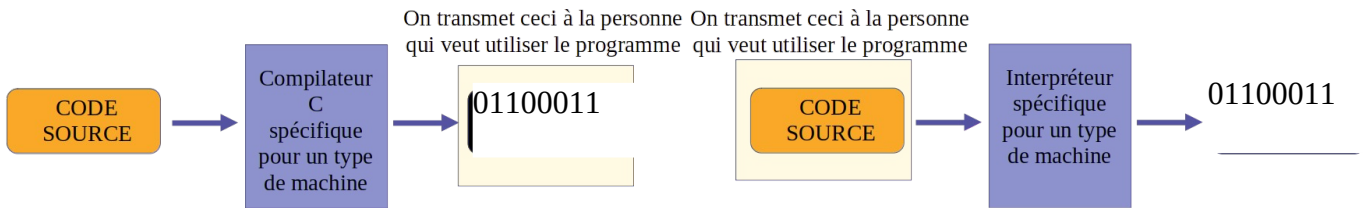


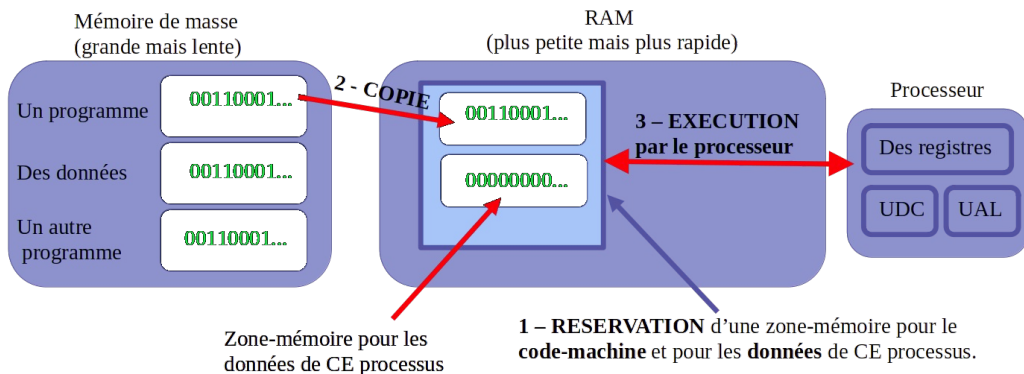
# Architecture 22 – Gestion des Processus

## I – Vocabulaire : Programme, processus et processeur



Document 1 : PRINCIPE LANGAGE COMPILÉ

Document 2 : PRINCIPLE LANGAGE INTERPRÉTÉ



DOCUMENT 3 : Création d'un processus

- Q 01° Qu'est-ce qui limite concrètement le nombre de processus pouvant être lancer en même temps ?
- Q 02° Qu'est-ce qu'un système d'exploitation multitâche ?
- Q 03° Que va devoir faire un système d'exploitation multitâche lorsque plusieurs processus fonctionnent "en même temps" sur un ordinateur ne disposant que d'un seul microprocesseur (un cœur) ?
- Q 04° Que se passe-t-il au démarrage de l'ordinateur ?

### COURS ET DEFINITIONS à connaître :

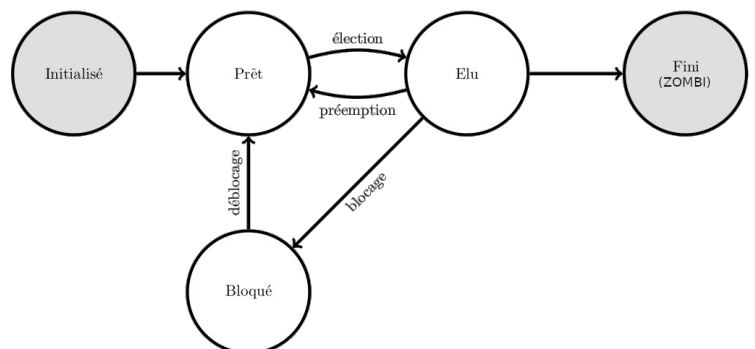
- PROGRAMME
- PROCESSEUR
- PROCESSUS
- CREATION D'UN PROCESSUS
- Chargeur d'amorçage
- DEMONS
- APPLICATION

## II – Etats d'un processus : son cycle de vie

05° Quels sont les états pendant lesquels la mémoire vive réserve de la place au processus ?

### COURS ET DEFINITIONS à connaître :

**Vous devez être capable de redessiner le graphe des états et expliquer clairement ce que signifie chacun des états.**



### III – Linux

#### DOCUMENT 4 : Exemple de commande **ps**

```
rv@rv-HP2:~$ ps
PID  TTY  TIME      CMD
17255 pts/1 00:00:00  bash
18060 pts/1 00:00:00  ps
```

**Q 06°** Lancer Blender de cette façon. Gardez-vous la main sur le bash tant que Blender reste ouvert ?

```
rv@rv-HP2:~$ blender
```

**Q 07°** Stopper Blender à l'aide de son interface graphique. Retaper ensuite un appel à Blender mais en rajoutant un &. Que constate-t-on au niveau du bash ? A quoi sert le rajout de & ?

```
Saved session recovery to '/tmp/quit.blend'
Blender quit
```

```
rv@rv-HP2:~$ blender &
```

```
[1] 19369
```

```
rv@rv-HP2:~$ ■
```

**Q 08°** A quoi correspondent les informations que nous avons sous l'appel de Blender ?

**Q 09°** Fermer le bash sans fermer Blender. Que constate-t-on ?

**Q 10° (voir Document 5)** Qui est le père (nom et ID) du processus basé sur **pstree** ? A-t-il d'autres enfants (noms et PID éventuels) ?

