

TD5

Gestion de processus : première approche

```
$ cat com1
echo je suis dans com1
ps jf
com2
echo je suis de retour dans com1
$ cat com2
echo je suis dans com2
ps jf
com3
echo je suis de retour dans com2
$ cat com3
echo je suis dans com3
ps jf
$
```

Essayer de lancer la commande "com1". Que se passe-t-il ? Que devez-vous faire pour que l'exécution puisse être lancée ?

```
$ com1
bash: ./com1: Permission non accordée.
$ ls -l
total 15
-rw-r--r--  1 guest  users          67 Nov 25 23:14 com1
-rw-r--r--  1 guest  users          67 Nov 25 23:15 com2
-rw-r--r--  1 guest  users          29 Nov 25 23:15 com3
$ chmod u+x com?
$ ls -l
total 15
-rwxr--r--  1 guest  users          67 Nov 25 23:14 com1
-rwxr--r--  1 guest  users          67 Nov 25 23:15 com2
-rwxr--r--  1 guest  users          29 Nov 25 23:15 com3
```

Une fois ce problème résolu, lancer la commande "com1" en redirigeant les traces dans le fichier "trace1".

```
[guest@localhost rep]$ com1 > trace1
```

Reprendre les fichiers correspondant aux commandes "com1", "com2" et "com3" et remplacer les appels aux commandes "com?" par ". com?". Vous pouvez créer de nouveaux fichiers "com1p", "com2p" et "com3p" ainsi modifiés.

```
$ cat com1p
echo je suis dans com1p
ps jf
. com2p
echo je suis de retour dans com1p
$ cat com2p
echo je suis dans com2p
ps jf
. com3p
echo je suis de retour dans com2p
$ cat com3p
echo je suis dans com3p
ps jf
$ com1p
bash: ./com1p: Permission non accordée.
$ chmod u+x com?p
```

Lancer la nouvelle commande "com1p" en redirigeant les traces dans le fichier "trace2".

```
$ com1p > trace2
```

Reprendre les fichiers correspondant aux commandes "com1", "com2" et "com3" et remplacer les appels aux commandes "com?" par "exec com?". Vous pouvez créer de nouveaux fichiers "com1e", "com2e" et "com3e" ainsi modifiés.

```
$ cat com1e
echo je suis dans com1e
ps jf
exec com2e
echo je suis de retour dans com1e
$ cat com2e
echo je suis dans com2e
ps jf
exec com3e
echo je suis de retour dans com2e
$ cat com3e
echo je suis dans com3e
ps jf
$ com1e
bash: ./com1e: Permission non accordée.
$ chmod u+x com?e
```

Lancer la nouvelle commande "com1e" en redirigeant les traces dans le fichier "trace3".

```
$ com1e > trace3
```

