

# Systemes d'exploitation

## Chapitre 7-3

Interpréteur de commande,  
Bourne shell, E/S standard, tuyaux  
séparation et regroupement  
des commandes

# Interpréteur de commandes

---

Bourne shell:

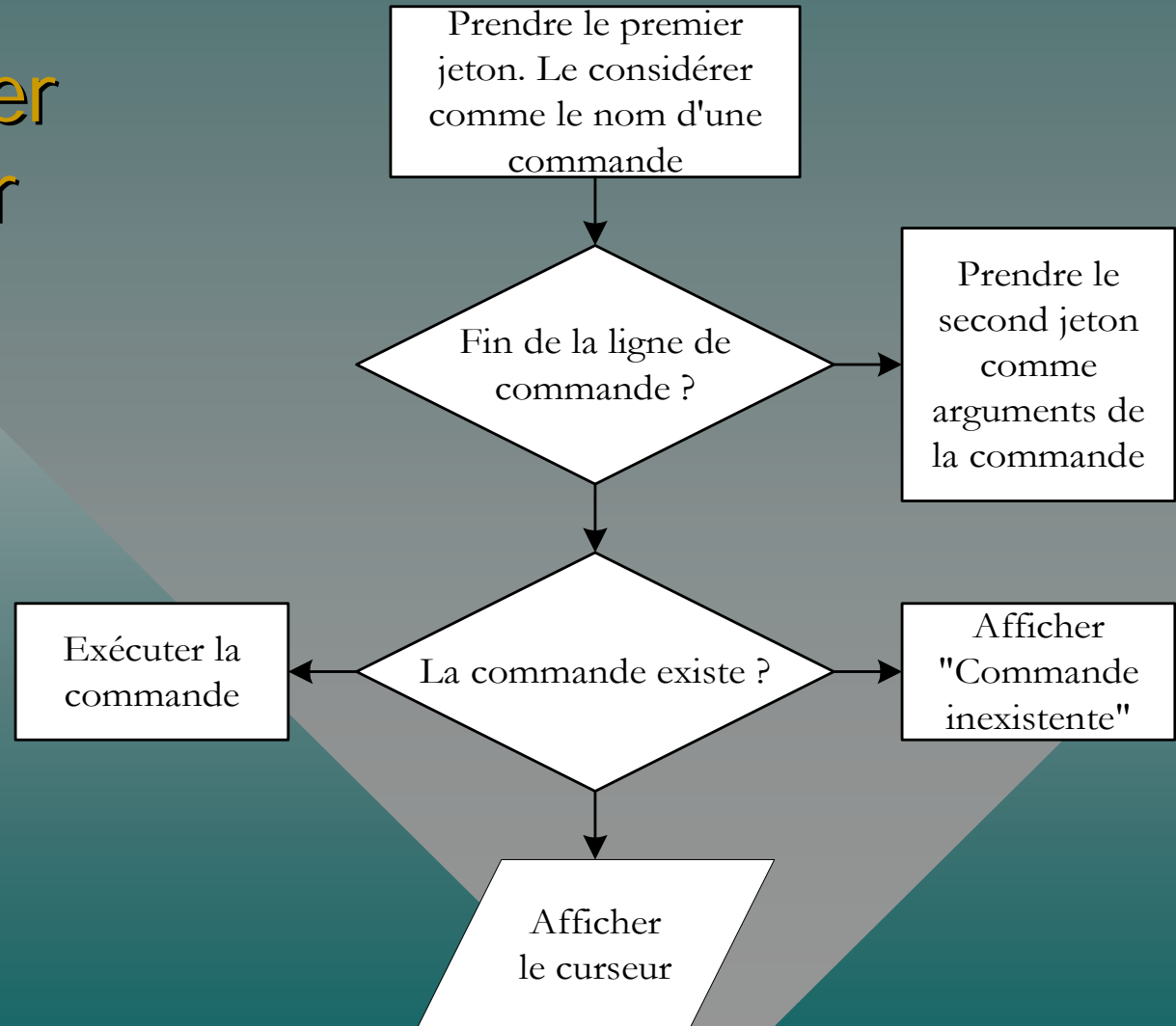
- créé par Steve Bourne;
- interpréteur par défaut de nombreux système UNIX.

Pourquoi utiliser le Bourne shell?

- plus ancien et plus répandu;
- très utilisé parmi les gestionnaires de systèmes et les développeurs de logiciels.

