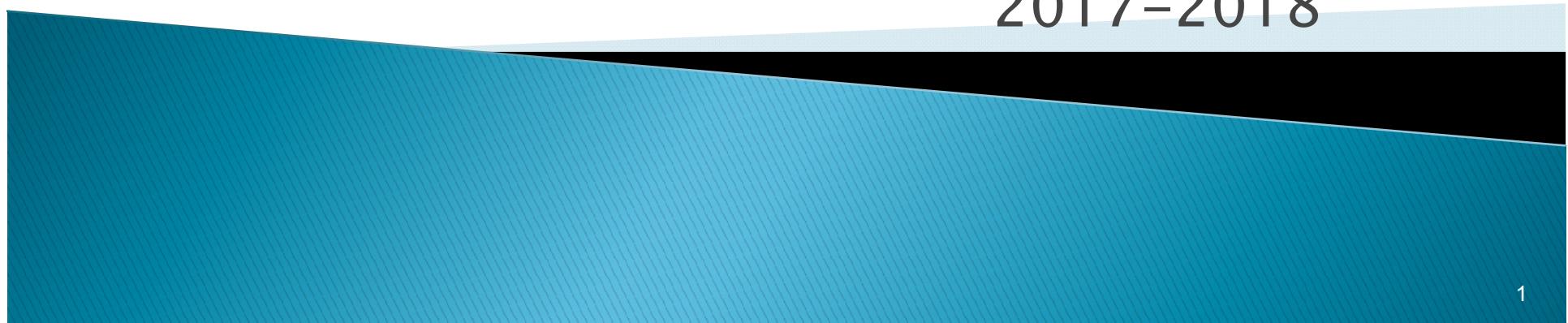


Ecole Nationale des Sciences Appliquées
Khouribga

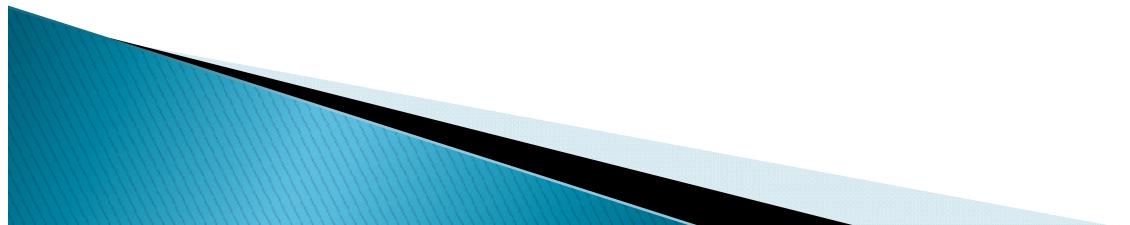
GESTION DES PROCESSUS

Med AMNAI
2017-2018



Introduction

1. Caractéristiques d'un processus
2. États des processus
3. Visualiser les processus
4. Arrêter un processus
5. Changer la priorité d'un processus
6. Lancer un processus en tâche de fond
7. Utilisation de la mémoire



Concept de processus

- ▶ Un **processus** est un programme en cours d'exécution;
- ▶ La plupart des commandes **exécutent un programme**, et donc lancent un processus;
- ▶ Le programme de démarrage correspond à un **processus présent pendant toute la session**;
- ▶ Des dizaines ou centaines de processus sont en fonctionnement à tout moment.

Concept de processus

Caractéristiques d'un processus

- ▶ identifié par un numéro entier
- ▶ code du programme en cours d'exécution
- ▶ données traitées par ce code
- ▶ identification des fichiers en cours de traitement et leur état
- ▶ répertoire courant
- ▶ identité du propriétaire du processus
- ▶ terminal associé
- ▶ etc.

