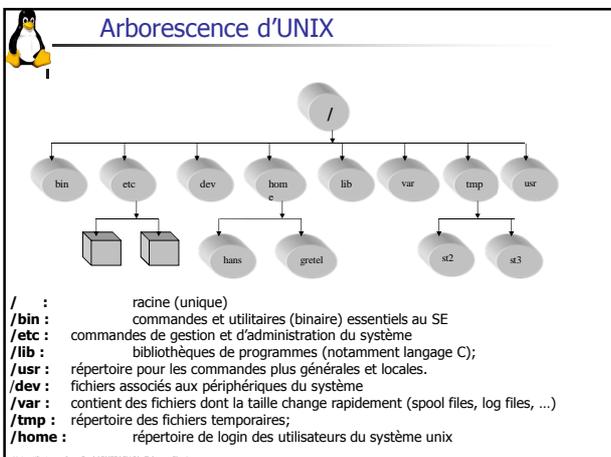
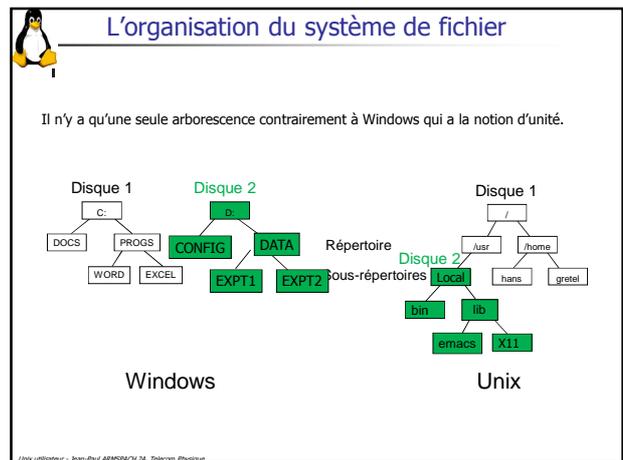
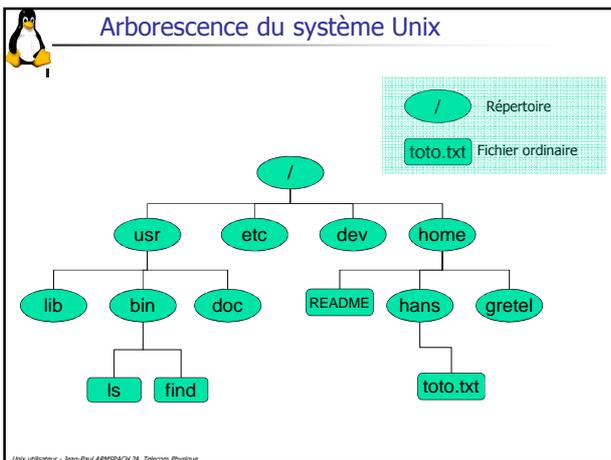


### Plan du cours

- Introduction
- Gestions des fichiers
- L'interpréteur de commande (shell) : le bash
- Les commandes Unix
- Initiation à la programmation shell
- Gestion de l'espace disque, sauvegarde
- Gestion des processus

### Le système de fichiers sous Unix

- L'unité d'information gérée par Unix est le fichier.
- Organisation des fichiers est une structure arborescente
  - de façon hiérarchique
  - une seule arborescence logique
  - graphe acyclique
  - **TOUT** est fichier
- L'arborescence logique est représentée sous forme d'arbre renversé :
  - Les noeuds : répertoires (directories)
  - Les feuilles : fichiers (files)
- Chaque utilisateur peut ajouter de nouvelles branches dans son répertoire d'accueil, ailleurs.



