

Commandes internes et externes

Ce chapitre présente et explique de manière détaillée les fonctionnalités de base du shell couramment utilisées dans les commandes Unix.

Une commande Unix appartient à l'une des deux catégories suivantes :

1. Les commandes externes

Une commande externe est un fichier localisé dans l'arborescence. Par exemple, lorsqu'un utilisateur lance la commande `ls`, le shell demande au noyau Unix de charger en mémoire le fichier `/usr/bin/ls`.

Sont considérés comme commandes externes les fichiers possédant l'un des formats suivants :

- Fichiers au format binaire exécutable ;
- Fichiers au format texte représentant un script de commandes (qui peut être écrit en Shell ou dans un autre langage tel que le Perl).

La commande `file` donne une indication sur le type de données contenues dans un fichier.

Exemples

La commande `ls` est un fichier au format binaire exécutable. Résultat de la commande `file` :

```
$ file /usr/bin/ls
/usr/bin/ls:  ELF 32-bit MSB executable SPARC Version 1,
dynamically linked, stripped
```

La commande `monscript` est un script shell. Résultat de la commande `file` :

```
$ file /home/christie/monscript
monscript:  ascii text
```

La commande `monscript.pl` est un script perl. Résultat de la commande `file` :

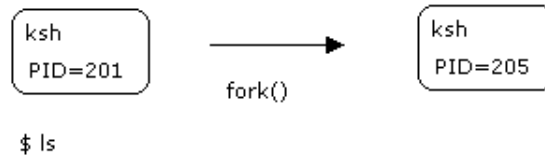
```
$ file /home/christie/monscript.pl
monscript.pl:  ascii text
```



L'argument de la commande `file` est un nom de fichier exprimé en relatif ou en absolu.

Les commandes externes sont exécutées par l'intermédiaire d'un shell enfant (cf. Figure 1).

1ère étape: Duplication du shell



2ème étape: Le code du shell enfant est substitué par le code de la commande "ls"

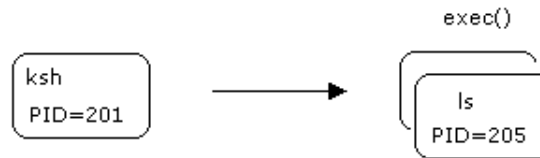


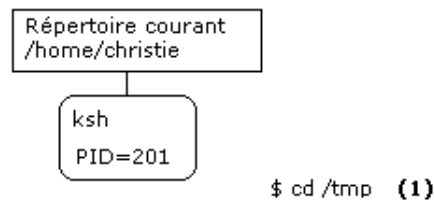
Figure 1 : Exécution d'une commande externe

2. Les commandes internes

Une commande interne est intégrée au processus shell (c'est le shell qui exécute l'action). Elle ne correspond donc en aucun cas à un fichier sur le disque.

La commande interne est exécutée par le shell courant (cf. Figure 2).

1ère étape: Lancement de la commande "cd". Le shell reconnaît l'une de ses commandes internes



2ème étape: Le shell modifie la valeur de son répertoire courant

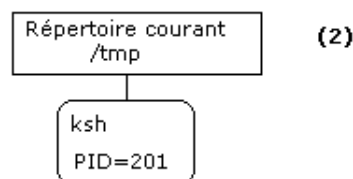


Figure 2 : Exécution d'une commande interne

La commande **type** indique si une commande possède une implémentation interne.


Exemples

La commande **cd** est une commande interne.

```
$ type cd  
cd is a shell builtin
```

La commande **ls** est une commande externe.

```
$ type ls
ls is /usr/bin/ls
```

 `type` prend en argument le nom d'une commande. Si cette dernière n'est pas interne, elle est recherchée dans les répertoires cités dans `PATH` (cf. chapitre Paramétrage de l'environnement de travail - Variables d'environnement).

3. Implémentation interne et implémentation externe

Certaines commandes ont une implémentation interne et une implémentation externe. Dans ce cas :

- la commande interne est lancée en priorité ;
- l'exécution d'une commande interne est plus rapide que l'exécution d'une commande externe ;
- la commande `type` indique que la commande est interne, mais ne précise pas qu'il existe également une implémentation externe.

Exemple

La commande `pwd` est une commande interne au shell :

```
$ type pwd
pwd is a shell builtin
```

La commande `pwd` possède également une implémentation externe :

```
$ ls -l /usr/bin/pwd
-r-xr-xr-x 1 bin  bin    4616 Oct 6 1998 /usr/bin/pwd
```

C'est la commande interne qui est exécutée en priorité :

```
$ pwd
/home/christie
```

Pour forcer l'exécution de la commande externe, il faut indiquer explicitement l'emplacement de la commande (en absolu ou en relatif) :

```
$ /usr/bin/pwd
/home/christie
$ cd /usr/bin
$ ./pwd
/usr/bin
$
```