Caractéristiques d'un processus : PID

- Le système attribue à chaque processus un numéro d'identification appelé **PID (Processus Identification)**;
- Un processus peut **créer** un autre processus, il devient donc un processus **parent** ou père, et le nouveau processus, un processus **enfant**.
- Le processus **enfant** est identifié par son **PID**, et le processus **père** par son numéro de processus appelé **PPID (Parent Processus Identification)** ;
- Tous les processus sont identifiés par leur **PID**, mais aussi par le **PPID** du processus qui la créé, car tous les processus ont été créés par un autre processus.
- Le seul qui ne suit pas cette règle est le **premier processus** lancé : le processus **init** qui n'a pas de père et qui a pour **PID 1**.

Caractéristiques d'un processus : Etats

Un processus devient tour à tour **actif/inactif** tout au long de sa vie ; il passe donc par différents états :

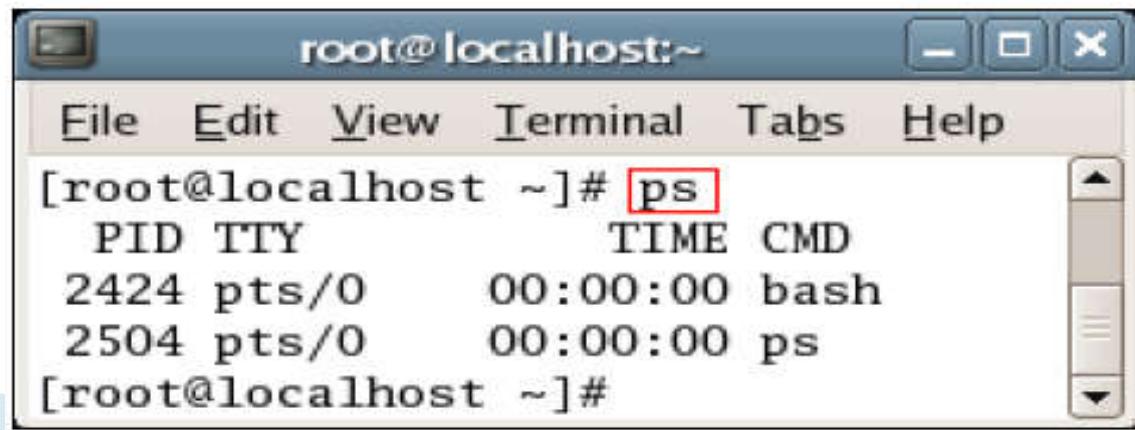
- **Élu** : le CPU **exécute** le code de ce processus.
- **Prêt** : il **dispose de toutes les ressources nécessaires** à son fonctionnement à **l'exception du CPU**. Il est dans une file d'attente associée au CPU.
- **Bloqué** : il est **en attente d'une ressource** autre que le CPU ou d'un événement.
- **Zombie** : le processus a **terminé son exécution** et est prêt à mourir ; à ce stade, l'ordonnanceur a détecté l'arrêt de la tâche mais le père du processus n'a pas encore pris en compte sa mort ; il existe donc toujours une entrée pour le processus dans la table des processus.

Caractéristiques d'un processus : Etats

- ▶ Les processus sont dans différents états :
 - **D** : **dormant non interruptible** ;
 - **R** : **en train de s'exécuter** ;
 - **S** : **endormi mais interruptible** ;
 - **T** : **arrêté** ;
 - **Z** : **zombi**.
- ▶ Tout processus est lancé par un processus père :
 - **Arbre généalogique** des processus
 - **Propriétaire réel** (utilisateur qui l'a lancé)
 - **Propriétaire effectif**

Visualiser les processus : ps

- **ps** permet de visualiser les processus du système.
- Elle **donne** une **image simplifiée des structures du noyau qui gèrent les processus.**
- Elle extrait toutes ses informations du pseudo-système de fichier **/proc** dans lequel le noyau place **toutes les informations concernant l'état du système en cours d'utilisation.**
- **Par défaut**, cette commande affiche uniquement les **processus lancés par l'utilisateur** à partir du **terminal**.