### La manipulation des répertoires

#### Principales commandes pour gérer répertoires

**cd (Change Directory)**

Changement de répertoire de travail

- Syntaxe :
 

```
cd [nom_répertoire]
```
- Description :
  - nom\_répertoire en relatif ou absolu, si nom\_répertoire n'est pas indiqué alors retourne au répertoire d'accueil
- Exemple :
 

```
hans> cd
hans> cd repl
hans> cd ..
```

## La manipulation des répertoires

Principales commandes pour gérer répertoires **pwd (Print Working Directory)**

Affiche le chemin d'accès complet du répertoire courant

- Syntaxe :
 

```
pwd
```
- Description : Affiche le chemin d'accès complet du répertoire courant
- Exemple :
 

```
hans> pwd
/home/hans/cours
hans> cd
hans> pwd
????????
```

http://lildemur - Jean-Paul ARNSPACH 24 - Telecom-Physique

## La manipulation des répertoires

Principales commandes pour gérer répertoires **mkdir (MaKe DIRectory)**

Création d'un ou plusieurs répertoires

- Syntaxe :
 

```
mkdir [-p] nom_répertoire[...]
```
- Options :
  - p : crée tous les répertoires intermédiaires, si nécessaire (--parent)
- Description : création du répertoire pour chaque nom passé en argument
- Exemple :
 

```
hans> mkdir repl
hans> mkdir repl/srepl
hans> mkdir -p a/b/c (équivalent mkdir a a/b a/b/c)
```

http://lildemur - Jean-Paul ARNSPACH 24 - Telecom-Physique

## La manipulation des répertoires

Principales commandes pour gérer répertoires **rmdir (ReMove DIRectory)**

Suppression d'un ou plusieurs répertoires

- Syntaxe :
 

```
rmdir [-p] nom_répertoire[...]
```
- Options :
  - p : supprime tous les répertoires intermédiaires, si vide (--parent)
- Description : suppression du répertoire vide pour chaque nom passé en argument
- Exemple :
 

```
hans> rmdir repl/srepl
hans> rmdir repl
hans> rmdir -p a/b/c (équivalent rmdir a/b/c a/b a)
```

http://lildemur - Jean-Paul ARNSPACH 24 - Telecom-Physique

## La manipulation des fichiers

Principales commandes de travail sur les fichiers **ls (liSt files)**

Affiche le contenu d'un répertoire

- Syntaxe :
 

```
ls -[aIF...] [nom_répertoire...]
```
- Description : sans argument, affiche la liste du répertoire courant
- Options :
  - l : liste détaillée des caractéristiques de chaque fichier
  - a : affiche également les fichiers commençant par le caractère . (fichiers cachés)
  - F : liste fichiers avec indication du type (/ : répertoire \* : exécutable, )
- Exemple :
 

```
hans> ls -F
a/ toto.txt prog*
```

http://lildemur - Jean-Paul ARNSPACH 24 - Telecom-Physique

## La manipulation des fichiers

Principales commandes de travail sur les fichiers **touch (TOUCH)**

Modification date d'accès au fichier

- Syntaxe :
 

```
touch fichier
```
- Description : création d'un fichier vide
- Exemple :
 

```
hans> touch f1
hans> ls -F
a/ toto.txt prog* f1
hans>
```

http://lildemur - Jean-Paul ARNSPACH 24 - Telecom-Physique

## La manipulation des fichiers

Principales commandes de travail sur les fichiers **cp (CoPy)**

Copie fichiers et répertoires

- Syntaxe :
 

```
cp -[ir] source destination
cp -[ir] source[...] repertoire_destination
```
- Description : Copie source vers destination ou multiple sources vers repertoire\_destination
- Options :
  - i : demande de confirmation en cas de copie sur fichier existant
  - r : copie récursive
- Exemple :
 

```
hans> cp f1 f2
hans> cp f1 repl
hans> cp -r repl save
```

http://lildemur - Jean-Paul ARNSPACH 24 - Telecom-Physique

### La manipulation des fichiers

Principales commandes de travail sur les fichiers

#### mv (MoVe list)

Renomme le nom d'un fichier

- Syntaxe :
 

```
mv [-i] source destination
mv [-i] source[...] repertoire_destination
```
- Description : renomme source par destination ou déplace source vers repertoire\_destination
- Options :
  - i : demande de confirmation en cas de copie sur fichier existant
- Exemple :
 

```
hans> mv f1 fichier1
hans> mv save sauvegarde
hans> mv fichier1 fichier2 repl
```

