TP - ASR7 Programmation Concurrente

—

Administration Système – GNU/Linux

Sylvain Brandel, Guillaume Damiand, Laurent Lefèvre, Matthieu Moy, Grégoire Pichon

Printemps 2022

Vous avez normalement fait au moins le DM0 qui introduit toutes les manipulations que nous ferons dans ce TP. Par ailleurs, en cours, on vous a demandé de lire les URLs suivantes en préparation de ce TP :

- `https://www.debian.org/doc/manuals/debian-faq/pkg-basics.fr.html`
- `https://www.debian.org/doc/manuals/debian-faq/pkgtools.fr.html`
- `https://www.debian.org/doc/manuals/debian-faq/uptodate.fr.html`

Elles contiennent les principales informations *théoriques* qu'il vaut mieux retenir... ;)

I Administration Système

Où il est « intéressant » de connaître les commandes `top source alias less tail head ps grep whoami chown chgrp passwd adduser addgroup sudo su lsmod modprobe apt-get aptitude dpkg man` et certaines de leurs options.

I.1 Création/re-création de VM

Pour ce TP, vous devez utiliser la machine virtuelle que nous vous avons créé et sur laquelle vous êtes administrateur. Vous pouvez revenir sur le TP1 pour les instructions pour vous y connecter (on supposera par la suite que `IP_VM` est son IP). Grâce au DM0, vous savez comment configurer un environnement agréable pour travailler, que ce soit sur votre compte Lyon 1 ou sur votre VM (notamment avec des alias comme `l='ls --color -l'`, un prompt `$1` défini correctement et en couleur, etc.).

I.2 Configurations d'environnement : confort d'utilisation avec la complétion "intelligente"

Vous pouvez remarquer que la complétion existe bien sur votre compte Lyon1, mais pas la complétion "intelligente". Nous avons vu dans le DM0 qu'elle était mise en place automatiquement grâce à la commande `source /etc/bash_completion` sourcée au démarrage d'un nouveau shell. Mais ce fichier n'existe pas sur les installations Fedora, les admins ont visiblement oublié de l'installer...

Q.I.1) - À partir de la VM, comment connaître son adresse IP ?

Q.I.2) - Le fichier `/etc/bash_completion` est-il le seul intéressant ? Pour le savoir, regardons quel package l'a installé (à moins que vous vous souveniez de son nom ?) : tapez `dpkg -S /etc/bash_completion`. Cela permet justement de donner le package qui a installé le fichier donné.

Q.I.3) - Comment lister tous les fichiers installés par le package `/etc/bash_completion` ?

Q.I.4) - On voit que le package installe notamment `/etc/profile.d/bash_completion.sh`. Qu'est-ce que ce fichier, et que contient-il (et pourquoi) ? Quelle différence avec `/etc/bash_completion` ?

Q.I.5) - On voit également le fichier `/usr/bin/dh_bash-completion`. Si on regarde son contenu, peut-on savoir à quoi il sert à priori ?

Q.I.6) - À l'aide de la commande scp, copiez le fichier `/etc/bash_completion` de votre VM sur votre compte Lyon1, par exemple dans `~/.bash_completion_VM`.

Q.I.7) - Sourcez-le. Est-ce que cela change quelque chose à l'environnement ? *A*ny comments ?

I.3 Retour sur les alias et fonctions

Q.I.8) - Toujours taper `ls -l --color` est long. Proposez un alias *l* pour faire la même chose, et rendez-le persistant !

Q.I.9) - La fonction suivante utilise le package `adb` pour communiquer avec un téléphone androïd. Que fait-elle ?

```
# $1 must be provide as date like 201509, 2015, or 20150926
function down_pics_from_phone {
    local photo_dir="/storage/sdcard1/DCIM/Camera"
    if [ "x$1" == "x" ] ;then
        adb shell "ls $photo_dir" > /dev/shm/list_pics2
        tr -d '\r' < /dev/shm/list_pics2 > /dev/shm/list_pics
        rm /dev/shm/list_pics2
        cat /dev/shm/list_pics
        echo List available in /dev/shm/list_pics
        echo Use $0 YYYYMMDD or subset to pull photos
        return
    fi

    adb shell "ls $photo_dir" | grep $1 > list_pics
    tr -d '\r' < list_pics > list_pics2
    for i in `cat list_pics2` ; do
        echo "Getting $i"
        adb pull $photo_dir/$i
    done
    rm list_pics list_pics2
}
```

I.4 Les utilisateurs et root – rappels d'ASR5

Q.I.10) - Avant tout, connectez-vous (ssh) sur la VM OpenStack en tant qu'utilisateur `chaprot` (reprenez l'énoncé du TP1 si besoin).

