Module

Système d'exploitation

1ère année TIC

Enseignante :
Sahar Ben Yaala

La gestion de processus

2019-2020

La gestion des processus

- Notion de processus
- Etats d'un processus
- Commutation entre processus
- Création d'un processus
- Terminaison d'un processus
- Processus sous UNIX
  - Commandes de base
  - Appels Système (Fork , exec,…)
- Ordonnancement des processus

La notion de processus

- processus: l'entité qui correspond à l'exécution d'un programme.
- Concrètement, un processus est défini par :

- un espace mémoire contenant le code, les données et la pile d'instruction.

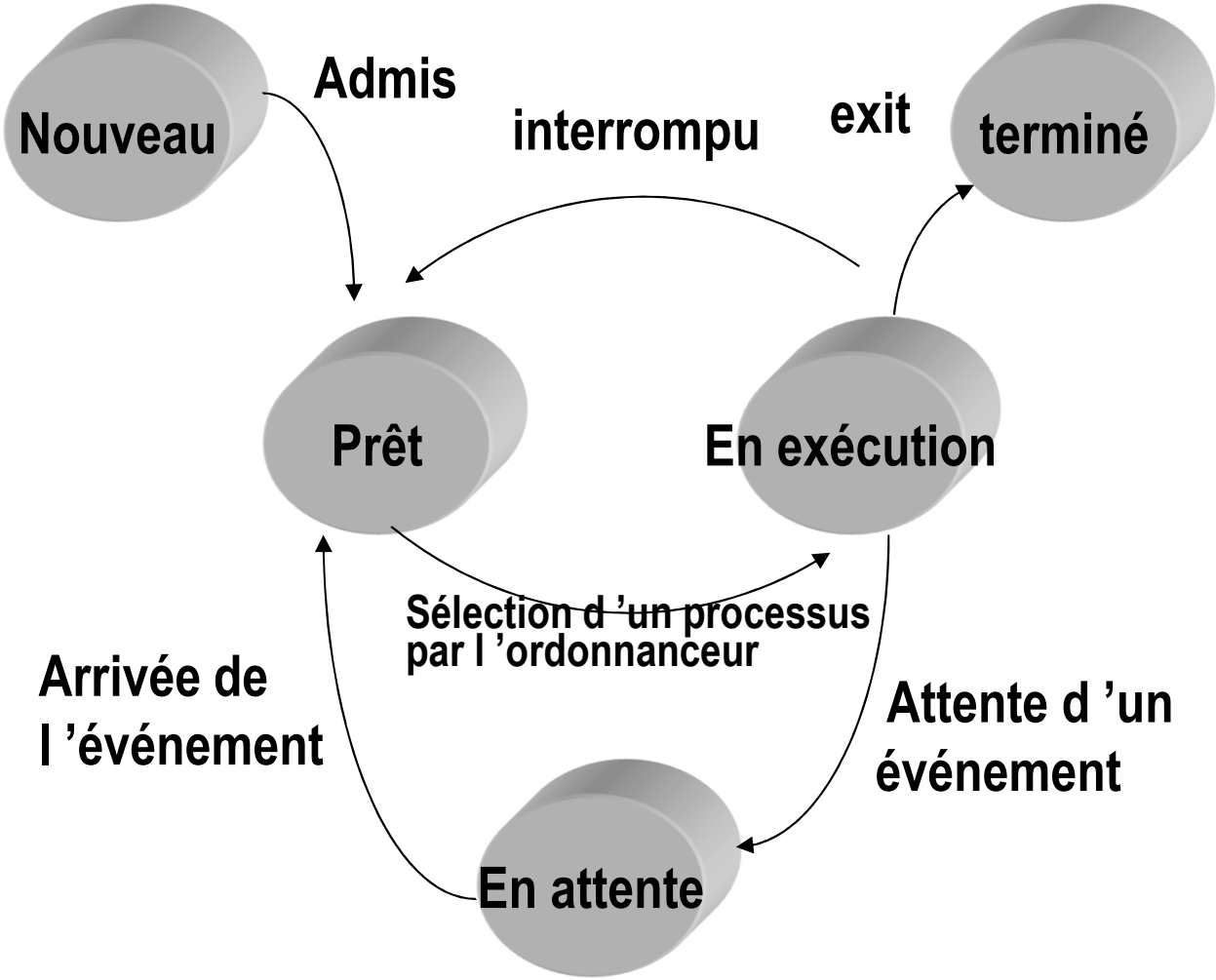
- un compteur ordinal (zone mémoire qui pointe sur l'instruction en cours).