A partir de l'analyse des trois fichiers de trace obtenus ("trace1", "trace2" et "trace3"), expliquer les différences entre l'appel d'une commande "com" (\$ com), l'utilisation de la commande "." (\$. com) et l'utilisation de la commande "exec" (\$ exec com).

```
$ cat trace1
je suis dans com1
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
 340 1321 1321  340  p0  1362  S    502  0:00  su -l guest
 1321 1322 1322  340  p0  1362  S    502  0:00  \_ -bash
 1322 1362 1362  340  p0  1362  S    502  0:00  \_ -bash
 1362 1363 1362  340  p0  1362  R    502  0:00  \_ ps jf
je suis dans com2
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
 340 1321 1321  340  p0  1362  S    502  0:00  su -l guest
 1321 1322 1322  340  p0  1362  S    502  0:00  \_ -bash
 1322 1362 1362  340  p0  1362  S    502  0:00  \_ -bash
 1362 1364 1362  340  p0  1362  S    502  0:00  \_ -bash
 1364 1365 1362  340  p0  1362  R    502  0:00  \_ ps jf
je suis dans com3
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
 340 1321 1321  340  p0  1362  S    502  0:00  su -l guest
 1321 1322 1322  340  p0  1362  S    502  0:00  \_ -bash
 1322 1362 1362  340  p0  1362  S    502  0:00  \_ -bash
 1362 1364 1362  340  p0  1362  S    502  0:00  \_ -bash
 1364 1366 1362  340  p0  1362  S    502  0:00  \_ -bash
 1366 1367 1362  340  p0  1362  R    502  0:00  \_ ps jf
je suis de retour dans com2
je suis de retour dans com1
```

```

$ cat trace2
je suis dans com1p
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
340  1321  1321  340  p0  1373  S    502  0:00  su -l guest
1321  1322  1322  340  p0  1373  S    502  0:00  \_ -bash
1322  1373  1373  340  p0  1373  S    502  0:00  \_ -bash
1373  1374  1373  340  p0  1373  R    502  0:00  \_ ps jf
je suis dans com2p
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
340  1321  1321  340  p0  1373  S    502  0:00  su -l guest
1321  1322  1322  340  p0  1373  S    502  0:00  \_ -bash
1322  1373  1373  340  p0  1373  S    502  0:00  \_ -bash
1373  1375  1373  340  p0  1373  R    502  0:00  \_ ps jf
je suis dans com3p
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
340  1321  1321  340  p0  1373  S    502  0:00  su -l guest
1321  1322  1322  340  p0  1373  S    502  0:00  \_ -bash
1322  1373  1373  340  p0  1373  S    502  0:00  \_ -bash
1373  1376  1373  340  p0  1373  R    502  0:00  \_ ps jf
je suis de retour dans com2p
je suis de retour dans com1p

```

```

$ cat trace3
je suis dans com1e
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
340  1321  1321  340  p0  1382  S    502  0:00  su -l guest
1321  1322  1322  340  p0  1382  S    502  0:00  \_ -bash
1322  1382  1382  340  p0  1382  S    502  0:00  \_ -bash
1382  1383  1382  340  p0  1382  R    502  0:00  \_ ps jf
je suis dans com2e
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
340  1321  1321  340  p0  1382  S    502  0:00  su -l guest
1321  1322  1322  340  p0  1382  S    502  0:00  \_ -bash
1322  1382  1382  340  p0  1382  S    502  0:00  \_ -bash
1382  1384  1382  340  p0  1382  R    502  0:00  \_ ps jf
je suis dans com3e
PPID  PID  PGID  SID  TTY  TPGID  STAT  UID  TIME  COMMAND
340  1321  1321  340  p0  1382  S    502  0:00  su -l guest
1321  1322  1322  340  p0  1382  S    502  0:00  \_ -bash
1322  1382  1382  340  p0  1382  S    502  0:00  \_ -bash
1382  1385  1382  340  p0  1382  R    502  0:00  \_ ps jf

