

# Automatiser des procédures à l'aide de scripts

Module 122

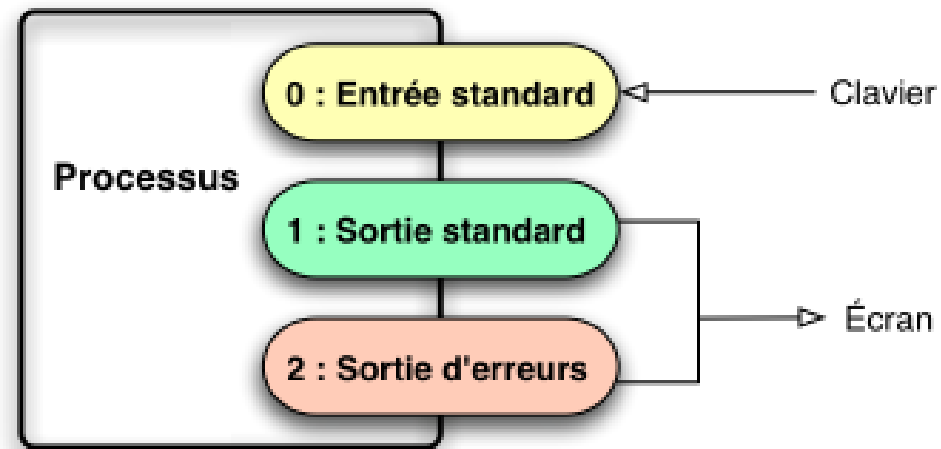


# Table des matières

- ▶ Les flux en Bash
- ▶ Redirection des flux
- ▶ Exemple pipeline
- ▶ Opérations arithmétiques
- ▶ Exercices

# Les flux en Bash :

- ▶ Chaque application lancée en Bash possède des flux d'entrée / sortie.
- ▶ Sous les systèmes Unix, chaque processus possède trois descripteurs de flux.
  - L'entrée standard
  - La sortie standard
  - La sortie d'erreurs



# Redirection des flux :

Ces trois flux peuvent être redirigé vers d'autres sources. On appelle ça du pipelining.

Voici les outils qui permettent de réaliser du pipelining

- > permet de rediriger le flux de sortie (stdout) dans un fichier
- < permet de rediriger un fichier en flux d'entrée (stdin) pour une commande.
- | permet de rediriger la sortie standard de la commande de gauche sur l'entrée standard de la commande de droite
- >> permet de rediriger la sortie d'une commande à la fin d'un fichier.

# Exemple pipeline :

```
(youn@youn) - [~/Documents]
$ echo "Bonjour / hello people" > hello.txt
```

```
(youn@youn) - [~/Documents]
$ cat hello.txt | grep "Bonjour"
Bonjour / hello people
```

```
(youn@youn) - [~/Documents]
$ wc < hello.txt
1 4 23
```

```
(youn@youn) - [~/Documents]
$ echo "on ajoute une ligne" >> hello.txt
```

# Opérations arithmétiques :

Shell prend en compte que les opérations arithmétiques de base.

Pour effectuer une opération arithmétique, on doit utiliser la commande  
`$(( expression ))`

```
#!/bin/bash
a=10
b=20
echo $((a+b))
```

```
(youn@youn) - [~/Documents]
$ ./script.sh
30
```

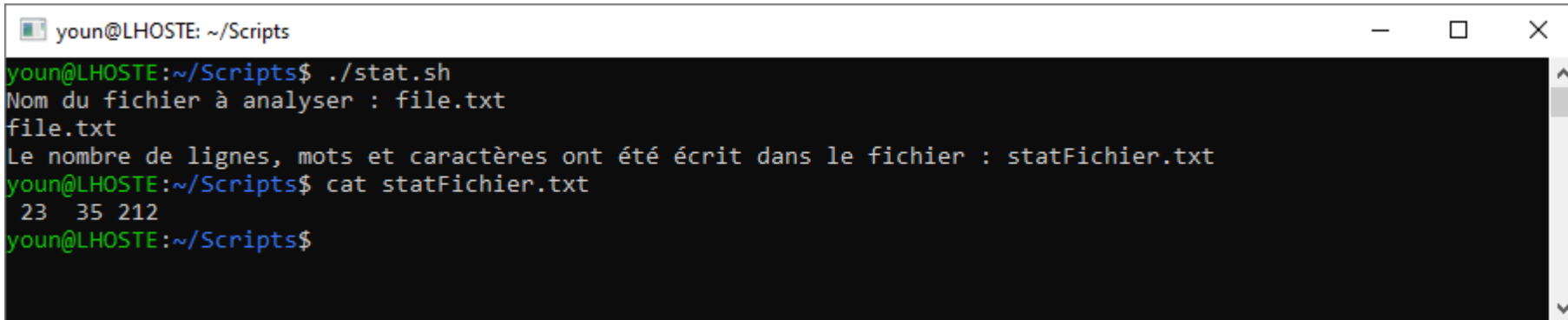
# Bash liste des opérations

► Voici une liste des opérations que vous pouvez effectuer avec Bash :

- `- +` plus et moins unaire
- `! ~` négations logique et binaire
- `* / %` multiplication, division, reste
- `+ -` addition, soustraction
- `<< >>` décalage arithmétique à gauche et à droite
- `<= >= < >` comparaisons
- `== !=` égalité et différence
- `&` ET binaire
- `^` OU exclusif binaire
- `|` OU binaire
- `&&` ET logique
- `||` OU logique
- `= *= /= %= += -= <<= >>= &= ^= |=` assignations

# Exercice 1 :

Créer un script qui demande à l'utilisateur de saisir un nom de fichier dont on va stocker le nombre de lignes, mots et caractères dans un fichier statFichier.txt

A terminal window titled 'youn@LHOSTE: ~/Scripts' with standard window controls. The terminal shows the following sequence of commands and output:

```
youn@LHOSTE:~/Scripts$ ./stat.sh
Nom du fichier à analyser : file.txt
file.txt
Le nombre de lignes, mots et caractères ont été écrit dans le fichier : statFichier.txt
youn@LHOSTE:~/Scripts$ cat statFichier.txt
23 35 212
youn@LHOSTE:~/Scripts$
```

Modifier ce script pour enregistrer que le nombre de ligne dans le fichier statFichier.txt



## Exercice 2 :

Réaliser un script qui permet de calculer le périmètre et l'air d'un cercle à partir d'un diamètre saisi par l'utilisateur.

```
youn@LHOSTE:~/Scripts$ ./air.sh  
Saisie d'un diamètre pour votre cercle : 10  
Le perimètre du cercle fait : 31.40 cm  
L'air du cercle fait : 78 cm^2  
youn@LHOSTE:~/Scripts$
```

# Questions :