- un ensemble de registres (zones mémoire à accès rapide situées au niveau du processeur).

États d'un processus

- Nouveau
  - le processus est en cours de création
- En exécution (Running)
  - les instructions sont en cours d 'exécution
- En attente (Sleep)
  - le processus attend qu'un événement se produise
    - événement : un signal , terminaison d'une E/S, ...
- Prêt (Ready)
  - le processus attend d'être affecté à une UC
- Terminé
  - le processus a fini l'exécution.

États d'un processus



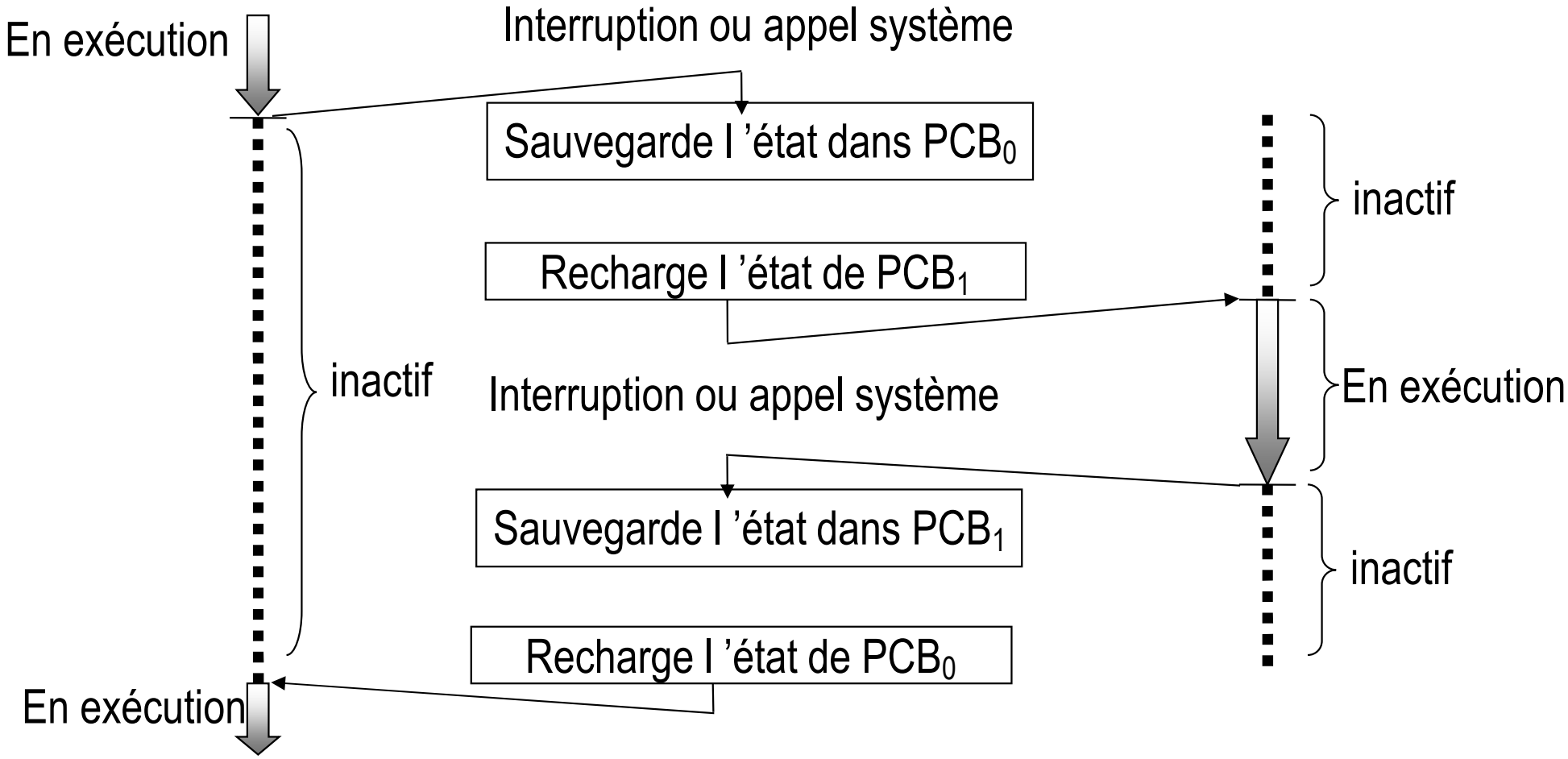
Bloc de contrôle de processus

- Chaque processus est représenté dans le SE par un bloc de contrôle (Process Control Bloc : PCB)

