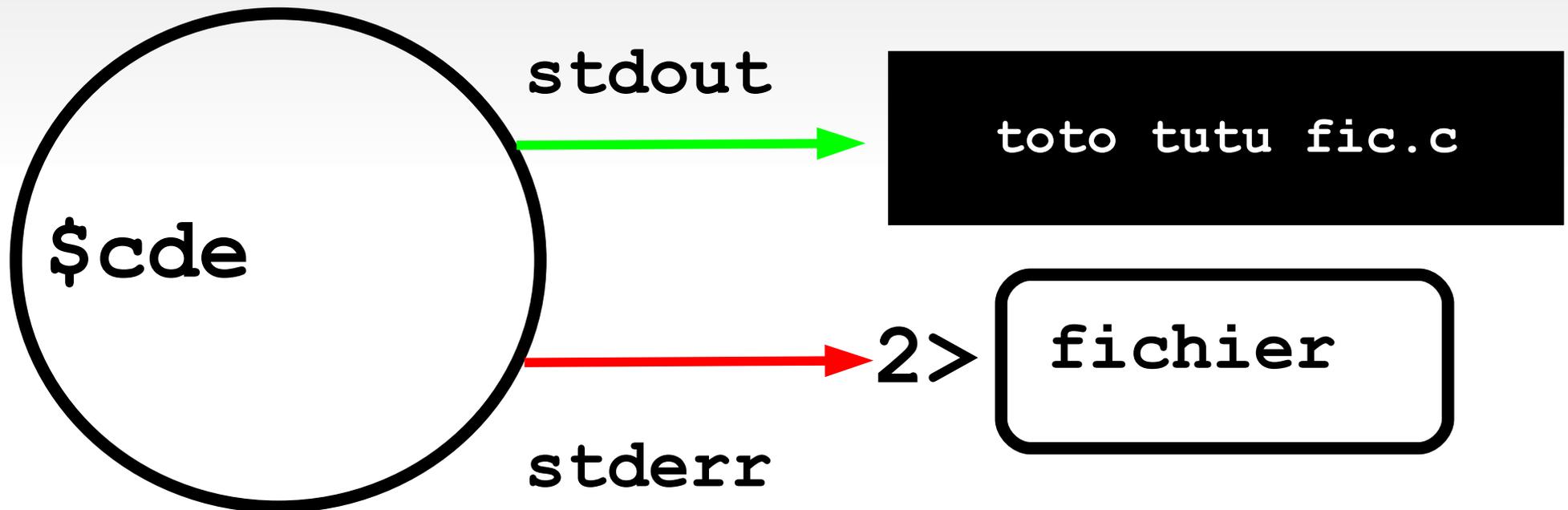
# redirection stderr

```
$ cde 2 > fichier
```



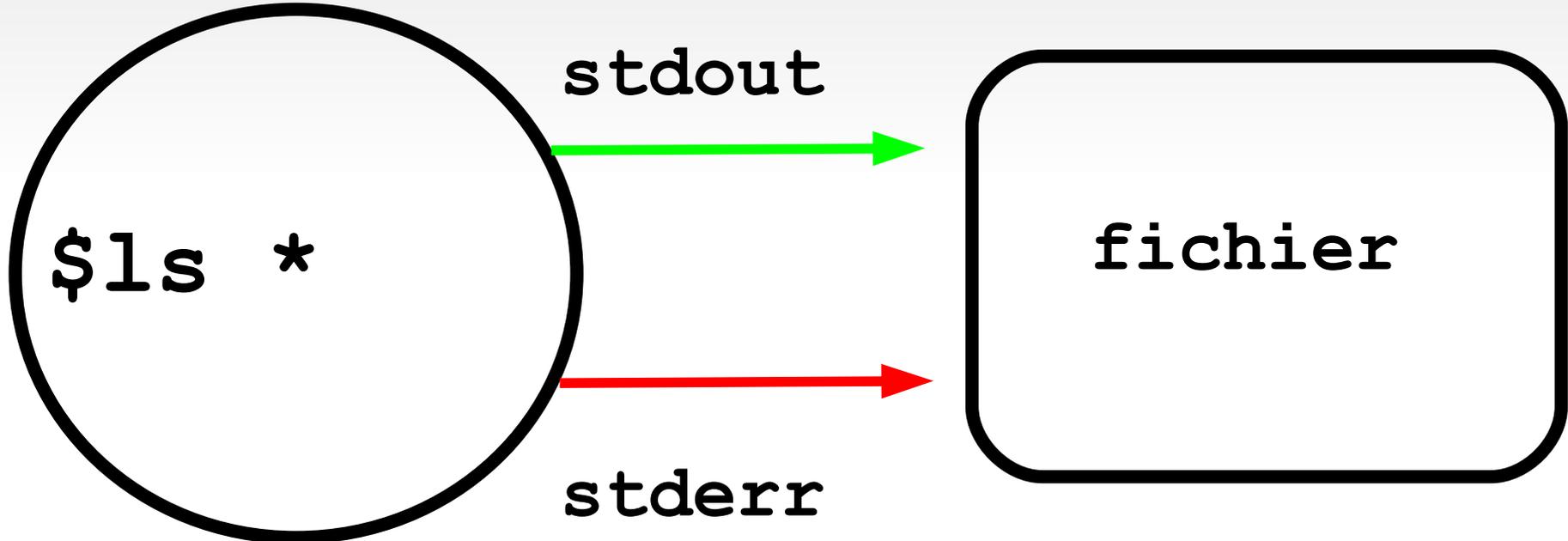
# redirection stderr

```
$ cde 2 > fichier
```



