– TD 1 – Mise en place d'un environnement sous GNU/Linux –

Lyon 2 / ICOM / L1 informatique / Environnement Informatique et Internet

Didier Puzenat `didier.puzenat@univ-lyon2.fr`

1 Introduction

L'objectif du présent TD est de mettre en place et de tester un environnement de travail sous GNU/Linux, pour pouvoir faire les TD de l'enseignement « Environnement Informatique et Internet » mais aussi pour tous les enseignements à venir nécessitant un environnement Linux. Les machines de l'Université utilisées pour cet enseignement étant hélas sous le système d'exploitation Microsoft Windows, nous allons utiliser une machine virtuelle connue sous le nom de WSL pour « *Windows Subsystem for Linux* » ou en français « sous-système Windows pour Linux ». WLS est un outil de Microsoft déjà installé sur les machines, avec une distribution Linux Debian.

2 Premier lancement de WSL

Comme vu en cours, la notion de compte est essentielle sous Unix, aussi la première chose que va vous demander la distribution Debian est de créer un compte et de choisir un mot de passe. Sous Unix, votre identifiant est appelé « *login* » et doit être limité à 8 caractères sans accents ni caractères spéciaux, par exemple la première lettre de votre prénom suivi des 7 premières lettres de votre nom de famille, soit `dpuzenat` pour Didier Puzenat. À faire :

1. cliquer sur l'icône Windows en bas à gauche de votre écran et lancer WLS, soit en saisissant WLS dans la barre de recherche, soit en cliquant sur l'icône Debian ;
2. créer votre compte en répondant aux questions.

Vous interagissez à présent avec l'interpréteur de commandes BASH qui vous invite à saisir une commande. À faire :

1. se situer et se promener dans l'arborescence avec les commandes vues en cours (`pwd` et `cd`), pensez à utiliser des chemins absolus et relatifs, à utiliser le tilde (`~`), etc.
2. lancer l'éditeur de textes `nano` (en tapant simplement son nom) et créer un petit texte contenant quelques mots de votre choix. Les commandes de l'éditeur sont listées en bas de la fenêtre, le symbole `^` correspondant à la touche `Ctrl` de votre clavier (pour contrôle).

3 Installation de programmes

Très peu de programmes sont installés par défaut, nous allons donc en installer quelques-uns, contenus dans des paquets (ou « *packages* » en anglais), ce qui suppose d'avoir les droits de l'administrateur de la machine, grâce à la commande `sudo`. Pour savoir ce que fait cette commande, vous pouvez utiliser le manuel, en tapant `man sudo`. Il existe normalement une « page de man » pour tous les programmes installés mais aussi sur de nombreux autres sujets. Faire `man man` pour en savoir plus ! Comme vous le constatez les pages par défaut sont en anglais, aussi nous allons commencer par installer deux packages contenant les pages en français. Nous allons utiliser le gestionnaire de paquets APT (*Advanced Packaging Tool*) et plus précisément la commande `apt-get`. N'hésitez pas à faire `man apt-get`, utiliser les pages de man pour lire au moins la partie « DESCRIPTION » est une bonne habitude. Mais attention, une page de man se veut

exhaustive, il ne s'agit pas du tout d'un tutoriel, aussi lorsqu'on débute il ne faut pas chercher à tout comprendre, mais plutôt rechercher une information spécifique. Pour comprendre comment naviguer dans une page de man vous pouvez utiliser la touche « h » pour accéder à une aide. Hélas, installer les pages de man en français n'est pas suffisant, il faut aussi indiquer que vous souhaitez que les différents programmes installés vous parlent en français. Ainsi, si plusieurs utilisateurs ont un compte, chacun peut utiliser un langue spécifique. Pour cette dernière tâche, nous allons utiliser un utilitaire interactif nommé `dpkg-reconfigure`. À faire :

1. mettre à jour la liste des paquets disponibles via la commande `sudo apt-get update`, donner votre mot de passe lorsque la commande `sudo` vous le demande ;
2. mettre éventuellement à jour la distribution (facultatif) via `sudo apt-get upgrade` ;
3. lancer la commande `sudo apt-get install manpages-fr` pour installer les principales pages de man, puis `sudo apt-get install manpages-fr-extra` pour être complet ;
4. lancer la commande `dpkg-reconfigure locales` et choisissez « `fr_FR.UTF-8 UTF-8` » dans la liste proposée ;
5. consulter quelques pages de man en français, par exemple celles de `ls`, `pwd`, `sudo` et `man`, à noter qu'il n'y a pas de page pour `cd` car cette commande est directement exécutée par l'interpréteur de commandes (BASH), on parle de commandes « *built-in* » ;
6. installer le paquet `tree`, parcourir le man et jouer avec la commande `tree`. À noter que vous pouvez (presque) toujours interrompre un programme en exécution (ou « processus ») via la combinaison de touches `ctrl-c` (utile si vous essayez « `tree /` »).

À noter que votre choix d'une langue est stocké dans une « variable d'environnement ». Lorsque vous écrirez vous-même un programme sous Linux, vous pouvez consulter ces variables par exemple pour afficher un message dans la langue choisie par l'utilisateur. À faire :

1. afficher toutes les variables d'environnement en utilisant la commande `env` ;
2. retrouver la variable renseignant sur la langue.