### La manipulation des fichiers

Principales commandes de travail sur les fichiers

#### rm (ReMove files)

Supprime un ou plusieurs fichiers

- Syntaxe :
 

```
rm [-ir] fichier[...]
```
- Description : supprime un ou plusieurs fichiers
- Options :
  - i : demande de confirmation
  - r : suppression récursive
- Exemple :
 

```
hans> rm -i fichier1
hans> rm -r repl
```

### La manipulation des fichiers

Principales commandes de travail sur les fichiers

#### less

Affiche le contenu d'un fichier

- Syntaxe :
 

```
less fichier
```
- Description : more ou less affiche le contenu d'un fichier page par page.
- Exemple :
 

```
hans> less /etc/passwd
affichage page par page du fichier
/etc/passwd
hans>
```

### La manipulation des fichiers

Principales commandes de travail sur les fichiers

#### ln

Permet d'établir un lien entre deux (plusieurs) noms et un fichier

- Syntaxe :
 

```
ln fichier_source fichier_cible
```
- Description : Permet de désigner un fichier par plusieurs nom, un seul espace disque
- Exemple :
 

```
hans> ln f2 f2_new
Le fichier porte deux noms mais n'occupe qu'un seul espace
disque.
hans> ls
f2 f2_new
```

### Permissions spéciales SUID/SGID/sticky bits

SUID (s)	Sgid (s)	Stb (t)	r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1	4	2	1
spécial			user			group			others		
r	w	x	r	w	x	r	w	x	r	w	x
r	w	S	r	w	S	r	w	t	r	w	t
		SUID			SGID			Sticky bit			

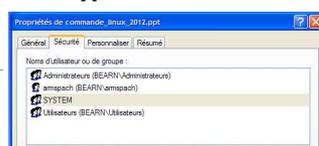
SUID : Set-UID = Set User identification  
 SGID : Set-GID = Set group identification

### Permissions sous Windows

#### types d'utilisateurs:

modes principaux de permission

- Contrôle total
- Parcours du dossier/exécuter le fichier
- Liste du dossier/lecture de données
- Attributs de lecture
- Lecture des attributs étendus
- Création de fichier/écriture de données
- Création de dossier/ajout de données
- Attributs d'écriture
- Écriture d'attributs étendus
- Suppression
- Autorisations de lecture
- Modification des autorisations
- Appropriation





## Modification des protections

- **Notation relative (aux droits existants)**

```
chmod [-R] [who]op[permission] fichiers
```

- op prend les valeurs + ou - et permet d'ajouter ou supprimer un droit
- Exemple :
 

```
hans> chmod -R o-rwx /home/hans
```

 enlève toutes les permissions d'accès des fichiers du rép. personnel de hans (et des sous-rép.) aux autres (others).
 

```
hans> chmod -R g-rwx,o-rwx /home/hans
```

 enlève toutes les permissions d'accès des fichiers du rép. personnel de hans (et des sous-rép.), à tous sauf au propriétaire, c'est-à-dire hans.

http://utilisateur - 3ans-Paul ASSOCIATION 24/08/2015 12:00:00



## Modification des protections

- **Notation absolue**

```
chmod [-R] [who]op[permission] fichiers
```

- Pour chaque type d'utilisateurs, elle permet de fixer les nouveaux droits qui remplacent les anciens. Si un type n'est pas présent, ses anciens droits s'appliquent.
- Exemple :
 

```
hans> chmod u=rwx,g=rw,o=r fichier
```

 remplace les permissions précédentes de fichier, en les fixant à `rw-rw-r--`.
 

```
hans> chmod u=rwx,g=r fichier
```

 fixe pour fichier les permissions à `rw-r--???` en ne changeant pas les permissions précédentes du groupe

http://utilisateur - 3ans-Paul ASSOCIATION 24/08/2015 12:00:00



## Protection par défaut

### umask

Permet d'établir la protection par défaut.