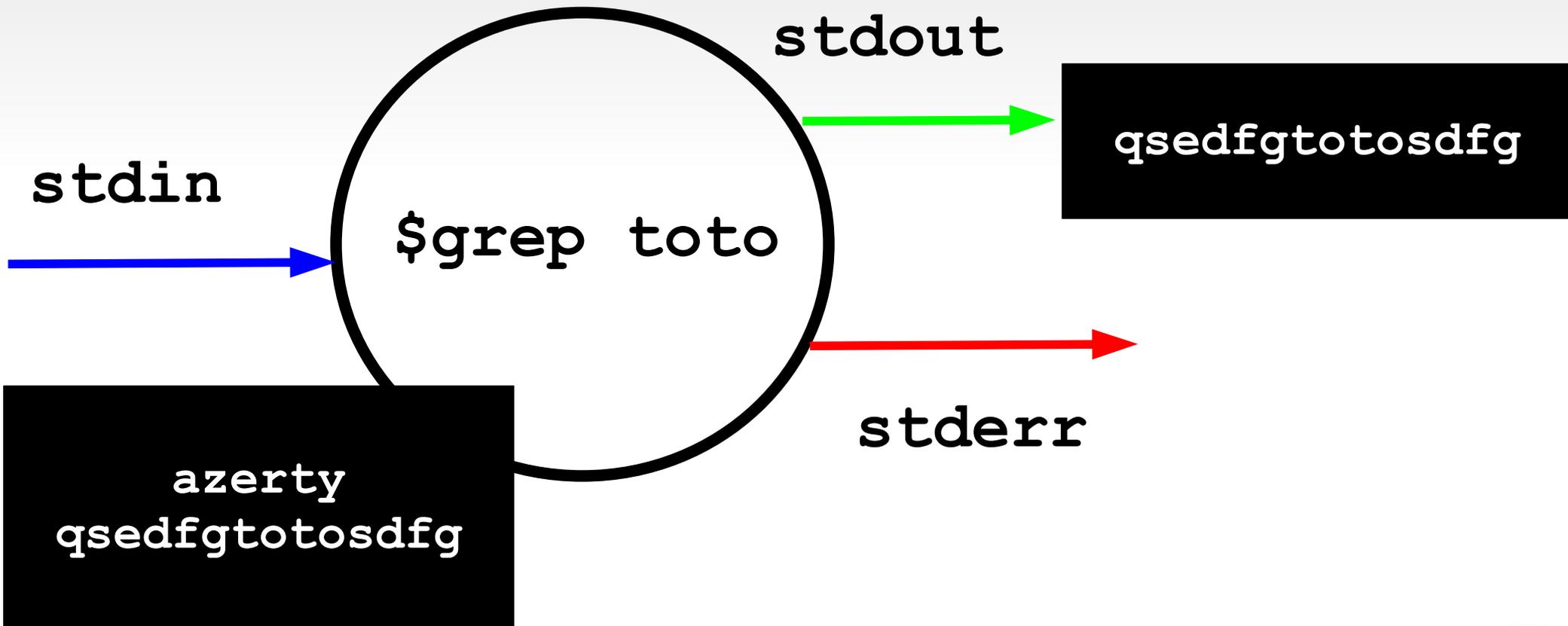
# redirection stdout

```
$ cde > fichier 2>&1
```