Q.I.11) - Quels sont tous les utilisateurs présents sur la machine ?

Q.I.12) - Sur votre VM, créez un autre utilisateur de login `lurong` (nom complet : Gai Luron), et donnez-lui un mot de passe différent de celui pour `chaprot` (rappel : la commande `adduser` est plus pratique que `useradd`, c'est en fait un wrapper.).

Q.I.13) - Depuis un shell appartenant à l'utilisateur `chaprot`, ouvrez un shell en tant qu'utilisateur `lurong`. Fermez ce shell pour revenir au shell de `chaprot`.

Q.I.14) - Depuis votre PC physique, ouvrez un nouveau terminal (pour conserver le shell `chaprot` ouvert), et dans ce nouveau terminal ouvrez une connexion sur votre VM en vous connectant *directement* en tant qu'utilisateur `lurong`.

Q.I.15) - Regardez le contenu du fichier `/etc/passwd` (il est accessible en lecture pour tout le monde, vous pouvez voir son contenu avec une commande comme `less` ou `cat`, ou l'ouvrir dans votre éditeur de texte). Vous devriez trouver une ligne pour l'utilisateur `lurong` contenant entre autres le nom complet que vous avez entré à la création, et d'autres informations qui ont été ajoutées automatiquement comme le répertoire personnel `/home/lurong` et le shell par défaut `/bin/bash`. Notez que le mot de passe ne se trouve pas dans ce fichier (il y a un `x` dans la colonne où il aurait pu se trouver, qui signifie qu'il est stocké ailleurs).

Q.I.16) - Regardez maintenant le contenu du fichier `/etc/shadow`. Celui-ci n'est accessible que par `root`. Dans ce fichier, il y a bien une information sur le mot de passe, mais pas le mot de passe lui-même. Pourquoi ?

Q.I.17) - Quel est l'`UID` (User IDentifier, le nombre qui identifie l'utilisateur de manière unique) de l'utilisateur `lurong` ?

Q.I.18) - Comment connaître les groupes auxquels il appartient ?

I.5 Utilisateurs et gestion des droits

Q.I.19) - Arrangez-vous pour avoir deux terminaux ouverts, l'un avec un shell appartenant à `chaprot`, l'autre avec un shell `lurong`. Si vous avez bien suivi les consignes ci-dessus, vous l'avez déjà fait. Vous devriez aussi avoir des invites de commandes différentes dans ces deux terminaux.

Q.I.20) - Dans les deux terminaux, vérifiez que vous êtes bien celui que vous pensez à l'aide de la commande `whoami`.

Q.I.21) - Vérifiez que chaque utilisateur se trouve, pour l'instant, dans son propre répertoire personnel (commande `pwd`).

Q.I.22) - Dans le shell sur la VM, sous l'identité `chaprot`, créez un fichier avec la commande `touch poi` et exécutez `ls -l`. Modifiez ce fichier avec la commande de votre choix.

Q.I.23) - Dans le shell sous l'identité `lurong`, retrouvez le fichier `poi`. Faites un `ls -l` dessus et regardez son contenu. Essayez de le modifier. Que se passe-t-il ?

Q.I.24) - Revenez sous le shell de `chaprot` et exécutez la commande `chmod go+w poi`. Réessayez de modifier le fichier en tant que `lurong` (cela devrait marcher maintenant). Faites un `chmod go-w poi` en tant que `chaprot`, et vérifier que la commande a bien coupé les droits en écriture.

Q.I.25) - Exécutez `chmod go-r poi` en tant que `chaprot` et vérifiez que les droits en lecture ont été coupés. Exécutez `chmod 644 poi` en tant que `chaprot` pour revenir à la situation précédente.

Q.I.26) - en tant que `lurong`, exécutez `chmod go+rw poi`. Vous devriez obtenir une erreur : vous n'êtes pas propriétaire du fichier, vous n'avez pas le droit de modifier ses permissions.