➤ Comment l'interpréteur traite-t-il une ligne de commande?

Pour invoquer  
l'interpréteur  
Bourne shell  
donner la  
commande  
sh(1)



# Entrée-sortie standard

---

- Entrée standard → normalement attachée à l'écran.
  - possède le numéro de descripteur 0.
- Sortie standard → normalement attachée au clavier.
  - possède le numéro de descripteur 1.
- Erreur standard → normalement attachée à l'écran.
  - possède le numéro de descripteur 2.

# Entrée-sortie standard

---

- Il est possible de rediriger les entrées-sorties standards à des fichiers ou à d'autres programmes et commandes.
- Redirection de l'entrée standard
  - `cat < NomFichier`

Commande

Redirection de l'entrée standard à un fichier.  
L'entrée de la commande `cat(1)` est le fichier  
NomFichier

# Entrée-sortie standard

---

## ➤ Redirection de la sortie standard

- `cat > NomFichier`

Commande

Redirection de la sortie standard à un fichier. La sortie de la commande `cat(1)` est enregistré dans le fichier `NomFichier`

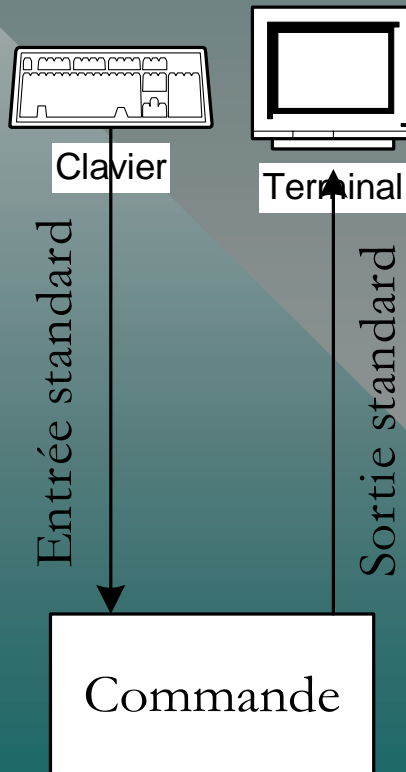
## ➤ Redirection de l'entrée et de la sortie standard

- `cat < FichierEntree > FichierSortie`

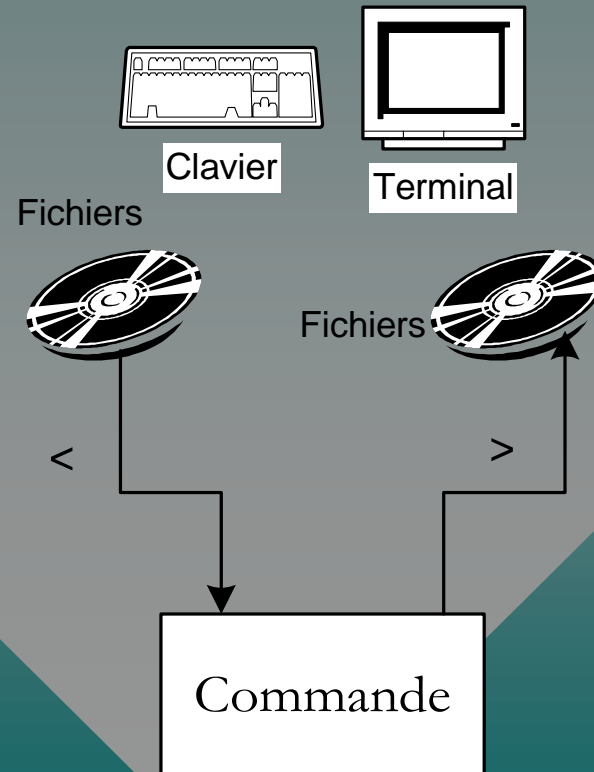
# Entrée-sortie standard

## ➤ Effet de la redirection:

Sans redirection



Entrée et sortie  
standard redirigées



# Entrée-sortie standard

---

- Rediriger la sortie vers un fichier → écrase le contenu du fichier de sortie.
- Utiliser la concaténation pour préserver le contenu du fichier de sortie:
  - `cat < toto.txt >> FichierSortie`