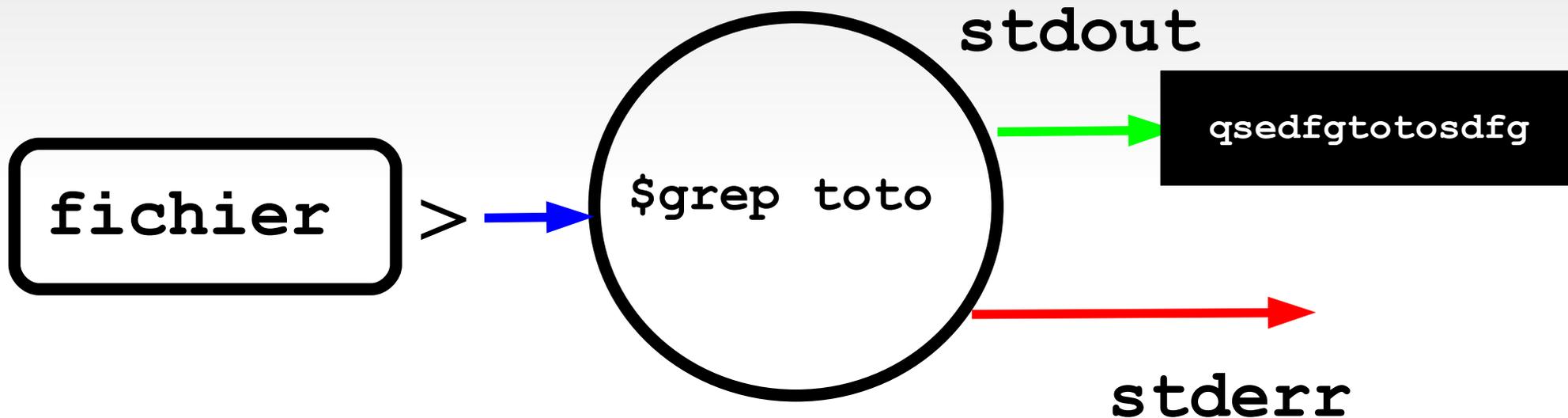
# stdin

\$cde



# stdin

```
$cde < fichier
```



Remarque: dans ce cas précis de commande

```
$grep toto < fichier <=> $grep toto fichier
```

# Qqs exemples:

```
$cat /etc/passwd
```

```
root:x:0:0:root:/root:/bin/zsh
```

```
dgeo:x:10102:13001:Desvernay Geoffroy:/users/info/dgeo:/bin/zsh
```

```
vajasse:x:30000:30001:ajasse virginie:/users/promo2006/vajasse:/bin/bash
```

```
rfortrie:x:20142:20001:Fortrie Remy:/users/prof/rfortrie:/bin/bash
```

```
mhamidou:x:20144:20001:Hamidou Mohammed:/users/prof/mhamidou:/bin/bash
```

```
...
```