4 Manipulation de processus

Un processus peut être dans différents états, en cours nous avons vu les états « prêt », « élu » et « bloqué ». Mais en pratique c'est un peu plus compliqué. La commande `top` permet de visualiser les processus qui sollicitent le plus le processeur. L'état de chaque processus est indiqué par la colonne « S ». À faire :

1. tester la commande `top`, sortir avec la touche « q » (même si `ctrl-c` fonctionne aussi) ;
2. consulter la page de man (qui risque d'être uniquement disponible en anglais) pour trouver les différents états que peut avoir un processus sous GNU/Linux ;
3. essayer de deviner à quoi correspond l'état « zombie » (Z).

Comme vu en cours, chaque processus s'exécute au nom – et avec les droits – de l'utilisateur qui l'a lancé, un mécanisme qui contribue grandement à la sécurité du système. Enfin, chaque processus possède un numéro unique appelé « PID » pour « *Process IDentifier* ». À faire :

1. essayer de créer un répertoire nommé `tmp` dans le répertoire `/root` (c'est à dire le répertoire `root` situé à la racine de l'arborescence), avec la commande `mkdir /root/tmp` ;
2. essayer de nouveau avec `sudo mkdir /root/tmp` ;
3. expliquer pourquoi la seconde commande a fonctionné (relire le cas échéant le début de la page de man de `sudo`).

Il peut arriver que vous souhaitiez arrêter un programme immédiatement, sans attendre qu'il ait terminé son traitement. Nous avons vu que c'est (le plus souvent) possible grâce à la combinaison de touches `ctrl-c`, mais c'est également possible (sans réserve) en lui envoyant un signal précis grâce à la commande `kill`, en précisant le PID du processus. Enfin il est possible de simplement suspendre un processus grâce à la combinaison `ctrl-z`, par exemple parce que vous voulez réserver les ressources de l'ordinateur (processeur, mémoire, disque, etc.) pour une autre tâche tout en gardant la possibilité de reprendre le traitement ultérieurement là où il en était. Vous pouvez relancer le dernier processus suspendu grâce à la commande `fg` pour *foreground* (premier plan) ou `bg` pour *background* (arrière plan). Un programme qui s'exécute en premier plan ne permet pas de lancer un nouveau programme dans le même interpréteur avant la fin de son exécution, contrairement à une exécution en arrière plan. Enfin, vous pouvez directement lancer un programme en arrière plan en terminant la ligne de commandes par le symbole « `&` ». À faire :

1. lancer la commande « `tree /` », suspendre l'exécution, lancer puis quitter le programme `top`, puis relancer le processus suspendu en premier plan, et enfin le tuer ;
2. idem en relançant la commande suspendue en arrière plan, tout se passe-t-il comme prévu lorsque vous voulez tuer ce processus ? Pourquoi ? Comment faire ?

5 Interface graphique

Le système GNU/Linux est un système extrêmement complet qui n'a rien à envier à des systèmes comme Microsoft Windows ou Apple MacOS. Hélas WSL est bien plus limité et ne comprend pas d'interface graphique, sans doute parce que Microsoft ne souhaite pas que les utilisateurs utilisent trop des applications Linux sous Windows. Heureusement, il existe de nombreuses alternatives à WSL, comme installer GNU/Linux à la place de Windows, ou « à côté » auquel cas un menu permet de choisir quel système utiliser au démarrage de la machine. Hélas ces deux solutions sont un peu techniques pour un étudiant de première année, notamment sur une machine récente comportant du matériel non encore supporté. Une troisième solution, assez proche de ce que Microsoft propose avec WSL, est d'installer une machine virtuelle complète. Avec un processeur ad hoc, cette solution peut être très performante, de plus l'installation est simple et sans danger. Un logiciel gratuit (mais non libre) permettant d'installer une distribution Linux complète, notamment sous Windows, est Oracle VM VirtualBox. Il est néanmoins possible de lancer les programmes nécessitant une interface graphique sous WSL, en installant un « serveur X » sous Windows. Expliquer proprement ce qu'est et comment fonctionne un serveur X nécessiterait un cours complet, aussi le reste de cette fiche ne fait que donner les commandes nécessaires pour être opérationnel, mais n'hésitez pas à vous renseigner en ligne si vous être intéressés. À faire :

1. installer un serveur X sous Windows (si un autre étudiant ne l'a pas déjà fait), par exemple « VcXsrv » disponible à l'URL https://sourceforge.net/projects/vcxsrv/ ;
2. lancer XLaunch (sous Windows), avec la valeur 0 dans le champ « Display number » ;
3. trouver l'adresse IP de la machine (on verra ce que c'est dans un prochain cours en amphi), vous pouvez trouver cette adresse via Windows ou un installant le paquet `net-tools` sous GNU/Linux puis en lançant la commande `ifconfig` ;
4. éditer votre fichier `.barshrc` (faire `nano ~/.barshrc`) et ajouter `export DISPLAY=IP:0` à la fin, où IP est l'adresse IP de votre machine trouvée à l'étape précédente ;
5. installer une application nécessitant un environnement graphique, par exemple un éditeur avec `sudo apt-get install gedit` ou un terminal pour jouer avec la commande `kill` ;
6. lancer le programme, en arrière plan pour garder la main dans l'interpréteur (`gedit &`).