Qui est le processus-racine (celui qui n'a pas de parent) ? Quel est le PID de ce processus si particulier ?

**Q 11°** A votre avis, que va-t-il se passer si on tape ceci ?

```
rv@rv-HP2:~$ kill 3791
```

#### DOCUMENT 5 : Exemple de commande **pstree** (pour afficher l'arbre avec les PID : **pstree -p**)

```
rv@rv-HP2:~$ pstree -p
systemd(1)─ModemManager(1064)─┬─{ModemManager}(1135)
│                               └─{ModemManager}(1139)
├─NetworkManager(922)─┬─{NetworkManager}(1038)
│                       └─{NetworkManager}(1063)
├─accounts-daemon(975)─┬─{accounts-daemon}(1009)
│                       └─{accounts-daemon}(1034)
├─systemd(1335)─(sd-pam)(1336)
│
└─gnome-terminal-(3791)─┬─bash(3799)─python(3808)─+++
│                       └─bash(20552)─blender(2056+
│                                       └─pstree(20563)
```

```
rv@rv-HP2:~$ pstree -s 20563          (permet d'afficher les parents du PID 20563)
systemd─systemd─gnome-terminal─bash─pstree
```

#### DOCUMENT 6 : Exemple d'autres commandes **ps**

```
rv@rv-HP2:~$ man ps
– documentation de la commande ps –
```

##### Option BSD

**a** : voir tous les processus      **u** : voir l'utilisateur      **x** : même ceux sans terminal

##### Option Linux

**-e** : voir tous les processus    **-f** : affichage complet

```
rv@rv-HP2:~$ ps aux
USER      PID  %CPU  %MEM  VSZ   RSS  TTY   STAT  START  TIME  COMMAND
root         1    0.0   0.1 169288 12032 ?     Ss    08:16  0:03 /sbin/init sp
root         2    0.0   0.0    0     0  ?     S     08:16  0:00 [kthreadd]
systemd+   809    0.0   0.1  25308 13280 ?     Ss    08:16  0:05 /lib/systemd/
rv        1335    0.0   0.1  20448 10808 ?     Ss    08:17  0:03 /lib/systemd/
rv       22334    0.7   0.0  15344  5956 pts/1  Ss    18:11  0:00 bash
rv       22342    0.0   0.0   16928  4336 pts/1  R+    18:11  0:00 ps aux
```

```
rv@rv-HP2:~$ ps -ef
UID      PID  PPID  C    STIME TTY      TIME      CMD
rv     12008  3122  0    20:37 ?        00:00:00  /snap/firefox/1025/usr/lib/f
rv     12542 11946  0    21:06 pts/2    00:00:00  ps -ef
```

```
rv@rv-HP2:~$ ps ax -ef
UID      PID  PPID  C  STIME TTY      STAT  TIME  CMD
rv     12008  3122  0  20:37 ?        Sl    0:00  /snap/firefox/1025/usr/li
root    12219  2     0  20:44 ?        I     0:00  [kworker/u8:0-flush-8:0]
rv     12660 11946  0  21:10 pts/2    R+    0:00  ps ax -ef
```

12° Un processus doit disparaître. Il passe en STAT Z. Qui peut le faire passer à l'état X ?

13° Les deux instances de Blender ont-elles été créées à partir du même endroit ?

```
rv@rv-HP2:~$ ps -fc blender
```

```
UID    PID    PPID    C      STIME  TTY    TIME      CMD
rv     23435  22334   6      18:52 pts/1   00:00:08  blender
rv     23498  22334  45      18:54 pts/1   00:00:02  blender
```

## IV – Interblocage (deadlock)

**DOCUMENT 7** : Illustration de l'interblocage



**Q14°** Imaginons deux processus 1 et 2 qui pourront exécuter leurs instructions à tour de rôle, une fois le processus 1, puis le 2....

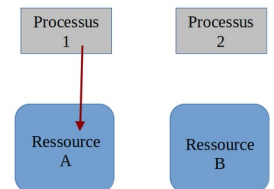
La ressource A contient "4" initialement et la ressource B contient "2".

Actions du processus 1

1. Prendre le contrôle exclusif de la ressource A ( B pour la question 15)
2. Prendre le contrôle exclusif de la ressource B ( A pour la question 15)
3. Ecrire "2" dans la ressource A
4. Copier la ressource A dans la ressource B

Actions du processus 2

1. Prendre le contrôle exclusif de la ressource A
2. Prendre le contrôle exclusif de la ressource B
3. Ecrire "8" dans la ressource B
4. Copier la ressource B dans la ressource A



On commence par exemple avec la première instruction du processus 1 : il prend le contrôle de la ressource A.

**Q15°** Même question mais en inversant les deux demandes dans le premier processus. Réaliser le schéma.

### MEMO : autres commandes usuelles sur Linux vues en 1<sup>er</sup>

Lister le contenu d'un répertoire : **ls** ou **ls -a** pour lister même les cachés.  
Changer de répertoire (directory) : **cd abc** pour aller dans le répertoire nommé abc  
Changer de répertoire (directory) : **cd ..** pour remonter dans le répertoire parent  
Créer un nouveau répertoire : **mkdir abc** pour créer un nouveau répertoire mkdir

**abc** peut être une adresse relative ou une adresse absolue.

Adresse relative : **images/nsi/logo.png** A partir de la localisation actuelle, aller dans...

Adresse absolue : **/home/eleve/images/nsi/logo.png** Idem mais à partir de la racine /