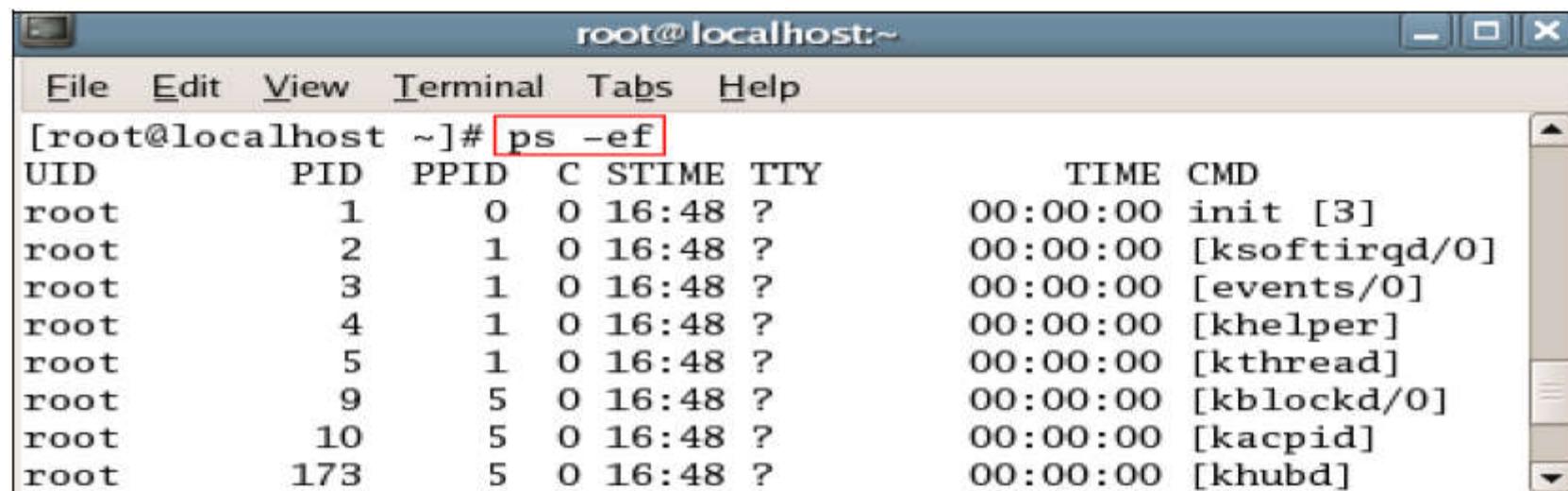


The screenshot shows a terminal window titled "root@localhost:~". The window has a menu bar with File, Edit, View, Terminal, Tabs, and Help. The main area displays the output of the "ps" command:

```
[root@localhost ~]# ps
  PID TTY          TIME CMD
 2424 pts/0        00:00:00 bash
 2504 pts/0        00:00:00 ps
[root@localhost ~]#
```

Visualiser les processus : ps

Les options les plus intéressantes sont **-e (affichage de tous les processus)** et **-f (affichage détaillée)**. La commande **ps -ef** affiche les informations suivantes:



The screenshot shows a terminal window titled "root@localhost:~". The window contains the command "ps -ef" highlighted with a red box. The output of the command is displayed in a table format:

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	16:48	?	00:00:00	init [3]
root	2	1	0	16:48	?	00:00:00	[ksoftirqd/0]
root	3	1	0	16:48	?	00:00:00	[events/0]
root	4	1	0	16:48	?	00:00:00	[khelper]
root	5	1	0	16:48	?	00:00:00	[kthread]
root	9	5	0	16:48	?	00:00:00	[kblockd/0]
root	10	5	0	16:48	?	00:00:00	[kacpid]
root	173	5	0	16:48	?	00:00:00	[khubd]

RQ: exemple **ps -Al** permet une sortie assez riche, en faisant un **man ps**, vous aurez l'éventail de tous les paramètres possibles.

