

200 Processus

INF3173

Principes des systèmes d'exploitation

Jean Privat

Université du Québec à Montréal

Hiver 2021

La dernière fois

Qu'est-ce qu'un SE ?

- Couche logicielle située entre le matériel et les applications d'un ordinateur
- A l'exclusivité des mécanismes matériels (mode noyau)

Objectifs

- À abstraire la couche matérielle (appels système)
- À répartir équitablement les ressources entre les différents processus et utilisateurs
- À protéger le matériel, les processus et les utilisateurs les uns des autres

Analogie facile

- Le SE est le « gouvernement » de l'ordinateur

Processus

Processus : Définition

Définition : Un processus est

- Un programme en cours d'exécution
- Par un processeur

3 concepts liés

- Processus : l'exécution en cours d'un programme
- Processeur : celui qui fait l'exécution
- Programme : la suite d'instructions (prédéterminée)

Fondamental

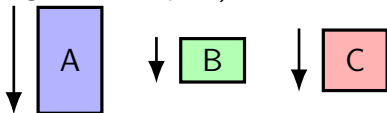
- Plusieurs instances d'un même programme
- Chacune de ces instances est appelée un processus
- Chaque processus est **autonome** et **isolé**

Exécution d'un processus

Chaque processus a l'impression d'être seul

Un processus

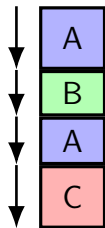
- Progrèsse de manière séquentielle dans son programme
- Progrèsse en même temps que les autres
- Ne prend pas en compte l'existence des autres processus (sauf si c'est programmé exprès)



Problème

- Un seul processeur (ou un nombre fixe)
- Plusieurs processus à exécuter en même temps

Illusion de parallélisme : multitâche



- Avancer chaque processus un petit peu à la fois
 - Changer le processus actif régulièrement (10ms–100ms)
 - On parle de **changement de contexte**
 - Le changement est couteux (à ne pas abuser)
 - Qui décide quel processus devient actif?
 - Le système d'exploitation
 - En fonction de politiques et de priorités
 - On parle d'**ordonnancement**
- On y reviendra...

Multiprocesseurs et multicœurs

La même idée de base

- Chaque cœur traite un processus actif
- On change les processus actifs régulièrement

C'est juste plus compliqué...

Analogie : la cuisine

- Des cuisiniers dans une cuisine
 - Des recettes rédigées sur papier
 - Des plats en train d'être préparées
 - Des entrées-sorties dans le four
 - Des clients qui veulent un service rapide
-
- Des processus dans un ordinateur
 - Des programmes enregistrés sur le disque
 - Des processus en train d'être exécutés
 - Des entrées-sorties sur les périphériques
 - Des utilisateurs qui veulent un service rapide

Analogie : la suite

- Un cuisinier peut préparer plusieurs plats en même temps
Voire plusieurs plats d'une même recette
- Un cuisinier optimise son temps
Il fait autre chose plutôt que d'attendre devant le four
- Un cuisinier répartit son temps entre les plats
Tous les plats avancent dans leur préparation
- Certaines tâches peuvent être plus prioritaires
En particulier, s'il y a une odeur de brûlé

Information des processus

Table des processus (ou *Process Control Block*)

Structure conceptuelle qui regroupe les informations d'un processus

- État du processus
- Registres du CPU : dont CO (*PC*), PP (*SP*), et mot d'état
- Info sur le processus : pid, priorité, utilisateurs, statistiques...
- Info sur la gestion mémoire
- Info sur les E/S: répertoire de travail, fichiers ouverts, blocs verrouillés

La nature et le détail varient d'un système à l'autre

Question

- Où est stockée la table des processus ?

PID

Chaque processus est identifié par un numéro : le pid

- `pid_t getpid(void)`

Chaque processus a un parent

- Hiérarchie de processus
- `init` est la racine de la hiérarchie (`pid=1`)
- `pid_t getppid(void)`



`ps(1)` et `top(1)` donnent de l'information sur les processus

- Nombreuses options et variations plus ou moins compatibles entre systèmes Unix

```
$ ps aux
```

Colonnes intéressantes

- PID: identifiant du processus
- PPID: identifiant du processus parent
- START: date de création du processus
- TIME: temps passé sur le processeur
- COMMAND: ligne de commande originale



- Typiquement nommé /proc
- Un sous-répertoire par processus, utilisant le PID
- Beaucoup d'information bas niveau et fluctuante
- Voir le man [proc\(5\)](#)

Expose de l'information

- Sous forme plus ou moins humaine
- Plus ou moins portable entre Unix

Pourquoi /proc ?

C'est plus simple ainsi

- Pour le noyau d'exposer de l'information
 - Pour les programmes d'aller chercher l'information
 - Qu'un ensemble dédié (et fluctuant) d'appels système
- ps, top, etc. utilisent directement /proc à l'interne

Question

- Est-ce que /proc contourne les appels système ?

Entrées intéressantes de /proc

- /proc/PID/exe un lien symbolique vers l'exécutable
- /proc/PID/cwd un lien symbolique vers le répertoire de travail
- /proc/PID/cmdline la ligne de commande utilisée
(\0 sépare les arguments)
- /proc/PID/environ les variables d'environnement
- /proc/PID/stat et /proc/PID/status de l'information brute
(utilisateurs, priorités, statistiques)
- /proc/PID/maps l'organisation de la mémoire (on y reviendra)
- /proc/PID/fd/ les descripteurs de fichiers utilisés (on y reviendra)
- /proc/PID/task/ les threads du processus (on y reviendra)
- /proc/self lien symbolique vers le processus courant

```
$ ls -l /proc/self/exe
$ readlink /proc/self/exe
$ cat --show-nonprinting --number /proc/self/cmdline
$ lolcat /proc/self/cmdline
```