La sortie de la commande `cat` est ajoutée à la fin du fichier `FichierSortie`



# Entrée-sortie standard

---

- Rediriger l'erreur standard → implique une syntaxe un peu différente.
- Utilisation des numéros de descripteur.
- `commande < FichierEntree`
- `commande <0 FichierEntree`
- `commande > FichierSortie`
- `commande 1> FichierSortie`
- `commande 2> FichierErreur`

Équivalente

Équivalente

Redirection de l'erreur standard

# Entrée-sortie standard

---

➤ `cat < entree.txt > sortie.txt 2> erreur.txt`

Les messages d 'erreur de la commande `cat` seront enregistrés dans ce fichier

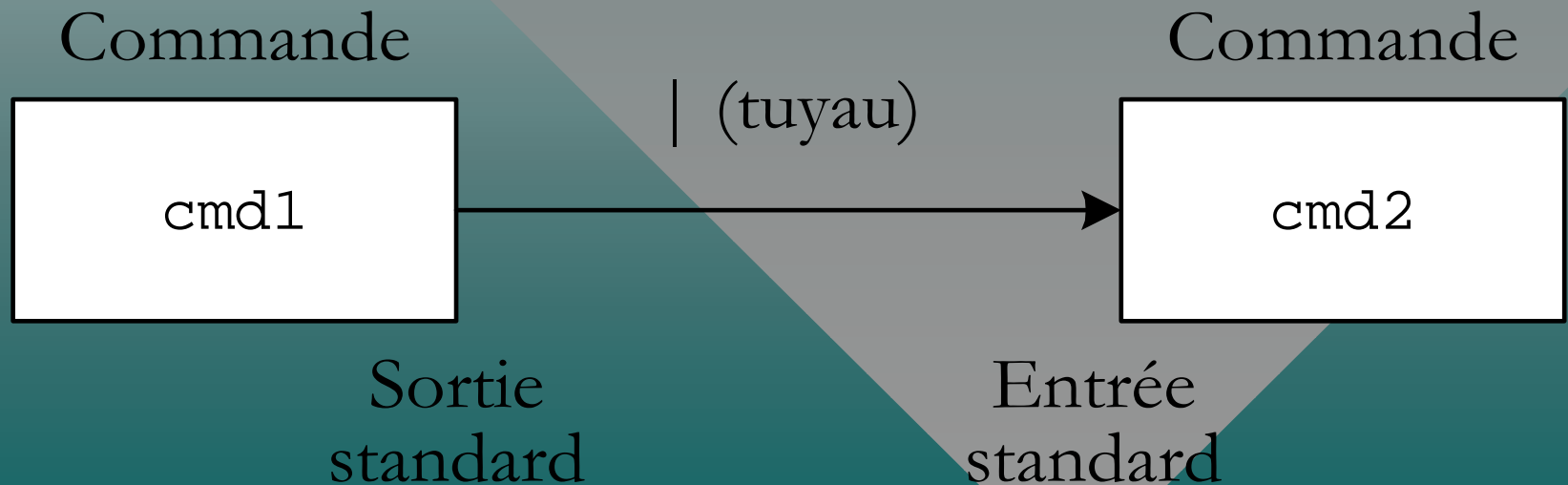
➤ `cat > sortie.txt 2>&1`

La sortie de `cat` est enregistrée dans le fichier `sortie.txt`.  
L 'erreur standard est redirigée à la sortie standard. Donc,  
l'erreur standard est également redirigée au fichier  
`sortie.txt`.

# Tuyaux (pipe)

- Tuyaux → brancher la sortie standard d'une commande à l'entrée standard d'une autre commande.

```
commandeA [-option] [parametres] | commandeB [-option] [parametres] | commandeC
```



# Tuyaux

---

- `ls -lR | more`
- La sortie de la commande `ls` est l'entrée de la commande `more`.
- `cat document.txt | tr abc ABC > traduit.txt`

Tuyau

`tr` (*translate*) remplace le caractère `a` par `A`, `b` par `B` et `c` par `C`.

La sortie de `tr` est enregistrée dans le fichier `traduit.txt`

# Exécution arrière-plan

---

- Toutes les commandes vues jusqu'à présent sont exécutées en avant-plan.
- Monopolisent le terminal.
- Ne convient pas à des tâches de longue durée.
- Lancer les commandes en arrière-plan:
  - commande [-options] [paramètre] &

Symbole indiquant une exécution en  
arrière-plan d'une commande

# Exécution arrière-plan

---

➤ Exemple:

➤ `cat passwd.dat | sort > sortie.txt &`

Laquelle des commandes sera-t-elle exécutée en arrière-plan ?