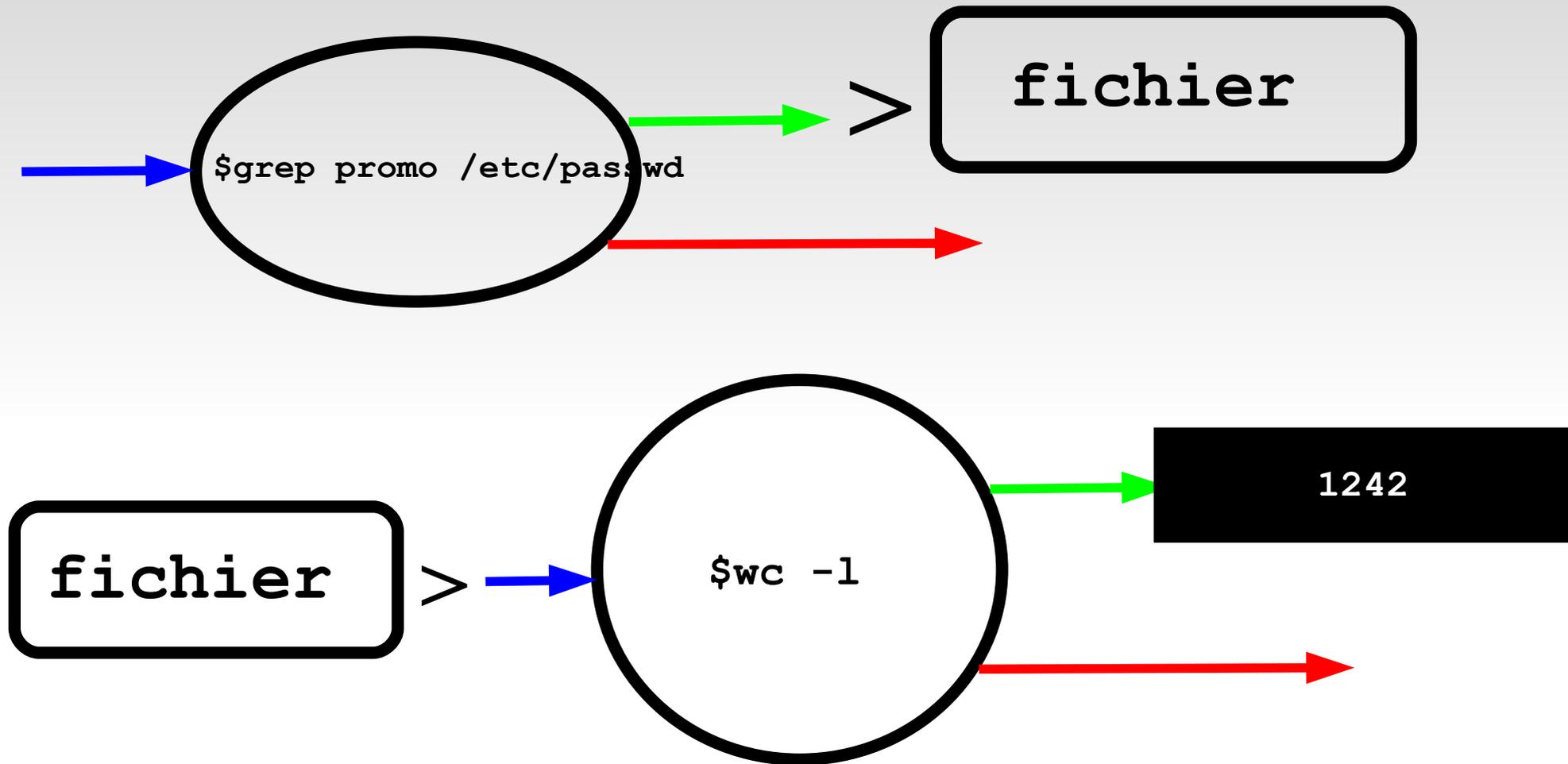
```
$grep promo /etc/passwd > fichier
```

```
$wc -l fichier
```

```
1242
```

```
Rem: $wc -l < fichier
```