- Syntaxe :
 

```
umask [-S] [who]op[permission]
```
- Description :
 Permet d'établir la protection par défaut. L'option -S permet d'afficher les protections par défaut de l'utilisateur en mode symbolique.
- Exemple :
 

```
hans> umask u=rwx,g=rwx,o=r
```

```
hans> umask -S
```

```
u=rwx,g=rwx,o=r
```

```
hans> touch f1
```

```
hans> ls -l f1
```

```
-rwxrwxrwx 1 hans hans 0 Sep 21 12:55 f1
```

http://utilisateur - 3ans-Paul ASSOCIATION 24/08/2015 12:00:00



## Protection par défaut

### chgrp

Changement de groupe.

- Syntaxe :
 

```
chgrp [-R] group fichiers[...]
```
- Description :
 Permet le changement de groupe pour les fichiers et répertoires. L'option -R permet la récursivité de la commande.
- Exemple :
 

```
/home/hans/> ls -l f1
```

```
-rw-rw-r-- 1 hans hans 0 Sep 21 12:55 f1
```

```
/home/hans> chgrp etudiant f1
```

```
/home/hans> ls -l f1
```

```
-rw-rw-r-- 1 hans etudiant 0 Sep 21 12:55 f1
```

http://utilisateur - 3ans-Paul ASSOCIATION 24/08/2015 12:00:00



## Les commandes Unix

Syntaxe d'une commande Unix  
L'entrée et les sortie standards  
La redirection des entrées-sorties  
La commande cat  
Les tubes, les filtres  
La substitution de commande

http://utilisateur - 3ans-Paul ASSOCIATION 24/08/2015 12:00:00



## Les filtres

### less (more)

Affiche le contenu d'un fichier à l'écran

- Syntaxe :
 

```
less [-option] [fichier...]
```
- Options :
- Description :
 filtre qui affiche page par page le contenu d'un fichier ou l'entrée standard
- Exemple :
 

```
hans> less /etc/passwd
```

 Affiche le contenu de passwd page par page
 

```
hans>
```

http://utilisateur - 3ans-Paul ASSOCIATION 24/08/2015 12:00:00





## Les filtres

### sort

Lie, trie le contenu d'un fichier et affiche le résultat à l'écran

- Syntaxe :  
**sort [-options] [+pos1] [-pos2][fichier...]**
- Options :
  - o file résultat du tri dans le fichier file
  - tx choix du séparateur de champs (x) (espace ou tab par défaut)
  - +pos1 exclure les champs de 1 à c1
  - pos2 exclure les champs à partir de c2
- Description :  
filtre qui lie et trie le contenu d'un fichier ou l'entrée standard en s'appuyant sur la notion de champs par ligne.
- Exemple :  
hans> **sort -t: +2, -3 -o passwd.tri /etc/passwd**  
Tri le contenu de passwd sur le numéro UID (3ième champs) et met le résultat dans passwd.tri du répertoire courant.  
hans>

http://lftbotm - Jean-Paul ARNSPACH 24 EOQS 2011-2012



## les caractères spéciaux

- / \ - ; \$ < > | ~ & ! ^ \* ? ( ) " ' ` [ ] { } # <espace> <new-line>
- ont une signification spéciale pour le shell
- ne jamais les utiliser dans un nom de fichier ou un paramètre d'une commande unix
- si besoin est, les neutraliser