➤ Attention aux entrées-sorties des commandes exécutées en arrière-plan !!!

# Regroupement des commandes

---

➤ Exécution séquentielle d'une ligne de commandes:

- `cmd1 ; cmd2 ; cmd3`

➤ Exécution séquentielle d'une ligne de commandes par regroupement:

- `(cmd1 ; cmd2) ; cmd3`
- `(cmd1 ; cmd2) & cmd3`

Plusieurs  
combinaisons sont  
possibles

Commandes regroupées sont exécutées dans un interpréteur enfant (*subshell*).

# Regroupement des commandes

## ➤ Exemple:

```
$ pwd ; (cd ../; pwd; cp user.txt test.txt; ls -l test.txt); pwd
```

Commandes entre parenthèses sont exécutées par un interpréteur enfant.

Quelle sera la sortie de ces deux commandes ?



# Regroupement des commandes

- Explication de l'exemple précédent:
- `pwd` → affiche le nom du répertoire courant.
- `(cd ..; pwd; cp user.txt test.txt; ls -l test.txt)`

Remonter  
vers le  
répertoire  
parent

Copier le fichier `user.txt`  
vers le fichier `test.txt`

Montrer que le  
fichier a été copié

Affiche le nom du  
répertoire courant

Toutes ces commandes  
sont exécutées dans un  
interpréteur enfant

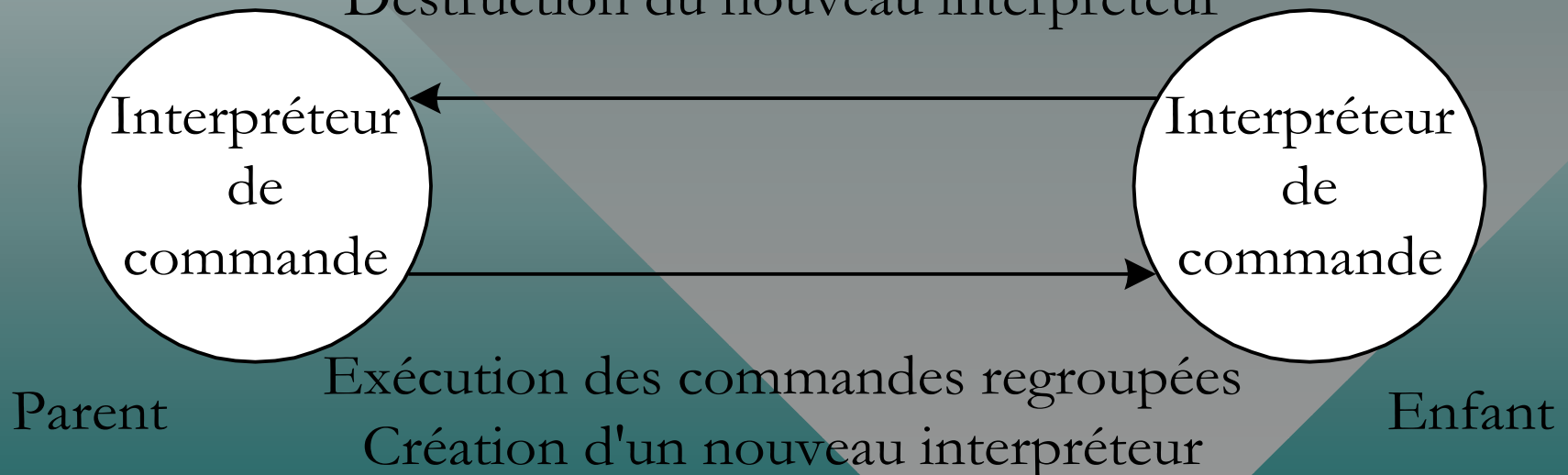
- `pwd` → revient à l'interpréteur père et affiche le nom du répertoire courant.