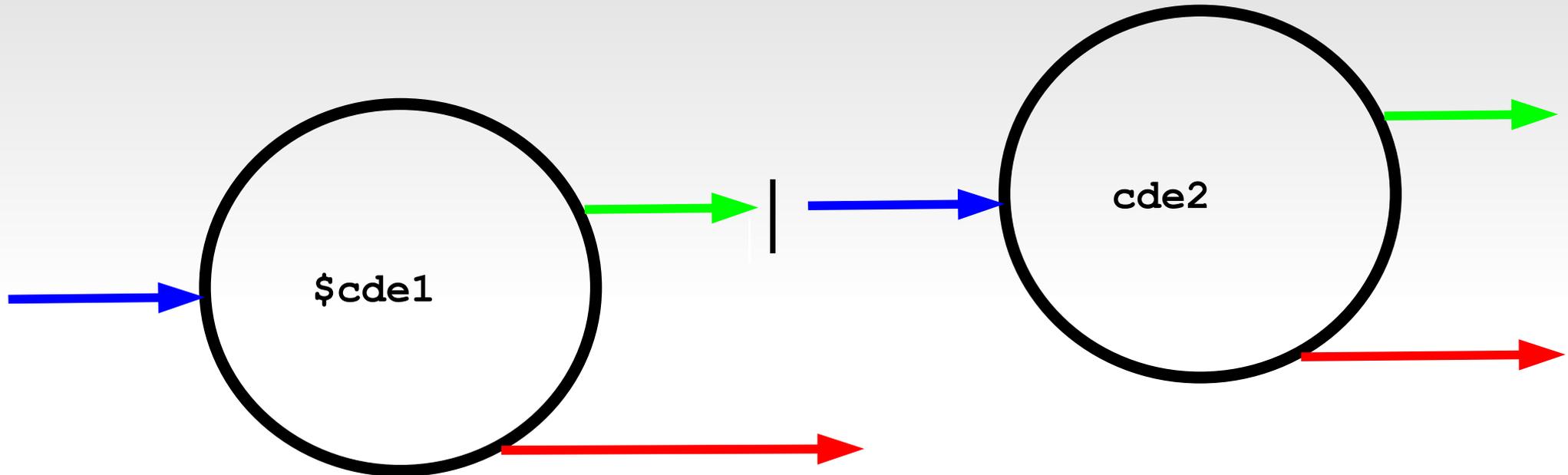
# Représentation



Remarque: Peut être peut on “brancher” directement la sortie standard de la première commande sur l'entrée standard de la seconde ?

# Filtres & Pipes

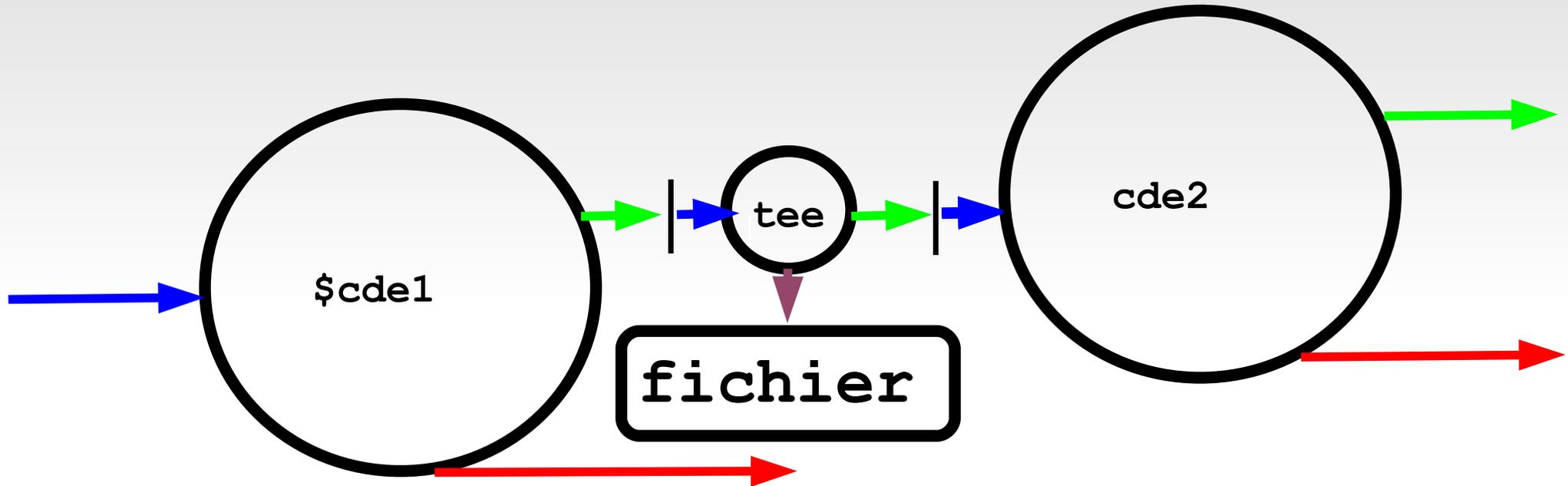
```
$cde1 | cde2
```



```
$grep promo /etc/passwd | wc -l
```

# Filtres & Pipes & derivations

```
$cde1 | tee fichier | cde2
```



```
$grep promo /etc/passwd | tee fichier | wc -l
```