

Pratique des Machines

TP 7

Halim Djerroud (hdd@ai.univ-paris8.fr)
Mehdy Tounsi (mehdy.tounsi02@univ-paris8.fr)
Stephane Malolepsza (stephane.malolepsza@univ-paris8.fr)

Exercice 1 : Les commandes utiles

1. Dans un terminal lancez la commande `ps`, puis lancez la commande `cat`. Utilisez le raccourci clavier `CTRL+z`. Relancez la commande `ps`
2. Ouvrez un nouvel onglet du terminal et lancez à nouveau la commande `ps`.
3. Dans le même onglet lancez la commande `ps -x`. Que constatez vous ?
4. Arrêtez le processus `cat` à l'aide de la commande `kill` et du signal `SIGKILL`. Vérifier à l'aide de la commande `ps`.
5. Retournez dans le premier onglet et relancez la commande `cat`. Puis dans le deuxième onglet, mettez le processus `cat` en pause à l'aide de la commande `pkill` et du signal `SIGSTOP`. Retournez dans le premier onglet. Que constatez vous ?
6. Arrêtez le processus `cat` à l'aide de la commande `pkill` et du signal par défaut de cette commande. `pkill [nom processus]`. Vérifiez . Que constatez vous .
7. Retournez dans le deuxième onglet et arrêtez le processus père de `cat` aidez vous de `ps -xf` et de `kill -9 [PID]` . Que constatez vous ?
8. Utilisez la commande `pstree` et identifiez le processus qui possède le PID 1.
9. Utilisez les commandes `top` et `htop` et identifiez les processus les plus gourmands en temps processeur.
10. Utilisez la commande `ps` avec l'option `-x` et l'option `-o` qui affichera le PID , le USER , le Terminal et le NOM de la commande et l'utilisation du processeur.

Exercice 2 : Utilisons les signaux en BASH

1. Créez un répertoire `tp7` , dans votre Home Directory
2. Copiez le script `/opt/tp7/trap.sh` (source :www.shellscript.sh) dans votre Home Directory/`tp7`. Analysez son fonctionnement et lancez le. Arrêtez le à l'aide du raccourci clavier adéquat. Quel signal doit on lui envoyer et auquel il répond ?
3. Relancez le et tentez de l'arrêter autrement depuis une autre console.

4. Modifiez le script afin qu'il affiche toujours le message lors de l'appui du raccourci clavier mais qu'il continue son exécution.
5. Modifiez le afin qu'il réagisse à la réception du signal `SIGQUIT` en affichant le message " *Signal SIGQUIT received... OKAY, OKAY i'll quit immediately*" et en s'arrêtant.
6. Modifiez le pour qu'il réagisse au signal `SIGKILL` en affichant le message " *no one can kill me*". Est-ce que cela fonctionne. Quelle(s) conclusion(s) en tirez-vous ?

Exercice 3 : Utilisons les signaux en C

1. Copiez le fichier `/opt/tp7/signal.c` (source : www.yolinux.com) dans `tp7`.
2. Allez dans le répertoire `tp7`, lisez bien et tentez de bien comprendre le code source. Compilez le fichier `signal.c` : `gcc signal.c`
3. Lancez le et arrêtez le à l'aide du signal programmé dans le code source.
4. Modifiez le code source afin que le programme réponde au signal `SIGQUIT` en affichant un message : " *bad signal ..continue*" et continue son exécution. Il devra continuer à répondre au signal `SIGINT` avec le même comportement. *Utilisez les conditions.*