Visualiser les processus : ps

La signification des différentes colonnes est la suivante:

UID	nom de l'utilisateur qui a lancé le processus
PID	correspond au numéro du processus
PPID	correspond au numéro du processus parent
C	au facteur de priorité : plus la valeur est grande, plus le processus est prioritaire
STIME	correspond à l'heure de lancement du processus
TTY	correspond au nom du terminal
TIME	correspond à la durée de traitement du processus
CMD	correspond au nom du processus.

RQ : Certains processus sont **permanents (PPID=1)**, c'est à dire qu'ils sont **lancés au démarrage du système et arrêtés uniquement à l'arrêt du système**. On appelle ces processus des **démons**.

Visualiser les processus : EXEMPLES

- Pour voir les processus d'un seul **utilisateur**, vous pouvez taper :

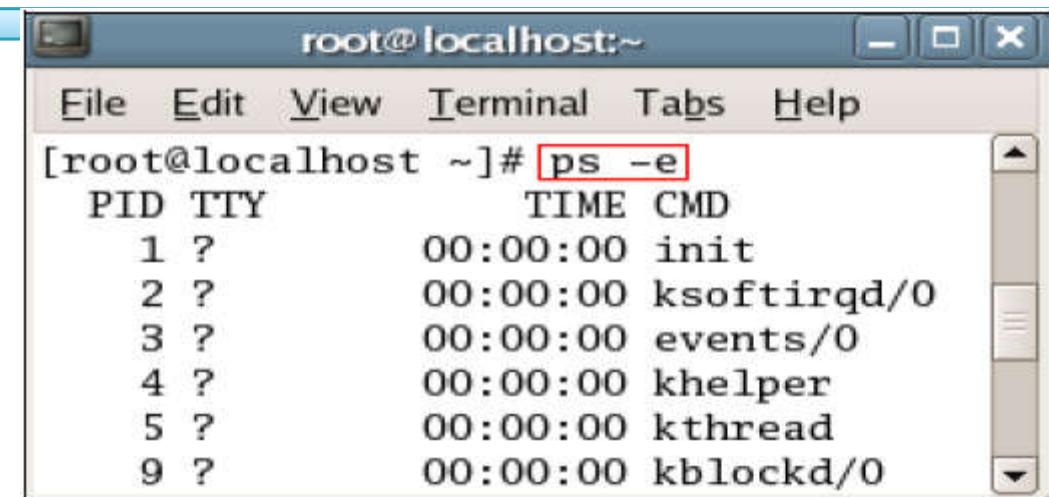
```
ps -u root
```

- La commande **ps ax** affiche une **liste des processus système en cours d'exécution**, y compris les processus appartenant à d'autres utilisateurs.
- Pour afficher le **propriétaire des processus**, utilisez la commande **ps aux**.
- Utiliser la commande **ps** combinée à la commande **grep** pour savoir **si une commande est en cours d'exécution**. Par **exemple**, pour déterminer si **vi** est encore en cours d'utilisation, utilisez la commande :

```
ps ax | grep vi
```

