

```
$ var=a.c
$ tree
.
└── b.c
$ cp ?.c $var
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. [Oui] Je l'exécute
6. [Non] Je lève une erreur

```
$ var=a.c
$ tree
.
└── b.c
$ cp ?.c $var ↵
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. [Oui] Je l'exécute
6. [Non] Je lève une erreur

```
$ var=a.c
$ tree
.
└── b.c
$ cp ?.c $var
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. **Expansion des variables**
3. Expansion des chemins
4. Est-ce une commande valide ?
5. [Oui] Je l'exécute
6. [Non] Je lève une erreur

```
$ var=a.c
$ tree
.
└── b.c
$ cp ?.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. **Expansion des variables**
3. Expansion des chemins
4. Est-ce une commande valide ?
5. [Oui] Je l'exécute
6. [Non] Je lève une erreur

```
$ var=a.c
$ tree
.
└── b.c
$ cp ?.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »

2. **Expansion des variables**

```
$ echo cp $unset_var destination/
cp destination/
```

```
$ var=a.c
$ tree
.
└── b.c
$ cp ?.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. **Expansion des chemins**
4. Est-ce une commande valide ?
5. [Oui] Je l'exécute
6. [Non] Je lève une erreur

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. **Expansion des chemins**
4. Est-ce une commande valide ?
5. [Oui] Je l'exécute
6. [Non] Je lève une erreur

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**

```
$ tree
.
├── b.c
├── d.c
└── destination

$ echo cp *.o *
cp *.o b.c d.c destination
```



```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. **Est-ce une commande valide ?**
5. [Oui] Je l'exécute
6. [Non] Je lève une erreur

**Shell :**

« Est-ce que je connais la commande **cp** ? »

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**

```
$ tree /bin
/bin
├── bash
├── ...
├── cp
├── mkdir
└── ...
```

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
$
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. **[Oui] Je l'exécute**
6. **[Non] Je lève une erreur**

**Shell :**

« Est-ce que je connais la commande **cp** ? »

→ Oui

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
$ tree
.
├── a.c
└── b.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. **[Oui] Je l'exécute**
6. **[Non] Je lève une erreur**

**Shell :**

« Est-ce que je connais la commande **cp** ? »

→ Oui

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
$ tree
.
├── a.c
└── b.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**

```
$ tree /bin
/bin
├── bash
├── ...
├── cp
├── mkdir
└── ...
$ unknowncmd
bash: unknowncmd: commande introuvable
```

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
$ tree
.
├── a.c
└── b.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »

2. Expansion des **variables**

```
$ tree
.
├── a.c
└── b.c
$ cp a.c b.c rep
cp: la cible «rep» n'est pas un répertoire
```

```
$ var=a.c
$ tree
.
├── b.c
$ cp b.c a.c
$ pstree
init──ModemManager─...
    │
    ├──...
    ├──...──bash──pstree
    └──...
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?

### 5. [Oui] Je l'exécute

1. Je passe les arguments au programme

### Shell :

« Est-ce que je connais la commande **cp** ? »

→ Oui

→ J'encapsule les arguments

→ J'invoque un **processus** de ce programme auquel je donne les arguments

```
$ pstree
init--ModemManager--...
    |
    |--...
    |--...--bash--pstree
    |--...

$ cd ..
$ tree

.
├── src
└── script.sh
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. [Oui] Je l'exécute
  1. Je passe les arguments au programme

## Shell :

« Est-ce que je connais la commande **cp** ? »

→ Oui

→ J'encapsule les arguments

→ J'invoque un **processus** de ce programme auquel je donne les arguments



```

$ pstree
init--ModemManager--...
    |
    |--...
    |--...--bash--pstree
    |--...

$ cat script.sh
#!/bin/bash
pstree
echo fin
$ ./script.sh
init--ModemManager--...
    |
    |--...
    |--...--bash--bash--pstree
    |--...

fin

```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. **[Oui] Je l'exécute**
  1. Je passe les arguments au programme

## Shell :

« Est-ce que je connais la commande **cp** ? »

→ Oui

→ J'encapsule les arguments

→ J'invoque un **processus** de ce programme auquel je donne les arguments

```

$ pstree
init--ModemManager--...
    |
    |--...
    |--...--bash--pstree
    |--...

$ cat script.sh
#!/bin/bash
pstree
echo fin
$ ./script.sh
init--ModemManager--...
    |
    |--...
    |--...--bash--bash--pstree
    |--...

fin

```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. **[Oui] Je l'exécute**
  1. Je passe les arguments au programme

## Shell :

« Est-ce que je connais la commande **cp** ? »

→ Oui

→ J'encapsule les arguments

→ J'invoque un **processus** de ce programme auquel je donne les arguments

```

$ pstree
init--ModemManager--...
    |
    |--...
    |--...--bash--pstree
    |--...

$ cat script.sh
#!/bin/sh
pstree
echo fin
$ ./script.sh
init--ModemManager--...
    |
    |--...
    |--...--bash--sh--pstree
    |--...

fin

```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. **[Oui] Je l'exécute**
  1. Je passe les arguments au programme

## Shell :

« Est-ce que je connais la commande **cp** ? »

→ Oui

→ J'encapsule les arguments

→ J'invoque un **processus** de ce programme auquel je donne les arguments