

Commandes / processus

A) Qu'est ce qu'une commande ou un programme ?

B) Comment lance-t-on un programme?

Par quelle “magie”, lorsque vous tapez le nom d'une commande/programme dans un terminal, se passe-t-il qqchose ?

Commandes / processus

A) Une commande/programme est un fichier exécutable situé “qqpart” dans l'arborescence (dont les droits d'exécution(x) sont positionnés)

B)

1) soit on décrit explicitement le chemin jusqu'à cet exécutable:

Ex:

```
$/usr/local/bin/programme
```

```
$ bin/mes_pgms/programme
```

Commandes / processus

2) soit cet exécutable est situé dans un des répertoires définis par le **PATH**

PATH est une variable qui définit une liste ordonnée de répertoires dans lesquels le système cherche les exécutables.

```
$echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/  
sbin:/usr/bin:/sbin:/bin:/usr/games
```

```
$which firefox
```

```
/usr/bin/firefox
```

```
sinon: "not found"
```

Commandes / processus

Modifier le PATH:

```
$ PATH=$PATH:nouveau_repertoire
```

Attention ! Vous risquez de ne plus pouvoir lancer de commandes si vous vous trompez !!!

```
Ex: $ PATH=~ /bin
```

Attention: le PATH n'est modifié que pour le terminal dans lequel vous l'avez défini

→ **~/ .bashrc** (c.f plus loin)

Commandes / processus

Lorsque vous lancez une commande, vous ne pouvez pas en lancer une autre tant que la précédente ne s'est pas terminée.

→ la commande est alors lancée en “premier plan” ou “**foreground**”

```
$ nedit ~olivier/coursunix/TP12
```

Vous pouvez interrompre (tuer) une commande avec

CTRL C

Commandes / processus

Vous pouvez suspendre (stopper) une commande avec

CTRL Z

le temps d'en taper une autre

\$!s

Vous pouvez relancer la commande précédente en la relançant en “premier plan”

\$fg

Ou en “second plan” (background)

\$bg

Commandes / processus

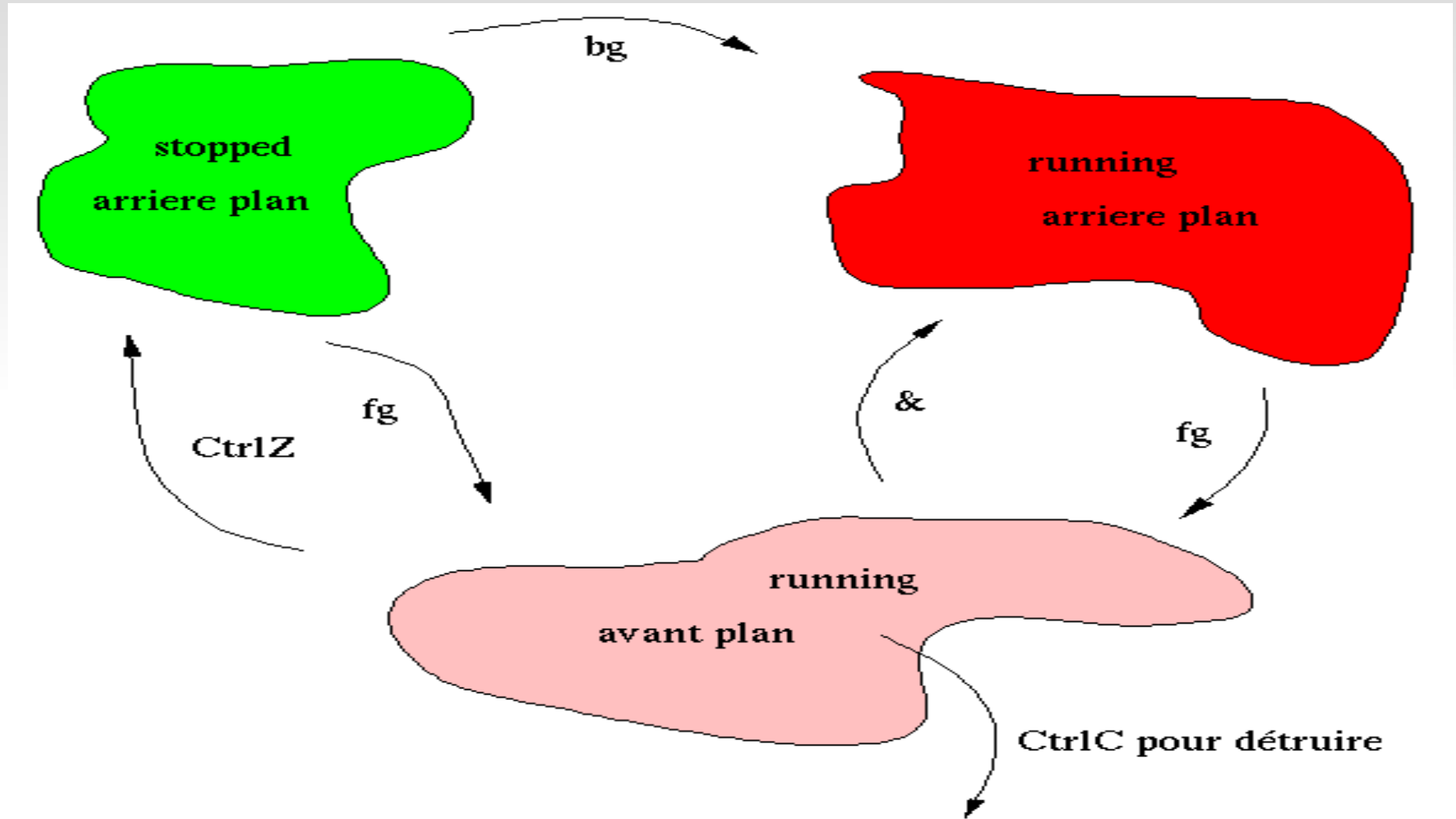
Vous pouvez lancer une commande directement en second plan (background):

```
$ cde [opt] [arguments] &
```