- Composants d'un PCB
  - l'état du processus : nouveau, prêt, en exécution ,
  - le compteur d'instructions : prochaine instruction à exécuter
  - les registres de l'UC
  - informations sur l'ordonnancement des processus
  - informations sur la gestion mémoire
  - informations de comptabilisation
  - informations sur l'état des E/S
  - ex : liste des fichiers ouverts

Commutation entre processus

- Processus P0 SE Processus P1



Création d'un processus

Il existe essentiellement quatre évènements provoquant la création d'un processus:

1. Initialisation du système (processus init en UNIX)
2. Exécution d'un appel de création de processus par un autre processus en cour d'exécution (fork() en UNIX)
3. Requête utilisateur sollicitant la création d'un nouveau processus
4. Initiation d'un travail

La fin d'un processus

L'arrêt d'un processus est causé par diverses raisons:

- Arrêt normal (volontaire): fin d'exécution de la tâche affectée (Terminaison).
- Arrêt pour erreur (volontaire?): division par 0
- Arrêt pour erreur fatale (involontaire): violation accès mémoire, …
- Le processus est arrêté par un autre processus (involontaire/externe): fin de tâche d'un processus (cas de Windows), commande kill en LINUX.

Processus sous Linux

+ Commandes de base

+ Informations liées aux processus

UID	PID	PPID	C	STIME	TTY	TIME	CMD
user	6937	6912	0	11:05	Pts/1 6	00:00:00	less Mypasswd

UID : nom de propriétaire qui a lancé le processus (user, root, ...).
PID : numéro du processus.
PPID : numéro du processus père (créateur).
C : facteur de priorité.
STIME : date de lancement du processus.
TTY : numéro du terminal qui exécute le processus.
TIME : la durée de traitement du processus
CMD : nom de la commande exécutée.

+ Commandes de gestion de processus

- Commande ps

Options	Fonctions
Sans options	processus en exécution.
-u **user**	processus en exécution pour l'utilisateur **user**.
-ef	informations complètes concernant les processus en cours d'exécution
-x	processus actifs de l'utilisateur courant
-ax	processus de tous les utilisateurs
-p **PID**	informations sur le processus **PID**
-l	afficher d'informations assez complète
-c	afficher les commandes exécutées.

Commandes de gestion de processus

- Commande ps

- Exemple

Affichage des informations suivantes sur le processus : (PID, TTY, TIMe & COMD)

```
$ ps
   PID TTY      TIME COMD
  1664 pts/4    0:06 csh
  2081 pts/4    0:00 ps
```

```
$ ps -ef
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   Dec 20 ?       0:17 sched
    root     1     0  0   Dec 20 ?       0:00 /etc/init -
    root     2     0  0   Dec 20 ?       0:00 pageout
    root     3     0  0   Dec 20 ?       4:20 fsflush
    root   374   367  0   Dec 20 ?       0:00 /usr/lib/saf/ttymon
```

Commandes de gestion de processus

- commande ps –l :

Liste des informations complètes

```
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000   6912   6911  0  80   0 -  1775 wait   pts/16   00:00:02 bash
0 T  1000   6936   6912  0  80   0 -  1128 signal pts/16   00:00:00 less
0 T  1000   6973   6912  0  80   0 -  1128 signal pts/16   00:00:00 less
0 T  1000   6982   6912  0  80   0 -  1153 signal pts/16   00:00:00 more
0 R  1000  10740   6912  0  80   0 -  1267 -      pts/16   00:00:00 ps
```

- Explication :

D sommeil ininterruptible
R Actif ou prêt (dans la file)
S Sommeil interruptible (attente d'un événement)
T Stoppé (par un signal)
X Mort
Z Deficient (''zombie'') processus, terminé mais données non recup par parent

+Commandes de gestion de processus

- Commande Top :

Options	Fonctions
Sans options	Table des processus qui se met à jour d'une manière continuelle.
-d	Configuration de délai de rafraichissement.
-n	Affichage des processus en arrière plan.