Q.I.27) - Ajoutez `lurong` au group `chaprot` à l'aide de la commande `usermod`, et vérifiez.

Q.I.28) - Est-ce que maintenant `lurong` peut modifier le fichier `poi` ?

Q.I.29) - Dans une console sur la VM, sous l'identité `root`, faites un `chmod go-rwx /etc/passwd`. En tant que `chaprot`, exécutez maintenant `ls -l` et expliquez ce que vous voyez (et pas ce que vous croyez voir). Remettez les bons droits au fichier `/etc/passwd`.

Q.I.30) - En local (la question n'a pas de sens sinon), comment passer en mode console ? Revenir au mode graphique ?

I.6 Les packages et leur gestion

Q.I.31) - Quelle est la liste complète des packages installés ?

Q.I.32) - Que dois-je faire si je souhaite trouver un package contenant le mot clé `keyring` par exemple ?

Q.I.33) - Faire une mise à jour de la base des packages.

Q.I.34) - Démarrer une mise à jour du système (pour ne pas perdre de temps vous pouvez annuler la mise à jour en répondant « n » à la dernière question).

Nous allons maintenant installer un serveur web sur votre VM. Un serveur web très connu est apache2, mais nous allons jouer dans un premier temps avec Nginx (à prononcer « engine-X »).

Q.I.35) - Comment savoir si Nginx est installé ?

Q.I.36) - Quel est le package concernant nginx ?

Q.I.37) - Voir si nginx est installé.
Quelle version serait installée si on décidait d'installer nginx ?

Q.I.38) - Installer le package `nginx`

Q.I.39) - Quels sont les fichiers installés par nginx ?

Q.I.40) - Voir les fichiers de configuration de nginx.

Q.I.41) - Où sont les fichiers de log de nginx

Q.I.42) - À votre avis, votre VM a t-elle un serveur ssh installé ? Comment le savoir ? Où sont ses fichiers ?

Q.I.43) - Vous pouvez regarder les packages `adb` (voir la question sur les fonctions), `french-conjugator`, `anki`, `nmap`, *etc.*

Q.I.44) - Installez le package `zmap` avec la commande `apt-get install zmap` (c'est un petit package, et ce n'est pas son installation ici qui est importante). Contrairement à `apt`, `apt-get` ne fait pas le ménage des packages téléchargés : on peut donc trouver les archives `.deb` dans `/var/cache/apt/archives/`.
Copiez le package debian qui s'y trouve dans le répertoire de connexion, et explosez l'archive.

Q.I.45) - Dans un sous-répertoire `tmp`, explosez l'archive `control.tar.gz`. Que contient-elle ? À quoi servent ces fichiers ?

Q.I.46) - Dans un sous-répertoire `pmp`, explosez `data.tar.xz` et commentez.

I.7 Gestion des services avec systemd

Nginx est un logiciel prévu pour tourner en tâche de fond sur la machine. Il est lancé au démarrage et tourne en permanence même si personne n'est connecté à la machine. On appelle cela un *démon* (daemon en anglais). Son rôle est de répondre aux requêtes HTTP reçues, donc c'est aussi un *serveur*. Ce serveur fournit un *service* aux clients.

Q.I.47) - Vérifiez que votre machine répond bien quand un navigateur web l'interroge sur le port 80. Plusieurs solutions :

- Lancer votre navigateur web habituel sur `http://IP_VM/` (cela ne marchera que si vous avez un accès direct à la VM, donc pas si vous tentez une connexion depuis l'extérieur de l'université ni depuis le wifi).
- Depuis un shell qui tourne sur votre VM, lancez la commande `links http://localhost/` (si besoin, installez `links` au préalable). Links est un navigateur en mode texte, peu convivial mais cela présente entre autres l'avantage de pouvoir être lancé facilement via une connexion SSH.
- Si vous voulez vous amuser : faire un tunnel SSH pour accéder à `IP_VM:80` depuis votre PC physique.

Vous devriez voir apparaître la page d'accueil par défaut de nginx (« Welcome to nginx ! »). Si vous le voulez, vous pouvez aussi voir et modifier le contenu de cette page dans le fichier `/var/www/html/index.nginx-debian.html`.