http://lftbotm - Jean-Paul ARNSPACH 24 EOQS 2011-2012



## L'interpréteur de commandes : bash

http://lftbotm - Jean-Paul ARNSPACH 24 EOQS 2011-2012



## L'interpréteur de commande (shell)

- programme qui sert d'intermédiaire entre l'utilisateur (saisie de la commande) et le système d'exploitation (exécute la commande).
  - Hors système (il y en a plusieurs, à chacun son shell)
  - Configurable (fichiers d'initialisation)
  - Ensemble de fonctions internes et de structures de contrôle logique
  - Langage de programmation interprété (scripts)
- Sa tâche essentielle est :
  - Interaction avec l'utilisateur pour la saisie de la commande,
  - Interpréter la commande saisie
  - Lance le programme avec les paramètres

http://lftbotm - Jean-Paul ARNSPACH 24 EOQS 2011-2012



## Le bash interactif

- Historique des commandes avec possibilité de rappel.
- Édition de la ligne de commande
- Recherche et complétement du nom d'une commande
- Recherche et complétement des noms de fichiers
- Substitution du caractère tilde ~
- copier-coller

http://lftbotm - Jean-Paul ARNSPACH 24 EOQS 2011-2012



## L'historique des commandes

- Mécanisme d'historique permet de rechercher puis modifier une ancienne commande.
- La commande **history** permet de visualiser les anciennes commandes
- Recherche d'une commande
  - ↩ recherche commande précédente dans l'historique
  - > recherche commande suivante dans l'historique
  - <ctrl-R> motif recherche incrémentale en arrière ; peut être répétée pour recherche avec même motif

http://lftbotm - Jean-Paul ARNSPACH 24 EOQS 2011-2012



## L'édition de la ligne de commande

- Après une saisie ou une recherche dans l'historique vous pouvez :
  - Effacer le dernier caractère saisi avec la touche backspace
  - Déplacer le curseur avec les flèches droite ou gauche
  - Insérer un caractère
  - Supprimer le caractère suivant avec la touche del
  - <ctrl-k> suppression du texte du curseur jusqu'à la fin de la ligne.
  - <ctrl-u> suppression du texte du curseur jusqu'au début de la ligne

http://tuxpenguin.com/Paul\_ABRASCHU/24\_EOQS\_2011-2012



## Recherche et complétement du nom d'une commande

- Il est possible de compléter le nom d'une commande.
  - La touche <tab>
  - Utilisation du PATH, des alias et des fonctions
  - Si ambiguïté (plusieurs commandes commencent par le même préfixe) alors bip
  - Sinon, le bash complètera totalement ou partiellement (suive d'un bip) le nom de la commande
  - Liste de toutes les commandes commençant par un préfixe : <tab><tab>

http://tuxpenguin.com/Paul\_ABRASCHU/24\_EOQS\_2011-2012



## Recherche et complétement d'un nom de fichier

- Il est possible de compléter le nom d'un fichier.
  - La touche <tab>
  - Si ambiguïté (plusieurs fichiers commencent par le même préfixe) alors bip
  - Sinon, le bash complètera totalement ou partiellement (suive d'un bip) le nom du fichier
  - Liste de toutes les fichiers commençant par un préfixe : <tab><tab>

http://tuxpenguin.com/Paul\_ABRASCHU/24\_EOQS\_2011-2012



## Copier-coller

- Utilisation du copier-coller avec une souris.
  - La sélection avec le bouton gauche de la souris (copier)
  - Le clic-milieu permet de coller un texte sélectionné.