Commandes de gestion de processus

- Priorité du processus
  - Modification de la priorité : commande **nice**
  - Valeur de la priorité :

    - Si simple utilisateur : entre 0 et 19
    - Si Super utilisateur : entre -20 et 19
    - Valeur par défaut = 0
    - Valeur de la priorité la plus basse = 19
    - Valeur de la priorité la plus haute = -20

  - Modification de la valeur de **nice** à l'aide de la commande **renice**

Commandes de gestion de processus

- Priorité du processus

Commande	Exemple	Explication
nice -priorité commande	nice -19 **find**	commande **find** avec la plus basse priorité.
Renice priorité PID	renice −20 2535	-Attribuer la priorité la plus haute au processus 2535. -Que le root peut attribuer cette priorité.

+ Signaux

❑ Définition

Moyens utilisés pour communiquer avec les processus.

❑Exemple

Commande	Fonctionnement
Ctrl+Z	Un signal numéro **19** (**SIGSTOP**) est envoyé au processus en cours d'exécution. Ce qui stoppe son traitement.
déconnexion	envoi d'un **SIGHUP** (signal 1) à tous les processus
Ctrl+C	Envoi d'un SIGINT) (signal 2) au processus courant.

- ❑ Il y'a 64 signaux avec des identifiants différents
  - 0 : seul signal qui n'a pas de nom
  - 1 à 31 : signaux classiques
  - 32 à 63 : signaux temps réels

+ Signaux

Les signaux Classiques

SIGHUP (1) Connexion physique interrompue ou terminaison du processus leader de la session
SIGINT (2) frappe du caractère intr (interruption clavier : Ctrl C)
SIGQUIT (3) frappe du caractère quit (interruption clavier avec sauvegarde de l'image mémoire dans le fichier de nom core) sur le clavier du terminal de contrôle (Ctrl \);
SIGILL (4) Instruction illégale
SIGTRAP (5) Point d'arrêt pour le debug mode (non POSIX)
SIGIOT/SIGABRT (6) Terminaison anormale
SIGBUS (7) Erreur de bus
SIGFPE (8) Erreur arithmétique
SIGKILL (9) signal de terminaison non déroutable
SIGUSR1 (10) Signal 1 défini par l'utilisateur
SIGSEGV (11) Adressage mémoire invalide
SIGUSR2 (12) Signal 2 défini par l'utilisateur
SIGPIPE (13) Écriture sur un tube sans lecteur
SIGALRM (14) Alarme
SIGTERM (15) Signal de terminaison, il est envoyé à tous les processus actifs par le programme shutdown, qui permet d'arrêter proprement un système UNIX. Terminaison normale.
SIGSTKFLT (16) Erreur de pile du coprocesseur
SIGCHLD (17) Terminaison d'un fils.
SIGCONT (18) Reprise du processus.
SIGSTOP (19) Suspension du processus (non déroutable).
SIGTSTP (20) Émission vers le terminal du caractère de suspension (Ctrl Z).
SIGTTIN (21) Lecture du terminal pour un processus d'arrière-plan.
SIGTTOU (22) Écriture vers le terminal pour un processus d'arrière-plan.
SIGURG (23) Données urgentes sur une socket
SIGXCPU (24) Limite de temps CPU dépassé
SIGXFSZ (25) Limite de taille de fichier dépassé
SIGVTALARM (26) Alarme virtuelle (non POSIX)
SIGPROF (27) Alarme du profileur (non POSIX)
SIGWINCH (28) Fenêtre redimensionnée (non POSIX)
SIGIO (29) Arrivée de caractère à lire (non POSIX)
SIGPOLL (30) Equivalent à SIGIO (non POSIX)
SIGPWR (31) Chute d'alimentation (non POSIX)
SIGUNUSED (32) Non utilisé

