

Processus & interblocage

Thème 3 > Partie A

✔ Objectifs et savoir-faire

- ✔ Comprendre le traitement des processus par un système d'exploitation (ordonnancement)
- ✔ Connaître le nom et le sens des différents états d'un processus
- ✔ Savoir interpréter le résultat d'une commande `ps`
- ✔ Déterminer si une situation relève d'un interblocage

Processus et cycle de vie

Définition

Processus

Un processus est une instance d'une application. Il est défini par :

- un ensemble d'instructions à exécuter (un programme);
- un espace mémoire pour les données de travail;
- éventuellement, d'autres ressources, comme des descripteurs de fichiers, des ports réseau, etc.

Définition

ordonnanceur

Dans la plupart des ordinateurs, les processeurs sont capables d'exécuter qu'un ou qu'un nombre très limité de processus en parallèle.

Les systèmes d'exploitation comportent donc un **ordonnanceur**. C'est le composant du noyau du système d'exploitation qui choisit l'ordre d'exécution des processus sur les processeurs d'un ordinateur. En anglais, l'ordonnanceur est appelé scheduler.

Le rôle de l' **ordonnanceur** du noyau, est de permettre à tous les processus de s'exécuter à un moment ou un autre et d'utiliser au mieux le processeur pour l'utilisateur. Il va donc choisir quand et quel processus s'exécute et lesquels seront en pause.

Exercice 1 ★

On considère P1, P2 et P3, tous soumis à l'instant 0 dans l'ordre 1,2,3 :

Nom du processus	Durée d'exécution en unité de temps	Ordre de soumission
P1	3	3
P2	2	1
P3	2	2

1. Dans cette question, on considère que les processus sont exécutés de manière concurrente selon la politique du tourniquet : le temps est découpés en tranches, nommé *quantums de temps*.

Les processus prêts à être exécutés sont placés dans une file d'attente selon leur ordre de soumission.

Lorsqu'un processus est élu, il s'exécute au plus durant un quantum de temps. Si le processus n'a pas terminé son exécution à l'issue du quantum de temps, il réintègre la file des processus prêts (côté entrée). Un autre processus, désormais en tête de file (côté sortie) est alors élu pour une durée égale à un quantum.

P2							
----	--	--	--	--	--	--	--

Exercice 2 ★

On considère P1, P2, P3 et P4.

Nom du processus	Durée d'exécution en unité de temps
P1	3
P2	1
P3	2
P4	1

1. Dans cette question, on considère que les processus sont exécutés de manière concurrente selon la politique du "plus court d'abord".

Les processus sont exécutés entièrement dans l'ordre croissant de leurs temps d'exécution, le plus court étant exécuter en premier.

--	--	--	--	--	--	--	--

Définition

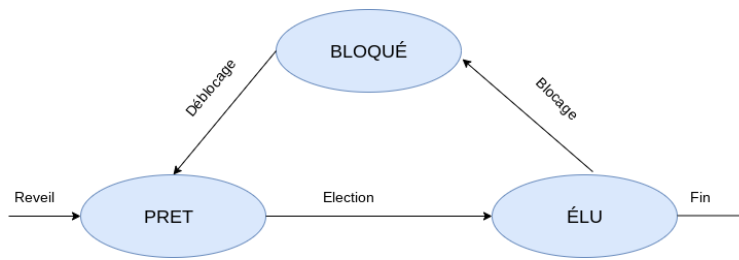
Etats des processus

Pour pouvoir gérer les différents processus, les systèmes d'exploitation vont gérer les demandes à tour de rôle. Ainsi les processus peuvent avoir différents états :

- **prêt** : Le processus a les ressources nécessaires et attend d'être traité
- **élu** : Le processus est cours d'exécution
- **bloqué** : Le processus ne peut plus poursuivre son exécution car il a besoin d'une ressource qui est indisponible.
- **terminé**

Définition

Cycle de vie



Analyse des processus en cours sous linux

i

Pour connaître l'ensemble des processus s'exécutant à un instant précis sur une machine, sur un système UNIX, on peut utiliser la commande `ps` avec certaines options.

Exemple

Exemple de résultat de la commande `ps -eaf`

```

1  UID      PID     PPID  C STIME TTY          TIME CMD
2  root         1         0  0 déc.21 ?        00:00:00 /sbin/init splash
3  root         2         0  0 déc.21 ?        00:00:00 [kthreadd]
4  root         3         2  0 déc.21 ?        00:00:00 [rcu_gp]
5  root         4         2  0 déc.21 ?        00:00:00 [rcu_par_gp]
6  root         5         2  0 déc.21 ?        00:00:00 [netns]
7  root         7         2  0 déc.21 ?        00:00:00 [kworker/0:0H-events_highpri]
8  ...         ...         ...    ...    ...
9  etienne    1637         1  0 déc.21 ?        00:00:04 /usr/bin/ibus-daemon
   ↳ --daemonize --xim
10 etienne    1640    1637  0 déc.21 ?        00:00:00 /usr/libexec/ibus-dconf
11 etienne    1642    1637  0 déc.21 ?        00:00:00 /usr/libexec/ibus-ui-gtk3
12 etienne    1643    1637  0 déc.21 ?        00:00:01
   ↳ /usr/libexec/ibus-extension-gtk3
13 etienne    1645         1  0 déc.21 ?        00:00:00 /usr/libexec/ibus-x11
   ↳ --kill-daemon
14 root      8233         2  0 07:57 ?        00:00:00
   ↳ [kworker/6:1-events_freezable]
15 root      8251         2  0 07:58 ?        00:00:00 [kworker/2:0-events]
16 root      8253         2  0 07:58 ?        00:00:00 [kworker/1:1]
17 etienne    8447    2193  0 08:06 pts/2    00:00:00 /bin/bash
18 etienne    8454    8447  0 08:06 pts/2    00:00:00 ps -eaf
  