Q.I.48) - Interrogez systemd pour avoir le statut du service nginx avec la commande :

```
systemctl status nginx.service
```

Q.I.49) - Vérifiez « à la main » que le processus nginx tourne : `ps aux | grep nginx`

Q.I.50) - Coupez le service nginx avec la commande :

```
sudo systemctl stop nginx.service
```

Q.I.51) - Ré-essayez de charger la page web : vous aurez une erreur du type « connection refused ».

Q.I.52) - Interrogez systemd pour avoir le status du service nginx avec la même commande que ci-dessus. La ligne « Active » doit être passée de « active (running) » à « inactive (dead) ».

Q.I.53) - Vérifiez « à la main » que le processus nginx ne tourne plus : `ps aux | grep nginx`

Q.I.54) - Redémarrez le service :

```
sudo systemctl start nginx.service
```

Q.I.55) - Regardez à quoi ressemble le fichier de description du service pour systemd : `/etc/systemd/system/multi-user.target.wants/nginx.service`

Vous n'avez pas besoin de comprendre les détails, mais ce fichier contient au moins :

- La commande pour démarrer le démon (`ExecStart`)
- La commande pour arrêter le démon (`ExecStop`)
- Un descriptif (`Description`)
- Les dépendances (`After`, `WantedBy`)

Q.I.56) - Pour lister les services disponibles, lancez la commande `systemctl`. Retrouvez la ligne concernant nginx avec `systemctl | grep nginx`.

Q.I.57) - Comment savoir si le serveur ssh est lancé (c'est-à-dire, comment connaître le status du serveur ssh) ?

I.8 Les logs !

Q.I.58) - Naviguez dans `/var/log`. Quels sont tous ces fichiers, leurs droits et que contiennent-ils ?

Q.I.59) - Lancez la commande `tail -f /var/log/nginx/access.log`, puis, pendant que cette commande tourne, rechargez la page d'accueil de nginx. Vous devriez voir apparaître une ligne par requête dans le fichier de log (`tail -f` les affiche au fur-et-à-mesure).

Q.I.60) - Chargez la page `http://IP_VM/404/`. Comme cette page n'existe pas, vous verrez apparaître la page d'erreur 404 (Not Found) de nginx.

Q.I.61) - Installez maintenant le paquet `apache2`. Ce paquet correspond à la version 2 du serveur web Apache httpd. C'est un concurrent direct de nginx, qui va essayez d'écouter sur le port 80 (Ce qui est impossible pour l'instant car nginx est déjà en écoute dessus, mais faisons semblant de ne pas savoir pourquoi un instant ...).

Q.I.62) - Forcez un lancement d'apache :

```
sudo systemctl start apache2
```

Q.I.63) - Essayez de recharger la page d'erreur 404 : c'est toujours nginx qui répond !

Q.I.64) - Regardez le status du service apache2 avec `systemctl status`. Vous devriez obtenir « Active : inactive (dead) », et la commande vous donne quelques lignes de logs qui devraient vous mettre sur la voie de la raison de l'échec du lancement. Vous pouvez retrouver les détails avec la commande `journalctl`.

Q.I.65) - Devant l'échec de cette tentative de lancer deux serveurs web sur la même machine, nous décidons d'abandonner et de désinstaller apache.

II Reviser en s'amusant : le jeu de piste

Une petite exercices pour réviser la gestion des paquets et des logs : un jeu de piste. Chaque étape vous demande une manipulation, à réaliser sur votre VM, et faire cette manipulation vous donne accès à l'étape suivante, et ainsi de suite. Le point de départ est ici : `https://asr-lyon1.gitlabpages.inria.fr/prog-concurrente/jeu-de-piste.html`. Toutes les commandes utilisées pendant le jeu de piste sont au programme, vous devez les connaître pour l'examen final.

III Si le temps le permet : copie de fichiers à distance (vu en ASR5 en L2)

Vous devez apprendre à utiliser ssh pour copier des fichiers à distance. Une excellent alternative est `rsync`.

Q.III.1) - Faites le en utilisant la ligne de commande, via `scp`, `pscp` (`PuTTY`) ou `rsync`.

Certains logiciels reposant sur la bibliothèque `FUSE` (File System in User Space) permettent de configurer un système de fichiers en réseau en tant que simple utilisateur. Vous pourrez en installer et utiliser un pour associer à un répertoire du compte chaprot de la VM un répertoire que vous utilisez à l'université pour les TPs de LIF12.

