

Episode 2 - Les signaux

Programmation système unix

BOOTSTRAP

Cet épisode est une introduction au fonctionnement des signaux dans un système Unix.

Le noyau Unix “informe” les processus au travers de signaux pour leur transmettre de potentiels problèmes (SIGSEGV par exemple pour une erreur de segmentation cf. épisode sur la mémoire en C)

Chaque signal est associé à une valeur par défaut.

Lisez le man concerné pour comprendre le comportement par défaut de traitement de chaque signaux par les processus.

Plusieurs appels systèmes existent pour émettre et attraper les signaux.

Nous allons en voir un partie, mais vous pouvez découvrir d’autres méthodes : have fun.

PREAMBULE

Vilebrequin est une chaîne YouTube francophone consacrée à l'automobile sur un ton humoristique et décalé, créée en janvier 2017 par deux vidéastes web français, Pierre Chabrier et Sylvain Levy. Les deux vidéastes mettent fin à la chaîne en décembre 2023.

Sylvain Levy, alors employé au sein de l'entreprise Valeo, rencontre Pierre Chabrier, directeur de la photographie, à l'occasion d'un covoiturage en 2016. Les deux hommes se lient d'amitié et se découvrent une passion commune pour le monde de l'automobile. Ils décident de créer la chaîne YouTube Vilebrequin le 5 janvier 2017. Ce nom est adopté en référence au dispositif mécanique et à la marque de vêtements du même nom, ainsi que pour son aspect jugé décalé par le duo.

Les deux vidéastes publient pour la première fois du contenu sur leur chaîne en juillet 2017. Leur notoriété croît lentement au cours des mois suivants, tout en restant limitée ; ils connaissent leur premier grand succès en avril 2018, avec une vidéo où ils tentent, sur route fermée, de franchir un dos d'âne à 130 km/h.

En septembre 2018, Sylvain Levy est sommé par son employeur Bosch de quitter l'entreprise, jugeant cette autre activité de vidéaste web, parfois filmée dans les locaux de la marque, incompatible avec son travail. Avec ses 89 000 abonnés, il ne gagne qu'environ 300 € par mois grâce à ses vidéos. En 2018 et 2019, la notoriété de la chaîne grandit pour atteindre 356 000 abonnés en janvier 2020. Le duo est invité à plusieurs reprises, début 2019, dans l'émission Sans se braquer sur Automoto la chaîne.

Les deux créateurs gagnent en popularité, notamment grâce à une approche humoristique et décalée, jusque dans leurs collaborations publicitaires.

Début 2022, la chaîne atteint le nombre de 1,72 million d'abonnés.

Voilà.

EXERCICES

EXERCICE 1

Ecrire un programme nommé **process_info** qui affiche les informations suivantes :

- Son process ID
- Son parent process ID
- Son group process ID

Exemple :

```
~/> ./process_info
PID: 18975
PPID: 18954
PGID: 18973
~/>
```

Tips :

man getpid

EXERCICE 2

Ecrire un programme nommé **kill_that** qui envoie le signal SIGQUIT au processus passé en 1er argument.

Exemple :

```
~/> ./kill_that 4284
~/>
```

Tips :

man 1 kill

man 2 kill

man 7 signal, chapitre **Signal numbering for standard signals** (pour architecture x86)

Pour tester, utiliser la commande ps pour obtenir le PID du programme à fermer (firefox, chrome, etc.)

EXERCICE 3

Ecrire un programme nommé **last_stand** qui patiente en ne faisant rien. Lorsqu'il reçoit un signal **SIGINT**, il affiche **BYE BYE POOR WORLD**\n et se termine.

Terminal 1

```
~/> ./last_stand &
[1] 4242

...

BYE BYE POOR WORLD
~/>
```

Terminal 2

```
~/> kill -INT 4242
```

Tips :

man 2 signal, man 2 pause

EXERCICE 4

Ecrire un programme nommé **who_push_me** qui patiente en ne faisant rien. Lorsqu'il reçoit un signal parmi :

- **SIGUSR1**
- **SIGUSR2**
- **SIGINT**
- **SIGQUIT**

il affiche une description du signal reçu et le processus de l'émetteur.

Terminal 1

```
~/> ./who_push_me &
[1] 4242
~/>
[FROM 4245] received SIGUSR1 (Signal User defined signal 1)
```

```
[FROM 4246] received SIGUSR2 (Signal User defined signal 2)
[FROM 4247] received SIGUSR1 (Signal User defined signal 1)
[FROM 4248] received SIGQUIT (Signal Quit)
[FROM 4249] received SIGINT (Signal Interrupt)
```

Terminal 2

```
~/> kill -USR1 4242
~/> kill -USR2 4242
~/> kill -USR1 4242
~/> kill -QUIT 4242
~/> kill -INT 4242
```

Tips :

man strsignal

man sigaction

EXERCICE 5

Ecrire un programme nommé **signal_me** qui patiente en ne faisant rien. Lorsqu'il reçoit un signal **SIGUSR1** ou **SIGUSR2**, il affiche le décompte du nombre de signaux de ce type reçu depuis son démarrage.

Terminal 1

```
~/> ./signal_me &
[1] 4242
~/>
SIGUSR1: 0 | SIGUSR2: 0
SIGUSR1: 0 | SIGUSR2: 1
SIGUSR1: 0 | SIGUSR2: 2
SIGUSR1: 0 | SIGUSR2: 3
SIGUSR1: 1 | SIGUSR2: 3
SIGUSR1: 1 | SIGUSR2: 4
```

Terminal 2

```
~/> kill -USR2 4242
~/> kill -USR2 4242
```

```
~/> kill -USR2 4242  
~/> kill -USR1 4242  
~/> kill -USR2 4242
```

Tips :

Vous pouvez utiliser une variable globale

<https://www.google.com/search?q=global+variable+c>

Cela va servir pour le projet **miniteams**