```

- **pid, ppid**

Définition

PID

Chaque processus est identifié par un nombre. Ce nombre est appelé **PID** (Process Identification).

Pour chaque processus créé, le système d'exploitation incrémente un compteur de 1 dont il se sert pour attribuer les PID aux processus. Ainsi le premier processus créé au démarrage du système a pour PID 0, le second 1 et ainsi de suite ...

Définition

PPID

Le système d'exploitation attribue aussi à chaque processus un **PPID** (Parent Process Identification) qui représente le processus parent d'un processus.

⚠ Seul le processus 0 ne possède pas de PPID.

- **Les différents états**



La commande **ps** permet aussi de connaître l'état des différents processus. Par exemple la commande ci-contre, **ps -eo pid, ppid, stat, command**, permet d'avoir une colonne STAT dans l'affichage de sortie.

Définition

Les différents états

- **D** : en sommeil non interruptible (normalement entrées et sorties);
- **R** : s'exécutant ou pouvant s'exécuter (dans la file d'exécution);
- **S** : en sommeil interruptible (en attente d'un événement pour finir);
- **T** : arrêté, par un signal de contrôle des tâches ou parce qu'il a été tracé;
- **W** : pagination (non valable depuis le noyau 2.6.xx);
- **X** : tué (ne devrait jamais être vu);
- **Z** : processus zombie (), terminé mais pas détruit par son parent;
- **<** : haute priorité (non poli pour les autres utilisateurs);
- **N** : basse priorité (poli pour les autres utilisateurs);
- **L** : avec ses pages verrouillées en mémoire (pour temps réel et entrées et sorties personnalisées);
- **s** : meneur de session;
- **l** : possède plusieurs processus légers;
- **+** : dans le groupe de processus au premier plan.

Exercice 3

La commande `ps -eo pid,ppid,stat,command`, nous l'affichage :

```

1  PID    PPID  STAT  COMMAND
2  1      0     Ss    /sbin/init splash
3  2      0     S     [kthreadd]
4  ...    ...   ...   ...
5  397958  2     I<    [kworker/0:0H-events_highpri]
6  459678  1     S     dbus-launch --autolaunch
7  459679  1     Ss    /usr/bin/dbus-daemon
8  459681  1     Sl    kwalletd5
9  460366  136   Sl    /usr/bin/codium -b /home/etienne/Téléchargements/Edt_CLAVE.ics
10 460369  1587  Sl    /usr/bin/codium -b /home/etienne/Documents/cours_processus.tex
11 461367  136   Sl    /usr/bin/codium -b /home/etienne/Téléchargements/test.txt
12 554030  7877  Sl    /opt/google/chrome/chrome
13 554031  7877  Sl    /opt/google/chrome/chrome
14 554052  1511  R+    ps -eo pid,ppid,stat,command

```

1. A quel commande correspond le processus qui à pour PID 459681 ?

.....

2. Donner le PID du processus qui a démarré le processus qui a pour PID 459681 ?

.....

3. Donner le PID d'un processus prioritaire ?

.....

4. Quelle commande est en cours d'exécution ?

.....

Exercice 4

La commande UNIX `ps` présente un cliché instantané des processus en cours d'exécution. Avec l'option `-eo pid,ppid,stat,command`, cette commande affiche dans l'ordre l'identifiant du processus PID (process identifier), le PPID (parent process identifier), l'état STAT et le nom de la commande à l'origine du processus.

Les valeurs du champ STAT indique l'état des processus :

- R : processus en cours d'exécution
- S : processus endormi

Sur un ordinateur, on exécute la commande `ps -eo pid,ppid,stat,command` et on obtient un affichage dont on donne ci-dessous un extrait.

```

1 $ ps -eo pid,ppid,stat,command
2 PID      PPID      STAT      COMMAND
3 1         0         Ss        /sbin/init
4 ...      ...      ...      ...
5 1912     1908     Ss        bash
6 2014     1912     Ss        bash
7 1920     1747     Sl        gedit
8 2013     1912     Ss        bash
9 2091     1593     Sl        /usr/lib/firefox/firefox
10 5437     1912     Sl        python programme1.py
11 5440     2013     R         python programme2.py
12 5450     1912     R+       ps -eo pid,ppid,stat,command

```

À l'aide de cet affichage, répondre aux questions ci-dessous.