+ Signaux

Les actions par défaut liées aux différents signaux

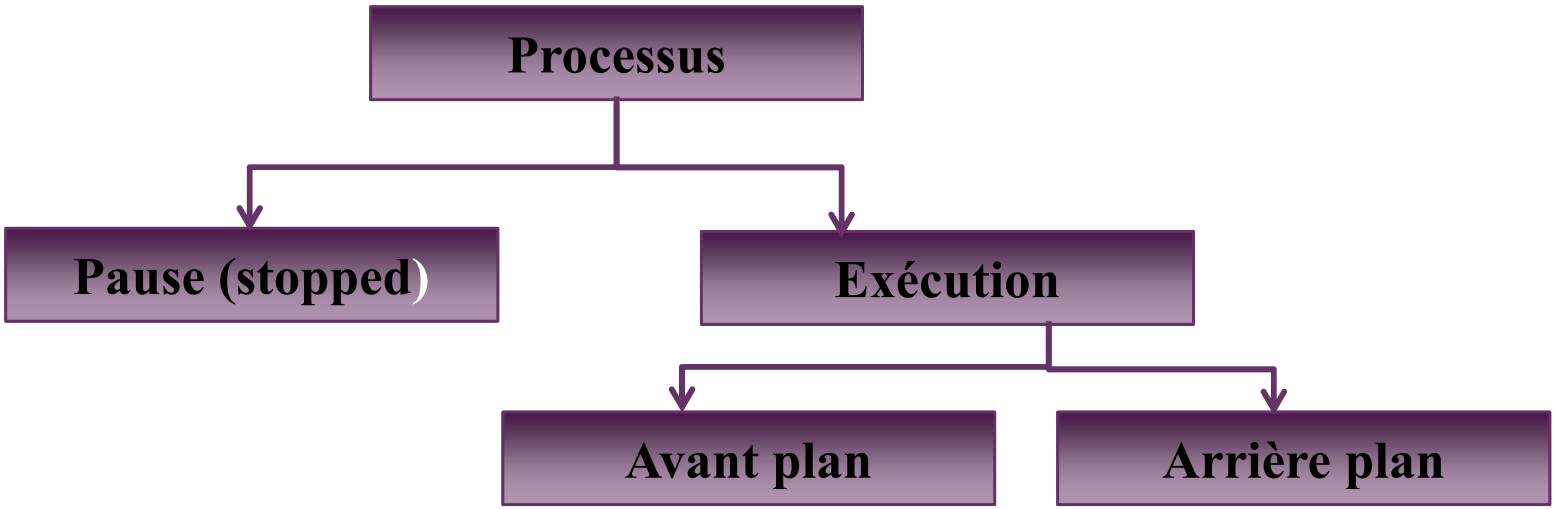
Actions par défaut	Nom du signal
Fin du process	SIGHUP, SIGINT, SIGBUS, SIGKILL, SIGUSR1, SIGUSR2, SIGPIPE, SIGALRM, SIGTERM, SIGSTKFLT, SIGXCOU, SIGXFSZ, SIGVTALRM, SIGPROF, SIGIO, SIGPOLL, SIGPWR, SIGUNUSED
Fin du process et création core	SIGQUIT, SIGILL, SIGTRAP, SIGABRT, SIGIOT, SIGFPE, SIGSEGV
Signal ignoré	SIGCHLD, SIGURG, SIGWINCH
Processus stoppé	SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU
Processus redémarré	SIGCONT

+ Signaux

❑Gestion de signaux : Kill

Commande	Exemple	Fonctionnement
kill	-l	Affichage de tous les signaux (64)
kill –numéro PID	Kill -9 6936	Tuer le processus de pid **6936** (envoi d'un SIGKILL)
Kill –NomSignal PID	Kill – SIGKILL 6936	Tuer le processus de pid **6936**

Etats d'un processus



+ Etats d'un processus

- Possibilité de lancer une commande en arrière plan :