Q.III.2) - Qu'est-ce qu'un disque réseau ? Donner un exemple de logiciel utilisant ce type de bibliothèque.

Q.III.3) - Installez le logiciel `sshfs`.

Q.III.4) - Regardez le manuel de la commande `sshfs`, utilisez-la pour que le répertoire `votrevm:/home/chaprot/univ/` corresponde au répertoire `"$HOME"/LIF12/VM` sur votre poste de travail de l'université.

Q.III.5) - Quel est l'intérêt de faire des systèmes de fichiers dans « l'espace utilisateur » ?

IV Pour aller plus loin

IV.1 Configuration de `sudo`

`sudo` permet d'autoriser à un utilisateur l'exécution d'une ou plusieurs commandes avec des droits privilégiés. C'est une commande **additionnelle** au système.

Q.IV.1) - Comment la machine ASR7, dont vous avez obtenu une copie pour créer votre VM, a été configurée pour donner la possibilité à `chaprot` de lancer n'importe quelle commande avec `sudo` ?

Q.IV.2) - Comment configurer `sudo` pour donner le droit à `lurong` d'exécuter la commande `cat` avec les droits `root` (afin par exemple de pouvoir afficher le contenu de `/etc/shadow`) ?

Q.IV.3) - Comment se fait-il que la commande `sudo` puisse donner le droit à un utilisateur de lancer un processus (donc à priori avec les droits propres à cet utilisateur) avec des droits `root` ?

IV.2 Les "fichiers" du noyau Linux

Q.IV.4) - Naviguez dans `/boot`, `/usr/src/`, `/lib/modules/`
Peut-on lire la même chose en local ? Quelles sont les capacités du noyau chargé ?

Q.IV.5) - Faire `dmesg`, distant et local.
Repérer le type de processeur, et de disque dur.
Faites `cat /proc/cpuinfo` et `cat /proc/meminfo`
Quelles informations obtenez-vous ?
Peut-on les obtenir en local et/ou distant ?
Du coup, quel utilisateur peut exécuter la commande ?
Comment était-il possible de le savoir avant ?

Q.IV.6) - Listez `/proc/` et commentez ce que vous y trouver.
Des remarques sur `/proc/sys/net/ipv4/ip_forward` ?

Q.IV.7) - Quels systèmes de fichiers le noyau installé est-il capable de lire ?

Q.IV.8) - Qu'est-ce que `/dev/` ?
Qu'est-ce que `/dev/shm/` ? Qu'est-ce que `/dev/disk/by-uuid/` ?

IV.3 Jouons à casser notre environnement ;-)

Comme le titre l'indique, ne pas faire ces manipulations sur votre compte habituel pour ne pas prendre de risque.

— Exécutez la commande `PATH=` (équivalente à `PATH=''`, c'est-à-dire avec une chaîne vide à droite du `=`), puis essayez `ls` et `cd`. Expliquez ce qu'il se passe.

IV.4 Préserver une session de la déconnexion avec `screen`

Un scénario assez classique :

— Un utilisateur lance une commande sur son serveur

— L'heure tourne, l'utilisateur doit rentrer chez lui

— De chez lui, l'utilisateur se reconnecte à son serveur via SSH, et aimerait reprendre la main sur ce qu'il a démarré avant de partir.

La commande `screen` (ou un de ses petits frères comme `tmux` ou le couple `dvtm/dtach`) permet de répondre à ce problème (et bien d'autres comme la possibilité de découper votre terminal en sous-fenêtres).

— Dans un terminal lancé sur votre VM, entrez la commande `screen`, et validez le message d'accueil avec Entrée si besoin.

— Entrez quelques commandes : tout se passe normalement.

— Entrez Control-a puis d. Vous devriez revenir au shell initial, mais ce que vous avez démarré dans screen tourne toujours. Si un calcul est en cours, le calcul continue.

— Entrez la commande `screen -r` : votre environnement screen est de retour.

— Refaites Control-a puis d. Déconnectez-vous complètement de la VM.

— (Pour respecter strictement le sénario ci-dessus, il faudrait rentrer chez vous ici ...)

— Reconnectez-vous via SSH et faites `screen -r` : votre environnement est encore là.

Une alternative est la commande `nohup` qui permet de lancer une commande qui survit à la fin de la session de l'utilisateur courant (mais ne permet pas de reprendre la main facilement dessus).