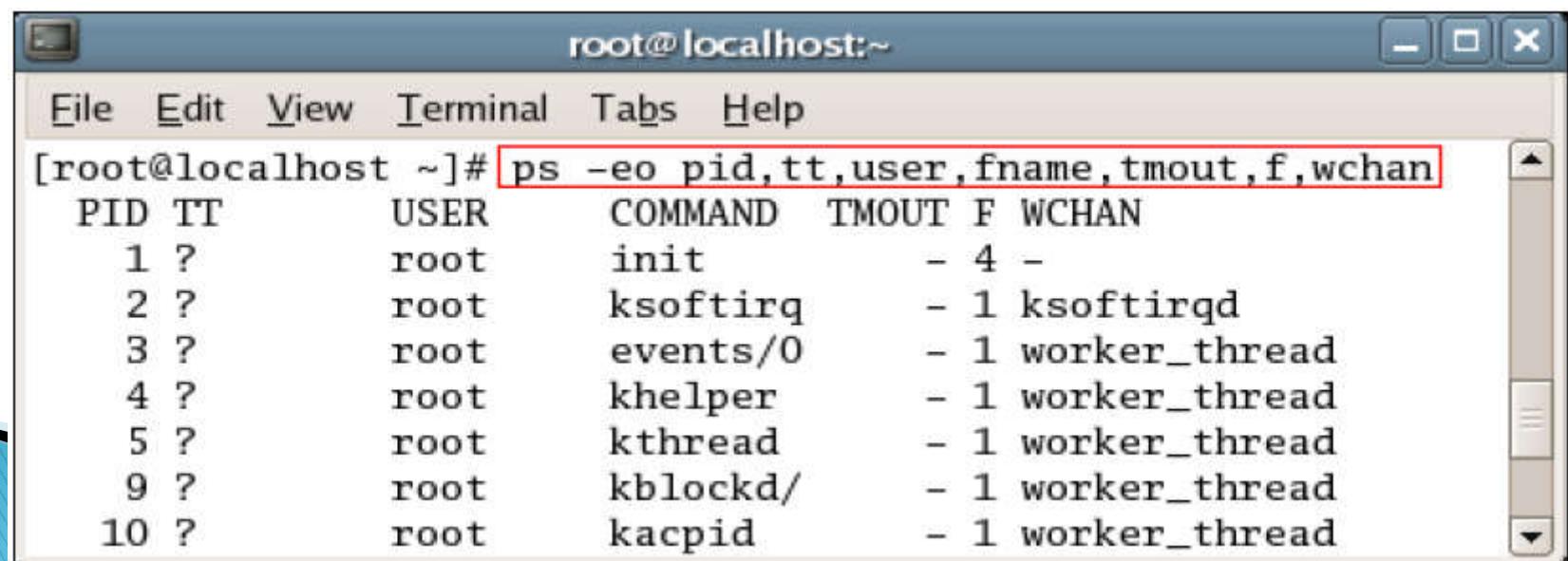
Visualiser les processus : EXEMPLES

- Format standard :



```
root@localhost:~# ps -e
  PID TTY      TIME CMD
    1 ?        00:00:00 init
    2 ?        00:00:00 ksoftirqd/0
    3 ?        00:00:00 events/0
    4 ?        00:00:00 khelper
    5 ?        00:00:00 kthread
    9 ?        00:00:00 kblockd/0
```

- On peut utiliser la syntaxe avec l'option **-o** pour ne présenter que les champs désirés :



```
root@localhost:~# ps -eo pid,tt,user,fname,tmout,f,wchan
  PID TT      USER      COMMAND   TMOUT F WCHAN
    1 ?      root      init      - 4 -
    2 ?      root      ksoftirq  - 1 ksoftirqd
    3 ?      root      events/0 - 1 worker_thread
    4 ?      root      khelper  - 1 worker_thread
    5 ?      root      kthread  - 1 worker_thread
    9 ?      root      kblockd/ - 1 worker_thread
   10 ?     root      kacpid  - 1 worker_thread
```

Visualiser les processus : EXEMPLES

- Les principaux champs disponibles sont (le libellé de la colonne est entre parenthèses) :

%cpu(%CPU)	Pourcentage de temps processeur utilisé depuis le lancement de la commande
%mem(%MEM)	Pourcentage d'utilisation de la mémoire disponible
comm(COMMAND)	Nom de la commande à l'origine du processus
command(COMMAND)	Nom de la commande avec ses arguments
time(TIME)	Temps total d'utilisation du CPU pour ce processus
etime(ELAPSED)	Temps écoulé depuis le lancement du processus
gid(GID)	GID sous lequel s'exécute le processus
group(GROUP)	Nom du groupe sous lequel s'exécute le processus
lstart(STARTED)	Date de lancement du processus
nice(NI)	Niveau de priorité du processus
pagein(PAGEIN)	Nombre de pages mémoire en SWAP devant être rapatriées en RAM pour l'exécution du processus
pid(PID)	Identifiant du processus
ppid(PPID)	Identifiant du processus père
pri(PRI)	Priorité du processus vis-à-vis de l'ordonnanceur
rss(RSS)	Quantité de mémoire vive utilisée (en kilo-octets)
session(SESS)	PID du processus maître du groupe auquel appartient le processus