http://tuxpenguin.com/Paul\_ABRASCHU/24\_EOQS\_2011-2012



## Gestion de l'espace disque, sauvegarde

Organisation  
Inodes  
Commandes ln, mv, rm

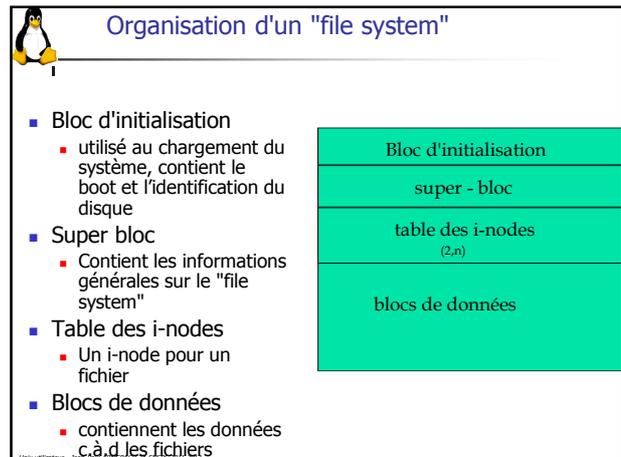
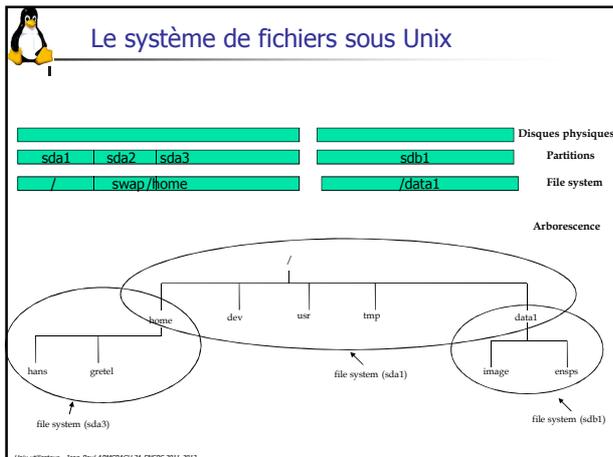
http://tuxpenguin.com/Paul\_ABRASCHU/24\_EOQS\_2011-2012



## Le système de fichiers sous Unix

- L'arborescence, organisation logique, acyclique, est basée sur une organisation physique constituée de un ou plusieurs disques.
- L'espace disque physique est découpé en partition (disque logique)
- Sur chaque disque logique il y a un « file system »
- Sous Unix, l'utilisateur ne voit pas les disques ni les partitions.
- L'arborescence des fichiers sous Unix est construite à partir de plusieurs « file system », le **montage**.

http://tuxpenguin.com/Paul\_ABRASCHU/24\_EOQS\_2011-2012



### Gestion de l'occupation disque : du

`du [-s][-a][-H] noms`

- Affiche le nombre de blocs occupés par les fichiers et répertoires noms .
  - -s : résultats par noms
  - -a : détails de noms et du contenu de noms (récuratif)
  - -H : exprimé plus clairement (K et G)

Exemples:  
 hans> `cd ; du -sH *`  
 Indique l'occupation disque de chacun de vos répertoires contenus dans votre répertoire d'accueil.  
 hans>

### La sauvegarde : la commande tar

- Cette commande permet d'archiver des fichiers ou une arborescence de fichiers, c-à-d de les regrouper dans un seul fichier
- Pour créer une nouvelle archive :  
`tar -cvf nom_archive.tar repertoire`
- Pour afficher le contenu d'une archive :  
`tar -tvf nom_archive.tar`
- Pour extraire les fichiers archivés :  
`tar -xvf nom_archive.tar`

Les fichiers sont créés à partir du répertoire courant s'ils sont stockés sur l'archive en chemin relatif.

### La compression

- **compress** : extension .Z
- **gzip** : extension .gz
- **bgzip2** : extension bz2

Exemples:  
 hans> `compress fichier?`  
 hans> `gzip file?`  
 hans> `ls`  
 fichier1.Z fichier2.Z file1.gz file2.gz  
 hans> `gunzip file1.gz file2.gz`

### La synchronisation : rsync

**rsync** permet de synchroniser le contenu de deux fichiers ou répertoires.  
**rsync** est similaire à la commande `cp` mais ne transfère que les différences entre fichiers.  
**rsync** est possible entre deux ordinateurs.  
**rsync** peut être assimilée à une commande permettant de réaliser un miroir entre deux disques.