Syntaxe : **commande &**

- Exemple :

```
~$ less mypasswd &

[3] 11210
```

[3] : est le numéro de processus
112010 : son PID

- La commande jobs

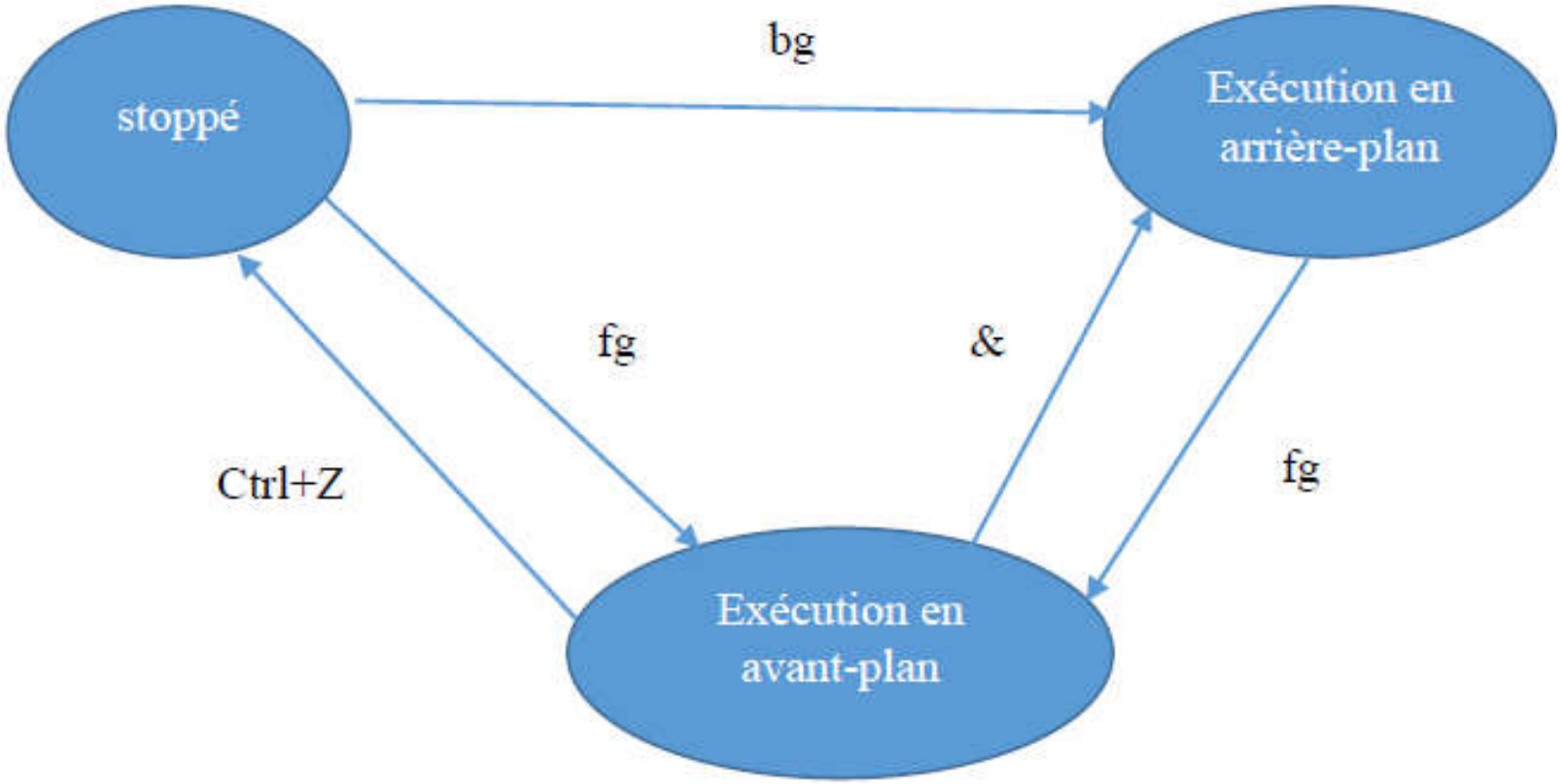
Indique la liste des processus en arrière plan.

+ Etats d'un processus

- Passage de l'avant plan vers l'arrière plan
  - Passage en mode pause : **Ctrl+Z**
  - Commande **bg** (background)
- Passage de l'arrière plan vers l'avant plan

1. Si un seul processus est exécuté en arrière plan
   - Commande **fg** (Foreground)
2. Si plusieurs processus sont exécutés sen arrière plan
   - **jobs** : liste des processus en arrière plan
   - **fg % i** : ramener le ième en avant plan

+ Etats d'un processus



Mauvaise Terminaison d'un processus Unix

- Si le processus père termine son exécution avant son fils, ce dernier devient un processus **orphelin**, qui sera attaché au processus initiateur (init).
- Si le processus fils meurt avant que son père ne se termine, celui-ci devient un processus **zombie**.