Puis la reprendre en premier plan

```
$ fg ... CTRL Z etc ....
```

Commandes / processus



c.f: http://www.idris.fr/data/cours/unix/user/unix_u_cours.html

Commandes / processus

Pour lister TOUS les processus en cours d'exécution:

```
$ ps auxw
```

| USER | PID | %CPU | %MEM | VSZ | RSS | TTY | STAT | START | TIME | COMMAND |
|---------|------|------|------|-------|------|-------|------|-------|------|----------------------------|
| root | 5106 | 0.0 | 0.1 | 13256 | 4912 | ? | S | 12:09 | 0:00 | /usr/bin/timidity -Os -iAD |
| root | 5111 | 0.0 | 0.0 | 1808 | 532 | tty1 | Ss+ | 12:09 | 0:00 | /sbin/getty 38400 tty1 |
| postfix | 5204 | 0.0 | 0.0 | 5840 | 1732 | ? | S | 12:09 | 0:00 | qmgr -l -t fifo -u |
| op | 6637 | 0.0 | 0.2 | 11364 | 7200 | ? | Rs | 12:27 | 0:00 | /usr/bin/xterm |
| 22456 | 6638 | 0.0 | 0.0 | 4812 | 2064 | pts/1 | Ss | 12:27 | 0:00 | bash |

Commandes / processus

Pour tuer un/plusieurs process(es):

```
$ kill -9 PID
```

```
$ killall nom_cde
```

Lister les n processus qui occupent le plus de CPU/MEM....

```
$ top
```

Commandes / processus

Attention: si on ferme sa session, les processus sont tués !!!

```
$ nohup cde [opts] [arguments]
```

On retrouve ce qui devait s'afficher dans le fichier nohup.out

Commandes / processus

Lancement en différé:

```
$ at qqchose
```

```
cde1
```

```
cde2
```

```
...
```

```
CTRL D
```

```
Ex: $ at now + 3 hours
```

```
Ex: $ at midnight ... (man at)
```

```
$ atq
```

```
$ atrm no_job
```

Commandes / processus

Périodiquement:

```
$ man crontab
```

Rendre un process moins prioritaire

```
$renice +level PID
```

Executer qd le système “n'a que ca a faire”

```
$ batch cde
```

Caractéristiques du bash

Il existe plusieurs interpréteurs de commande / shells

sh, ksh, csh, tcsh, bash, zsh ...

Par défaut, vous utilisez le bash

Pour changer: demandez aux sysadm !!!

Ou alors tapez le nom du shell désiré ...

\$ bash

Caractéristiques du bash

Mécanisme d'historique

```
$ history
```

```
$ !!
```

```
$ !3
```

```
$ !-2
```

```
$ !truc
```

```
$ mkdir toto
```

```
$ rm -r !$
```

```
$ mkdir toto tutu
```

```
$ rm -r !*
```

```
$ mkdir !:2
```

Caractéristiques du bash

Mecanisme de substitution sur la ligne de commande

```
$ cp ~oliver/coursunix/TP22 ~
```

```
$ ^er^ier
```


Caractéristiques du bash

Alias(es): `$ alias`

```
$ alias h='history'
```

```
$ alias fi='firefox'
```

```
$ alias ll='ls -l'
```

```
$ alias rm='rm -i'
```

```
$ alias cp='rm'
```

```
$ \cp fic1 fic2 ou $unalias cp
```

Les aliases n'existent que dans le terminal dans lequel ils ont été définis !!!

Comment définir des aliases valables dans tous les terminaux ??? → `~/ .bashrc`

Caractéristiques du bash

`~/ .bashrc` est un fichier qui s'exécute à l'ouverture de chaque terminal/shell.

On y place donc les commandes que l'on souhaite voir s'exécuter systématiquement ...

Rem: `~/ .bashrc` s'exécute après `/etc/bashrc`