Visualiser les processus : EXEMPLES

stat(STAT)	État du processus
tty(TT)	Nom du terminal auquel est rattaché le processus
uid(UID)	UID sous lequel s'exécute le processus
user(USER)	Nom d'utilisateur sous lequel s'exécute le processus
wchain(WCHAIN)	Nom de la primitive du noyau dont la fin est attendue par le processus

- Les états définis par le champ **STAT** contiennent 3 caractères.

R (Running)	Processus en cours d'exécution
T (sTopped)	Processus stoppé (avec [Ctrl, Z] par exemple)
S (Sleeping)	Processus endormi, en attente du CPU
D (device)	Processus endormi, en attente d'une ressource (I/O par exemple). Les processus dans cet état ne peuvent pas être interrompus
Z (Zombie)	Processus. Il restera dans cet état jusqu'à ce que son père récupère son code de retour.

- Des caractères additionnels peuvent figurer pour donner des informations complémentaires.
- Le second caractère du champ **STAT** est positionné à **W** si le processus est déplacé en espace de pagination.
- Enfin un **N** apparaît en troisième caractère si le processus présente une valeur **nice** positive.

Commande time

- La commande time **mesure les durées d'exécution d'une commande**, idéal pour connaître les temps de traitement, et retourne trois valeurs :



A screenshot of a terminal window titled "root@localhost:~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The command prompt is "[root@localhost ~]#". A red box highlights the command "time gzip install.log". The output shows the execution times for three categories: real, user, and sys. The output is as follows:

```
real    0m0.026s
user    0m0.000s
sys     0m0.008s
[root@localhost ~]#
```

- **real:** durée totale d'exécution de la commande ;
- **user:** durée du temps CPU nécessaire pour exécuter le programme
- **system:** durée du temps CPU nécessaire pour exécuter les commandes liées à l'OS (appels système au sein d'un programme).

Commande pstree

- ❑ **pstree** affiche les processus sous forme d'arbre généalogique

```
# pstree
init+-apmd
| -gpm
| -keventd
| -6*[mingetty]
| -syslogd
| -xfs
`-xinetd--in.telnetd--login--csh--su--bash--pstree
```

- ❑ **top** affiche en **temps réel la liste des processus actifs**, réactualisée en permanence.

Arrêter un processus

- Commande **kill** permet d'arrêter un processus. Pour pouvoir arrêter un processus, vous devez connaître son **PID** (commande **ps**), puis vous tapez :

```
kill -9 PID
```

- Un utilisateur **ne peut arrêter que ses propres processus**. Seul l'administrateur système a le droit d'arrêter un processus ne lui appartenant pas.

Arrêter un processus : Commande kill

- La commande **kill** termine un ou plusieurs processus référencés par leurs **IDs**.

kill *options* **ID**

- Options

- **-l** : Affiche la liste des signaux.
- **-signal** : Le signal (référencé par son numéro ou par son nom) est envoyé aux processus. Le signal **9 (KILL) tue systématiquement les processus**.

```
kill -9 PID
```

```
# kill -9 24721
```

Changer la priorité d'un processus

- Les processus tournent avec un certain degré de **priorité**;
- un processus plus **prioritaire** aura tendance à **monopoliser** plus souvent les **ressources** du système pour arriver le plus vite possible au terme de son exécution;
- le rôle du **système** d'exploitation de **gérer ces priorités**.



Changer la priorité d'un processus : Commande nice

- La commande **nice** permet **de lancer une commande avec une priorité plus faible**, afin de permettre éventuellement à d'autres processus de tourner plus rapidement;

```
  nice valeur commande
```

- Plus le nombre est grand, plus la priorité est faible. Par exemple **-2** est plus prioritaire que **0**;
- Une valeur **positive** causera une **baisse** de **priorité**, une valeur **négative** augmente la priorité (si autorisé).
- La valeur doit être comprise entre **-20** et **20**.

