

Pratique des machines, installation, utilisation

Utiliser le Shell bash :
Tâches, tâches en arrière plan et
environnement

Passer un processus en arrière plan

Lancer un processus puis le placer en tâche de fond

```
$ sleep 100000
```

```
^Z
```

```
[1]+  Arrêté                sleep 100000
```

Ctrl+Z place la tâche en arrière plan et l'arrête

Voir les processus en arrière plan

```
$ jobs
```

```
[1]+  Arrêté                sleep 100000
```

Il faudra ensuite redémarrer le processus...

Lancer un processus directement en tâche de fond avec &

```
$ sleep 400000 &
```

```
[2] 30135
```

Repasser un processus au premier plan

Remettre un processus au premier plan :

```
$ fg [job_id]
```

Par défaut **job_id** correspond au processus signalé par +

```
$ jobs
[1]   En cours d'exécution   sleep 100000 &
[2]   En cours d'exécution   sleep 200000 &
[3]-  En cours d'exécution   sleep 300000 &
[4]+  En cours d'exécution   sleep 400000 &
```

```
$ fg
sleep 400000
^C
```

```
$ jobs
[1]   En cours d'exécution   sleep 100000 &
[2]-  En cours d'exécution   sleep 200000 &
[3]+  En cours d'exécution   sleep 300000 &
```

```
$ fg 2
sleep 200000
^C
```

Redémarrer un processus en arrière plan

Redémarrer un processus placé en arrière plan :

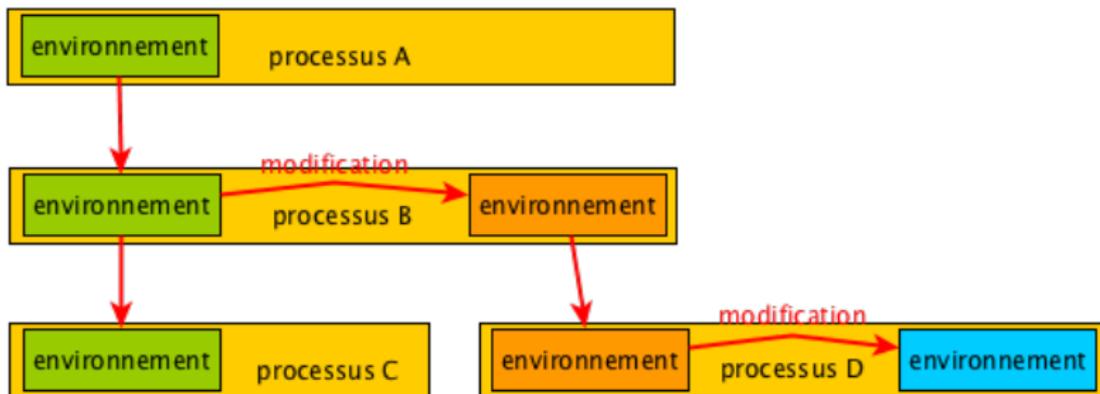
```
$ bg [job_id]
```

Par défaut **job_id** correspond au processus signalé par +

```
[1]+  Arrêté                sleep 100000
[3]   En cours d'exécution  sleep 300000 &
[4]-  Arrêté                sleep 400000
$ bg 4
[4]-  sleep 400000 &
$ jobs
[1]+  Arrêté                sleep 100000
[3]   En cours d'exécution  sleep 300000 &
[4]-  En cours d'exécution  sleep 400000 &
$ bg
[1]+  sleep 100000 &
```

L'environnement

- ▶ Ensemble de variables
- ▶ Une copie de l'environnement est transmise à tous les processus enfants
- ▶ Le processus parent n'est pas averti d'une modification de l'environnement de l'enfant
- ▶ L'enfant transmettra une copie de cet environnement modifié à ses enfants



Afficher l'environnement

```
$ printenv # Afficher tout l'environnement
XDG_VTNR=7
XDG_SESSION_ID=c4
CLUTTER_IM_MODULE=xim
SELINUX_INIT=YES
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/alinehuf
SESSION=ubuntu
(...)
```

```
$ printenv SESSION # afficher une seule variable
ubuntu
```

```
$ printenv | grep alinehuf # chercher un mot clef
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/alinehuf
USER=alinehuf
PATH=./home/alinehuf/bin:/usr/local/sbin:/usr/local/bin:/usr
HOME=/home/alinehuf
LOGNAME=alinehuf
XAUTHORITY=/home/alinehuf/.Xauthority
```

Voir/Modifier l'environnement

Modifier une variable de l'environnement

```
$ echo $PATH      # autre manière de l'afficher
/home/alinehuf/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin
$ PATH=".:$PATH"
$ printenv PATH
./home/alinehuf/bin:/usr/local/sbin:/usr/local/bin:/usr/sbi
```

Ajouter une variable à l'environnement

```
$ VAR="ma valeur"    # simple variable
$ printenv VAR
$ export VAR         # on la place dans l'environnement
$ printenv VAR
ma valeur
```

Supprimer une variable de l'environnement

```
$ unset VAR
$ echo $VAR

$ printenv VAR
```

La commande set

- ▶ Permet de modifier le comportement des fonctions et des variables de l'environnement
- ▶ Différentes options permettent d'activer ou désactiver certains comportements
 - ▶ empêcher les redirection d'écraser le contenu fichiers existants
 - ▶ avertir immédiatement quand une commande en arrière plan se termine
 - ▶ remplacer les chemins symboliques par des chemins physiques dans les commandes
etc...

La commande source

- ▶ Permet d'exécuter une commande dans le contexte du Shell courant
- ▶ Les variables locales du Shell, les programmes lancés, etc... sont visibles dans le contexte du programme lancé

Manipulations

1. Lancez un second Shell à partir du premier avec la commande **sh** (exit pour sortir).
2. Vérifiez avec **printenv** que l'environnement est le même dans les deux Shells parent et enfant.
3. Modifiez l'environnement dans le second Shell, sortez et constatez qu'il n'a pas changé dans le Shell parent.
4. Lancez un Shell avec **VAR="coucou" sh**,
VAR est-elle dans l'environnement du second Shell ?
Est-elle dans l'environnement du premier Shell ?
5. Retentez avec **VAR="coucou" VAR2="hello" sh** : quel est l'intérêt de cette manière de lancer le second Shell ?

Pratiquons !

- ▶ Copier l'archive contenant le TP dans votre répertoire personnel.

```
$ cp /opt/tp8.tar.gz ~/
```

- ▶ Décompresser le fichier tp8.tar.gz qui se trouve dans votre home.

```
$ tar xzvf tp8.tar.gz
```

- ▶ Rentrer dans le répertoire 'tp8' contenant le TP.

```
$ cd tp8
```

- ▶ Exécuter les commandes contenues dans 'start' dans l'environnement du shell (configuration pour le TP).

```
$ source start
```