```

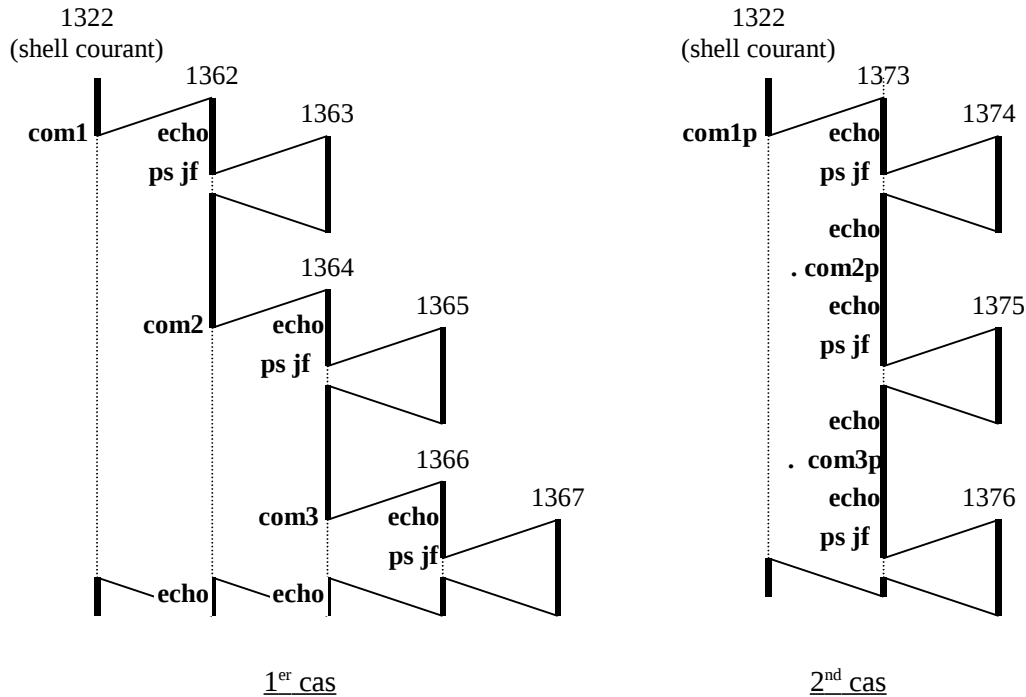
La commande suivante donne le numéro du processus correspondant au shell courant.

```

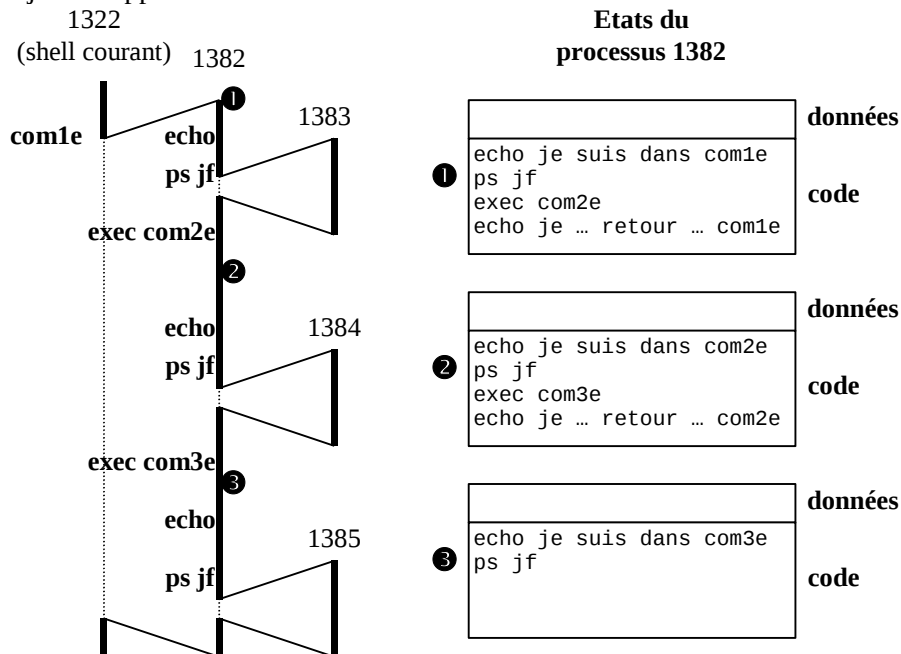
$ echo $$
1322

```

Dans le premier cas, un processus est créé à chaque appel de commande. Dans le second cas, un processus est créé pour l'exécution de la commande "com1p". L'utilisation de la commande "." provoque l'exécution des fichiers script "com2p" et "com3p" dans ce même processus. En revanche, la commande "." ne peut pas être utilisée avec des fichiers binaires tels que "ps". Un processus est donc créé à chaque appel de la commande "ps".



Dans le troisième cas, un processus est créé pour l'exécution du fichier "com1e". Chaque processus est composé d'une partie données et d'une partie contenant le code. L'utilisation de la commande "exec" provoque le remplacement du code du processus appelant par celui de la commande appelée. Ceci explique pourquoi d'une part, aucun processus n'est créé lors de l'appel aux commandes "exec com2e" et "exec com3e" et d'autre part, que les commandes "echo je suis de retour dans com1e" et "je suis de retour dans com2e" ne soient jamais appelées.



3^{ième} cas