

TP Linux - système: process, jobs et devices

Consignes :

- Ouvrez ce document dans le lecteur pdf installé sur la machine (PAS dans le navigateur).
- Répondez aux questions en tapant votre réponse dans les champs du document.
- Sauvegarder le fichier en TP-process.pdf.
- Déposer le fichier sur l'espace de cours dédié, sur le cours "R504-R509".

1 Gestion des jobs & process

1.1 PID d'un process Shell

Q1.1 - Dans un dossier TP_process dans votre "home", créer un script start.sh qui doit afficher le PID du programme puis se mettre en sommeil (commande sleep) pendant 10mn.

Q1.2 - Dans une console, afficher le PID courant. Puis lancer le programme précédent. Est-ce le même ?

Q1.3 - Avec la commande ps aux filtrée sur le PID initial, dans quel état se trouve le process initial ?

Q1.4 - Arreter le programme (CTRL-C) puis le relancer en ajoutant le "&" à la fin pour garder la main.
Est-ce le même PID ?

Q1.5 - De façon similaire, dans quel état se trouve le process initial ?

Q1.6 - Rechercher dans /proc le dossier correspondant au process que vous avez lancé. Proposer une commande qui donne le nombre de dossier qui s'y trouvent¹ :

Q1.7 - Combien en trouvez-vous ?

Q1.8 - Proposer une commande qui donne le nombre de fichiers qui s'y trouvent :

Q1.9 - Combien en trouvez-vous ?

Q1.10 - Ouvrez une deuxième console, et envoyer le signal SIGKILL à la première². Donner la commande que vous avez utilisé :

1.2 Premier-plan et arrière-plan, avec xeyes

Q1.11 - Minimisez toutes les fenêtres, lancez le programme xeyes depuis la console. Déplacez votre souris et observez l'observateur.

Q1.12 - Suspendez-le en utilisant la combinaison de touches Ctrl-Z : vérifiez que son interface graphique est complètement gelée.

Q1.13 - Avec la commande ps aux filtrée sur xeyes, dans quel état est le processus xeyes ?

Q1.14 - Faites-le passer en arrière-plan avec la commande bg : vérifiez que son interface graphique est à nouveau fonctionnelle.

Avec la commande jobs, quel est son état ? :

Quelle différence aurions-nous pu observé si nous avions exécuté la commande fg ?

1. Indice : utiliser la commande find

2. Indice : man kill

2 Periphériques

2.1 Comptage

Q2.1 - Proposer une ligne de commande utilisant "find" qui indique le nombre de périphériques de type "block".³

Q2.2 - Combien en trouvez-vous :

Q2.3 - Proposer une ligne de commande qui indique le nombre de périphériques de type "character"

Q2.4 - Combien en trouvez-vous :

2.2 Periphériques type "terminal"

Q2.5 - La console courante est vue comme un périphérique, on obtient son identifiant avec la commande `tty`. Quel est le périphérique associé au terminal ouvert ?

Q2.6 - Avec un seul terminal ouvert, combien de périphériques sont visibles dans `/dev/pts`

Q2.7 - Ouvrez un deuxième terminal, et refaites le même comptage :

Q2.8 - Depuis le 1er terminal, envoyer une chaîne de caractères dans le 2ème, en faisant

`$ echo "envoi vers 2" > /dev/pts/XXX`

(en remplaçant XXX par le numéro du 2ème terminal)

Q2.9 - Fermez ce deuxième terminal. A-t-il disparu du système de fichier virtuel `/dev/pts` ?

3 Périphériques : commandes `lsXXX`

Q3.1 - Essayer la commande `lshw -short`, en lui ajoutant le filtre `less` si besoin. Comparer avec `lshw`.

Q3.2 - Ce dernier affichage étant peu lisible, ré essayer avec l'option `-html`, en redirigeant la sortie vers un fichier `/tmp/lshw.html`, puis ouvrir ce dernier avec un navigateur;

Q3.3 - Avec cette commande, on peut n'obtenir que les infos pour une certaine "classe" de hardware, avec l'option `-class`, suivie de l'un de ces items :
memory, bridge, bus, input, network, communication, multimedia, storage, system, disk, power, generic

Q3.4 - Comparer les infos fournies par `sudo lshw -class processor` avec celles-fournies par la commande `lscpu`

Laquelle est la plus complète ?

Q3.5 - Essayer l'interface graphique fournie `lshw-gtk` (à installer)

Q3.6 - Périphériques de type "block" : essayer la commande `lsblk`. Combien en trouvez-vous ?
Est-ce la même valeur que la question Q2.1

3. Indice : voir le man de `find` et chercher à "type"

4 Exploration du BIOS

Avec l'outil `dmidecode`, vous allez récupérer les informations du BIOS, via le protocole SMBIOS.

Chaque matériel référencé dispose d'un enregistrement, que l'outil permet de récupérer. Chaque enregistrement comporte un en-tête de 4 octets, plus des octets contenant des données spécifiques, ou des informations (caractéristiques) en clair.

Les 4 octets d'en-tête sont :

- un "handle", unique, sur 2 octets
- un type
- d'une taille en octets

Vous trouverez la liste des types dans le "man" : <https://linux.die.net/man/8/dmidecode>

Q4.1 - Donner la version de l'outil :

Q4.2 - Lancer l'outil (avec "sudo") et observer la structure de l'affichage. Réessayer en filtrant avec `less`.

Q4.3 - via `sudo dmidecode | head`, donner la version de SMBIOS indiquée :

Q4.4 - Cette dernière commande affiche aussi le nombre d'objets référencés par la BIOS ("Handle").
Combien y en a-t-il ?

Q4.5 - Afin de vérifier, donner la commande utilisant `wc` qui va compter le nombre de "Handle" :

Q4.6 - Donner la ligne de commande (utilisant le "grep" adéquat) qui va compter le nombre de sondes de température :

Q4.7 - L'outil "grep" dispose d'une option permettant d'afficher le "contexte" d'un match, c'est à dire au choix n lignes avant le match, ou n lignes après le match.

Donner la commande complète qui va afficher pour chaque sonde de température les 8 lignes suivantes :