# Regroupement des commandes

- Explication de l'exemple précédent (suite):

Fin de l'exécution des commandes regroupées

Destruction du nouveau interpréteur



# Fichiers de commande

---

- Conditions d'un fichier de commande:
  - contient des commandes et instructions légales de Bourne shell;
  - fichier est lisible et/ou exécutable par le Bourne shell.
- Exécution d'un fichier de commande:
  - sh FichierCommande
  - contient à la première ligne du fichier la directive: `#!/bin/sh`

# Variables de l'interpréteur

## ➤ Variables prédéfinies:

Variable	Signification
HOME	Répertoire de travail de l'utilisateur.
PATH	Chemins de fouille pour trouver une commande, un programme. Les chemins sont séparés par le caractère <code>:</code> .
MAIL	Chemin indiquant la boîte de courriers de l'utilisateur.
PS1	Le caractère représentant le curseur. Par défaut, le caractère <code>\$</code> joue le rôle de curseur.
PS2	Lorsqu'une commande s'étend sur plus d'une ligne, l'interpréteur de commande affiche un second caractère pour indiquer qu'il faut continuer l'entrée. Par défaut c'est le caractère <code>&gt;</code> qui est contenu dans cette variable.
IFS	Les caractères qui jouent le rôle de séparateur des commandes. En plus des caractères espace et TAB, l'interpréteur considère les caractères contenus dans IFS comme séparateurs supplémentaires.
CDPATH	Le contenu de cette variable modifie le comportement de la commande <code>cd(1)</code> . Voir le manuel en-ligne pour les détails de cette commande.
TZ	Cette variable indique la zone horaire (Time Zone) utilisée.

# Variables de l'interpréteur

## ➤ Variables prédéfinies (suite):

Variable	Signification
\$*	Contient les paramètres entrés dans la ligne de commandes lors de l'invocation du fichier de commandes.
\$0	Nom du fichier de commande.
\$n	Valeur du n° paramètre de la ligne de commande.
\$*	Le contenu de la ligne de commande.
\$#	Le nombre de paramètres dans la ligne de commandes lors de l'invocation du fichier de commandes.
\$?	L'état de terminaison (un numéro) de la dernière commande exécutée.
\$\$	Le numéro du processus PID correspondant à l'interpréteur de commande.
#!	Le numéro du processus PID correspondant à la dernière tâche exécutant en arrière-plan.

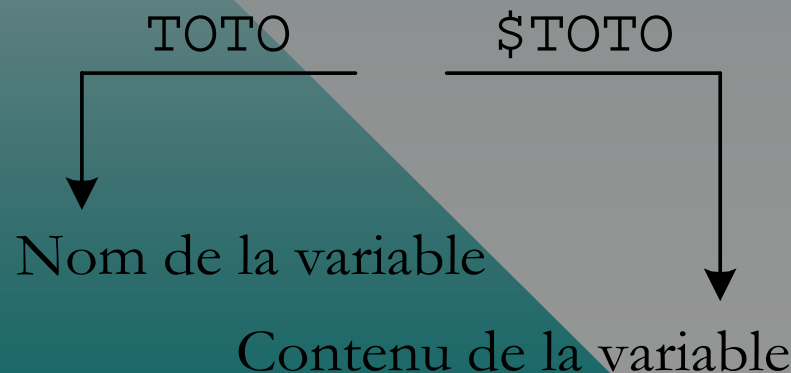
# Variables de l'interpréteur

---

## ➤ Réglage des variables:

- `var=valeur`
- `TOTO="Bonjour les SEPSistes"`
- `Allo="$TOTO du cours GPA435"`

## ➤ Explication:



# Variables de l'interpréteur

---

- Éliminer une variable de l'environnement:
  - `unset NomVariable`
- Voir le contenu d'une variable:
  - `echo $NomVariable`
- Voir le contenu de toutes les variables d'environnement:
  - `set(1)`

# Variables de l'interpréteur

- Transmettre une variable à un interpréteur enfant:
  - export(1)
- Exemple d'exportation d'une variable:

```
$ REPERTOIRE="Mon repertoire de travail est: $HOME"  
$ echo $REPERTOIRE  
Mon repertoire de travail est: /export/home/exa/personnel/wong  
$ sh  
$ echo $REPERTOIRE  
$
```

Variable REPERTOIRE n'est pas exportée et elle n'est pas disponible dans les interpréteurs enfants.



# Variables de l'interpréteur

## ➤ Exemple d'exportation d'une variable:

```
$ REPERTOIRE="Mon repertoire de travail est: $HOME"  
$ export REPERTOIRE  
$ echo $REPERTOIRE  
Mon repertoire de travail est: /export/home/exa/personnel/wang  
$ sh  
$ echo $REPERTOIRE  
$ Mon repertoire de travail est: /export/home/exa/personnel/wang
```

Variable REPERTOIRE est exportée et elle est disponible dans les interpréteurs enfants.