1. Quel est le nom de la première commande exécutée par le système d'exploitation lors du démarrage?

.....

2. Quels sont les identifiants des processus actifs sur cet ordinateur au moment de l'appel de la commande `ps`? Justifier la réponse.

.....

3. Depuis quelle application a-t-on exécuté la commande `ps`? Donner les autres commandes qui ont été exécutées à partir de cette application.

.....

4. Expliquer l'ordre dans lequel les deux commandes `python programme1.py` et

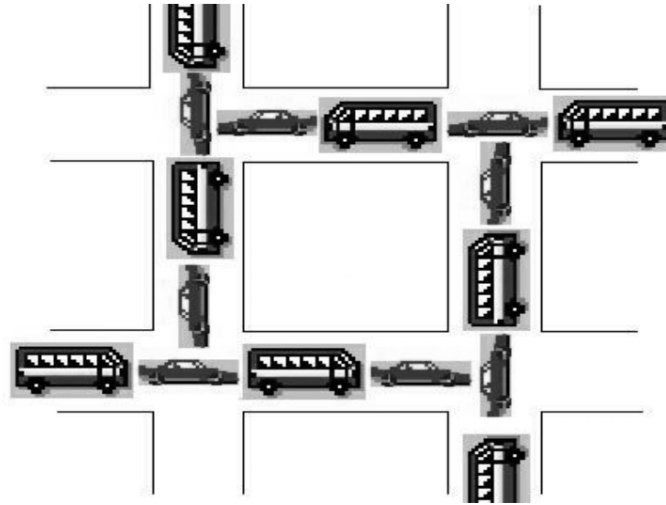
.....
 `python programme2.py` ont été exécutées.

.....

5. Peut-on prédire que l'une des deux commandes `python programme1.py` et `python programme2.py` finira avant l'autre?

.....

Interblocage



Définition

Etapes de l'utilisation d'une ressource

L'utilisation d'une ressource passe par les étapes suivantes :

- Demande de la ressource : Si l'on ne peut pas satisfaire la demande il faut attendre. La demande sera mise dans une table d'attente des ressources.
- Utilisation de la ressource : Le processus peut utiliser la ressource.
- Libération de la ressource : Le processus libère la ressource demandée et allouée

Définition

Interblocage

Un interblocage (deadlock en anglais) est un phénomène qui peut survenir en programmation concurrente. L'interblocage se produit lorsque des processus concurrents s'attendent mutuellement. Les processus bloqués dans cet état le sont définitivement.

Exemple d'interblocage : le processus P1 utilise la ressource R2 qui est attendue par le processus P2 qui utilise la ressource R1, attendue par P1. P1 ne peut pas libérer la ressource R2 car il attend la ressource R1 et P2 ne peut pas libérer la ressource R2 car il attend la ressource R2.

Exercice 5

On trouvera ci-dessous deux programmes rédigés en pseudo-code **Verrouiller** un fichier signifie que le programme demande un accès exclusif au fichier et l'obtient si le fichier est disponible.

Programme 1

```
Verrouiller fichier_1
Calculs sur fichier_1
Verrouiller fichier_2
Calculs sur fichier_1
Calculs sur fichier_2
Calculs sur fichier_1
Déverrouiller fichier_2
Déverrouiller fichier_1
```

Programme 2

```
Verrouiller fichier_2
Verrouiller fichier_1
Calculs sur fichier_1
Calculs sur fichier_2
Déverrouiller fichier_1
Déverrouiller fichier_2
```

1. En supposant que les processus correspondant à ces programmes s'exécutent simultanément (exécution concurrente), expliquer le problème qui peut être rencontré.

.....

.....

.....

.....

2. Proposer une modification du programme 2 permettant d'éviter ce problème.

.....

.....

.....

.....

.....

Exercice 6

Un constructeur automobile utilise des ordinateurs pour la conception de ses véhicules. eux-ci sont munis d'un système d'exploitation ainsi que de nombreuses applications parmi lesquelles on peut citer :

- un logiciel de traitement de texte ;
- un tableur ;
- un logiciel de Conception Assistée par Ordinateur (CAO) ;
- un système de gestion de base de donnée (SGBD).

Chaque ordinateur est équipé des périphériques classiques : clavier, souris, écran et est relié à une imprimante réseau. Un ingénieur travaille sur son ordinateur et utilise les quatre applications citées au début de l'énoncé.

Pendant l'exécution de ces applications, des processus mobilisent des données et sont en attente d'autres données mobilisées par d'autres processus.

On donne ci-dessous un tableau indiquant à un instant précis l'état des processus en cours d'exécution et dans lequel D1, D2, D3, D4 et D5 sont des données. La lettre **M** signifie que la donnée est mobilisée par l'application ; la lettre **A** signifie que l'application est en attente de cette donnée.

Lecture du tableau : le logiciel de traitement de texte mobilise (M) la donnée D1 et est en attente (A) de la donnée D2.

	D1	D2	D3	D4	D5
Traitement de texte	M	A			
Tableur	A				M
SGBD		M	A	A	
CAO		M	A	M	A

Montrer que les applications s'attendent mutuellement. Comment s'appelle cette situation ?