`rsync [options] source destination`

- Options :
  - -a : --archive option complète (-rlptgoD)
  - --delete : supprime les fichiers destination n'existant plus dans la source
  - -v : affiche le nom et répertoire traités

Exemples:  
 hans> `rsync -av cours/ /data1/cours.sav`  
 hans>

 **La gestion des processus**

Notions théorique sur les processus  
Exécution d'une commande  
La gestion des processus, le job control

http://ltdl.univ-lille.fr/~jean-paul.arnould/24\_0805/2011-2012

 **Introduction : Les processus**

- Sous Unix toute activité s'exécute sous forme de processus.
- Il peut donc y avoir simultanément plusieurs processus du même programme en exécution en mémoire.
- Le noyau Linux lance, gère les processus et contrôle leurs échanges avec les périphériques.

http://ltdl.univ-lille.fr/~jean-paul.arnould/24\_0805/2011-2012

 **La gestion des processus : la commande ps**

**ps**

donne un ensemble de renseignements sur les processus en court

- Syntaxe : `ps [options]`
- Options :
  - -e : affiche des renseignements sur tous les processus en cours
  - -f : (full) information complète
  - --forest : affiche les commandes en hiérarchie
  - -u user : sélectionne les processus de user

http://ltdl.univ-lille.fr/~jean-paul.arnould/24\_0805/2011-2012

 **La gestion des processus : kill**

**kill**

Envoie un signal à un processus

- Syntaxe : `kill -signal pid`
- Options :
  - -signal : correspond au signal envoyé au processus
    - SIGKILL SIGnal KILL : tuer le processus (-9)
    - SIGSTOP SIGnal STOP : stopper le processus
- Description : La commande kill envoie un signal à un processus. Essentiellement utilisé pour tuer (-9) un processus. `kill -l` liste des signaux.

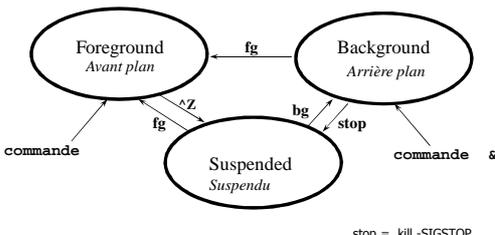
Exemples :

```
hans> kill -9 1635 (kill -SIGKILL 1635)
Cette commande tue le processus 1635. Le numéro PID est obtenu avec la commande ps
```

http://ltdl.univ-lille.fr/~jean-paul.arnould/24\_0805/2011-2012

 **L'état d'un job : Synthèse**

Un job peut se trouver dans 3 états



```

graph TD
    C[commande] --> F((Foreground  
Avant plan))
    C --> B((Background  
Arrière plan))
    C --> S((Suspended  
Suspendu))
    B -- fg --> F
    S -- fg --> F
    S -- bg --> B
    B -- stop --> S
    
```

stop = kill -SIGSTOP

http://ltdl.univ-lille.fr/~jean-paul.arnould/24\_0805/2011-2012

 **Le job control : synthèse**

Action	Etat initial du processus	Etat final du processus
hans> commande		processus en avant plan
hans> commande &		processus en arrière plan
hans> fg numéro_job	processus en arrière plan	processus en avant plan
<ctrl-z>	processus en avant plan	processus stoppé (suspendu)
hans> fg numéro_job	processus stoppé	processus en avant plan
hans> bg numéro_job	processus stoppé	processus en arrière plan
• xstra> KILL -STOP numéro_job • tentative lecture sur le terminal (si stty testop)	processus en arrière plan	processus stoppé

http://ltdl.univ-lille.fr/~jean-paul.arnould/24\_0805/2011-2012