Changer la priorité d'un processus : Commande renice

- La commande **renice** fonctionne un peu comme **nice** mais elle permet de modifier la priorité en fonction d'un **utilisateur**, d'un **groupe** ou d'un **PID**. **La commande visée être en exécution .**

renice [-n prio] [-p] [-g] [-u] ID

- La priorité doit être comprise entre **-20** et **20**. **L'utilisateur standard ne peut utiliser que les valeurs entre 0 et 20** permettant de baisser la priorité. L'option **-p** précise un **PID**, **-g** un **GID** et **-u** un **UID**;

Changer la priorité d'un processus : Commande nohup

- Quand le **shell** est **quitté (exit, [Ctrl] D...)** le signal **1 SIGHUP** est envoyé aux enfants pour qu'ils se terminent aussi.
- Lorsqu'un traitement long est lancé en tâche de fond et que l'utilisateur veut quitter le **shell**, ce traitement sera alors arrêté et il faudra tout recommencer.
- Pour éviter qu'un processus ne soit arrêté à la déconnexion du terminal, il faut avoir lancé sa commande en la précédent de **nohup**.

nohup *commande*

Utilisation de la mémoire : Commande free

- La commande **free** affiche la quantité totale de mémoire physique et d'espace swap du système, de même que la quantité de mémoire utilisée, libre, partagée, tampon dans le noyau et cache.



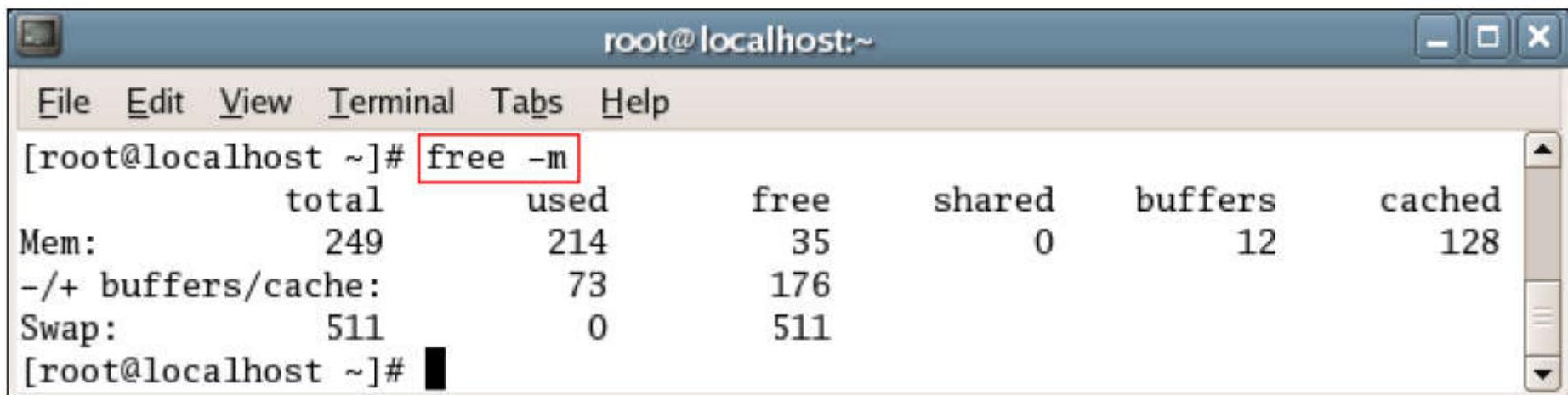
The screenshot shows a terminal window titled "root@localhost:~". The window contains a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". Below the menu is a command-line interface. A red box highlights the command "free". The output of the "free" command is displayed in a table:

	total	used	free	shared	buffers	cached
Mem:	255880	219656	36224	0	13196	131288
-/+ buffers/cache:		75172	180708			
Swap:	524280	0	524280			

[root@localhost ~]#

Utilisation de la mémoire : Commande free

- La commande **free -m** (-k, -g) affiche les **mêmes informations**, mais en **méga-octets**, ce qui les rend plus faciles à lire.



The screenshot shows a terminal window titled "root@localhost:~". The window contains the following text:

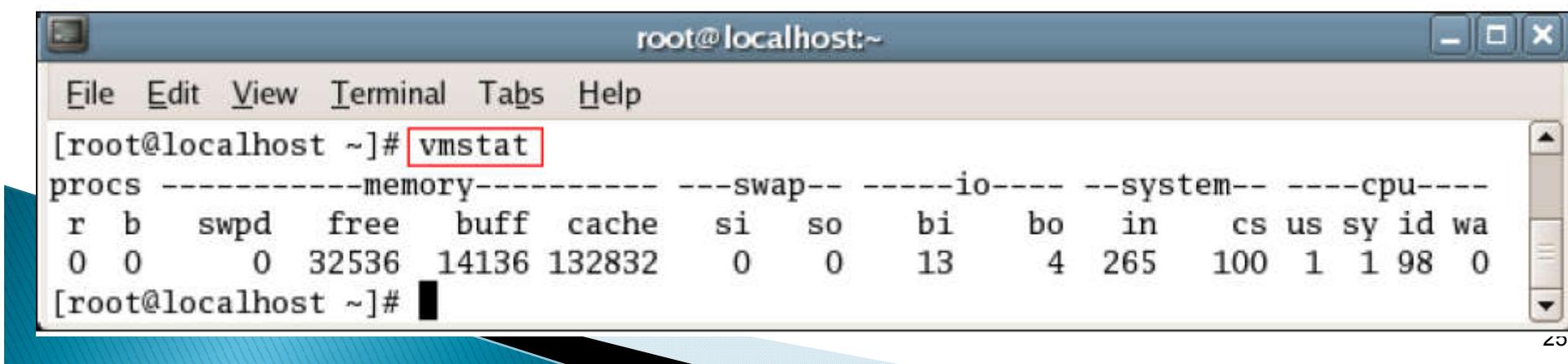
```
root@localhost:~# free -m
              total        used        free      shared    buffers     cached
Mem:       249         214         35          0         12        128
-/+ buffers/cache:    73        176
Swap:      511          0        511
root@localhost:~#
```

The command `free -m` is highlighted with a red box.

Utilisation de la mémoire : Commande vmsat

- La commande **vmstat** donne de nombreuses informations sur la mémoire virtuelle.
- **Par exemple**, pour afficher les statistiques sur la mémoire virtuelle, **sans réécrire les en-têtes de colonnes à chaque fois (-n)**, **toutes les secondes (1) et ce, 5 fois (5)**.

```
# vmstat -n 1 5
```



The screenshot shows a terminal window titled "root@localhost:~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The command [root@localhost ~]# vmstat was entered, and the output is displayed:

procs	memory				swap		io		system			cpu			
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa
0	0	0	32536	14136	132832	0	0	13	4	265	100	1	1	98	0

Utilisation de la mémoire : Commande vmsat

- La signification des champs est :

procs	Nombre de processus prêts à être exécutés (r), endormis et ne pouvant être interrompus (b)
memory	Quantité en kilo-octets de mémoire virtuelle utilisée (swpd), de mémoire physique libre (free), de mémoire utilisée comme tampons (buff) et comme cache (cache)
swap	Quantité de mémoire paginée en kilo-octets par seconde lue depuis le disque (si) et transférée sur disque (so)
io	Nombre de blocs lus (bi) et écrits (bo) par second sur des périphériques de type blocs
system	Nombre d'interruptions par seconde (in) et nombre de changement de contexte (cs)
cpu	Pourcentage d'utilisation du processeur en mode utilisateur (us) et en mode noyau (sy), ainsi que le pourcentage d'inactivité (id) et d'attente d'i/o (wa)