ASTUPS - CampusFab
LECHALUPÉ Julien

Cours d'initiation à l'utilisation de Linux sur Raspberry Pi

Université Paul Sabatier Novembre 2015

Sommaire

Remerciements :

Jean-François ERDELYI pour son aide sur la partie « Qu'est-ce que l'Open Source ».

Raphaël LALLEMENT pour son aide sur la structuration du cours.

David TSANG HIN SUN (Décembre 2014) et à Erwan ALARY (Novembre 2015) pour la diffusion du cours auprès du Fablab la mise à disposition d'une salle de cours pour l'initiation.

Introduction

Le Raspberry Pi est un ordinateur mono-carte (ou micro-pc) low-cost créer par David Braben en 2006. Il possède les mêmes fonctionnalités qu'un ordinateur, mais pour un prix (de 25€ à 40€ selon les versions) et une taille (celle d'une carte bleue) bien inférieure. Ces deux principaux atouts la rendent très appréciée de la communauté robotique et DIY (Do It Yourself). Son cout très bas et sa polyvalence permet de l'utiliser dans bon nombre de projets comme des centres multimédia, des serveurs web mais aussi des projets embarqués, comme des caméras de surveillance, du contrôle de drones, etc.

Autre avantage du Raspberry Pi : il est libre de droit, autant du côté logiciel (Open Source : il utilise un système d'exploitation Linux) que du côté matériel (Open Hardware). C'est-à-dire que tous les documents de spécification, conception et développements sont disponible sur internet et réutilisable !

Cette initiation a pour but de présenter la carte Raspberry Pi et d'introduire à son utilisation. Dans un premier temps nous nous intéresserons à l'histoire de la carte ainsi qu'à ces performances. Dans un deuxième temps, nous nous intéresserons aux systèmes d'exploitation Linux et au monde du logiciel libre. Nous poursuivrons avec la présentation d'un système d'exploitation pour Raspberry Pi (Raspbian, voir tutoriel d'installation sur le site officiel : http://www.raspberrypi.org/downloads/). Nous rentrerons dans le vif du sujet grâce à SSH (Secure Shell). Et nous terminerons par la présentation et l'utilisation des commandes Linux les plus fréquentes.

I) Présentation de la Raspberry Pi

A) L'origine du projet

Le Raspberry Pi a été imaginé en 2006 par David Braben un créateur de jeux vidéo. Le but initial du projet étant « *d'encourager la jeunesse à programmer* » (cf. *Wikipédia : Raspberry Pi*). Le premier prototype était de la taille d'une clé USB et était équipée d'un port USB d'un côté et d'un port HDMI de l'autre. Suite à la réalisation de ce prototype, la fondation de David Braben opte finalement pour deux modèles différents :

- Un modèle à 25 $ US, appelé modèle A.
- Un modèle à 35 $ US, appelé modèle B.

Finalement, les premiers prototypes du modèle B sont réalisés en 2011. Les résultats sont très encourageants, un système d'exploitation linux est installé sur la carte et a pu lancer la lecture d'une vidéo HD (1080p) ou encore un test de performance d'OpenGL ES (API d'affichage 3D).

La première version du modèle B grand public est finalement commercialisée début 2012. C'est un grand succès commercial, à peine six mois écoulé depuis l'ouverture des ventes au grand public, près de 500 000 unités ont été vendues. Le modèle A fera son apparition un peu plus tard (début 2013).

Suite au succès du Raspberry Pi, plusieurs évolutions des produits ont été effectuées : augmentation de mémoire et possibilité d'overclocker (augmenter la rapidité du processeur), d'autres modèles plus adapté à la demande (modèle A+ et B+). Et plus récemment une seconde version du Raspberry Pi fait son apparition.

B) Fiche technique

Suite aux nombreuses évolutions de la Raspberry Pi, pas moins huit modèles ont été commercialisés. Bien que certaines versions ont des caractéristiques très différentes, la majorité des composants utilisés sont les mêmes. Parmi les composants que l'on retrouve sur tous les modèles, il y a :

- La carte graphique (ou GPU), fabriqué par Broadcom, il est capable de décoder des vidéos Full HD (1080p) et supporte OpenGL ES 2.0.
- Un port HDMI pour une éventuelle sortie vidéo.
- Des ports USB pour connecter les périphériques.
- Une sortie audio Jack 3,5 mm.

D'autres composants sont présents selon le modèle de Raspberry Pi. On trouve notamment un nombre plus grand de ports USB et un port Ethernet. Voici un tableau récapitulatif des composants par modèle :

Composant	Modèle A	Modèle A+	Modèle B	Modèle B+	Modèle 2
Processeur	ARM11 mono-cœur, 700Mhz (overclockable jusqu'à 1Ghz)				Quadricœur ARM Cortex-A7900, 900Mhz
Mémoire vive	256 Mo	256 Mo	256-512 Mo	512 Mo	1 Go
Port USB	1	1	2	4	4
Port Ethernet	Non	Non	10/100 Mb/s	10/100 Mb/s	10/100 Mb/s
Sortie vidéo	Composite et HDMI.	HDMI seulement	Composite et HDMI	Jack et HDMI.	Jack et HDMI.
Port SD	SD/MMC	Micro SD	SD/MMC	MicroSD	MicroSD
Ports GPIO	8	17	8	17	17

Les ports GPIO sont des broches d'entrées/sorties multifonctions. Elles peuvent être utilisées pour générer des signaux électriques pour contrôler de moteur ou l'ouverture de volet, mais aussi

vérifier si un interrupteur a été appuyé, etc. Plus généralement, ils peuvent être utilisés pour piloter ou mesurer des signaux électriques numériques.

Enfin, le Raspberry Pi doit être accompagné d'une alimentation externe micro USB (chargeur de téléphone ou tablette) ainsi que d'une carte SD (ou micro SD) pour embarquer le système d'exploitation et autres données. L'alimentation doit pouvoir délivrer un courant d'environ un Ampère.

II) Linux et le monde du Libre

A) Le libre

Un logiciel libre est un programme informatique dont la diffusion et la modification (partielle ou globale) sont permise. Dans un terme plus technique : quiconque a accès à son code sans (presque) aucune restriction (Open Source).

Attention cependant à ne pas mélanger libre et gratuit : en effet ces deux aspect bien que souvent liées, ne désignent cependant pas la même chose. La confusion viens du fait qu'en anglais, *libre* et *gratuit* sont des homonymes (*free*). Un programme libre (dont les sources sont ouvertes) n'est pas forcément gratuit, a contrario, un logiciel dit propriétaire n'est pas forcément payant !

On ne peut pas parler de logiciel libre sans parler de licence de ces logiciels. Il existe plusieurs licence du domaine du libre, notons cependant 3 plus connue à ce jour :

GNU/GPL (GNU is Not Unix / General Public License)

Cette licence est dite *contaminante.* En effet si un projet quelconque utilise une partie ou la totalité d'un logiciel sous licence GPL, la totalité de projet doit être GNU/GPL. Cette licence, initialement conçue pour protéger le monde du libre, est cependant critiquée par ce dernier pour son côté peu permissif dans certain cas, comme par exemple la censure de certaines parties brevetées.

BSD (Berkeley Software Distribution)

Il s'agit d'une des licences les moins contraignantes. En effet cette licence ne protège pas contre les restrictions. Il impose seulement que le nom de l'auteur initial soit présent sur le projet terminal. Petit bémol : certains y voient un aspect non protecteur pour le libre. En effet, rien n'empêche un industriel de récupérer du code sous licence BSD et de le revendre dans un produit avec une licence propriétaire.

Creative Commons

Une sorte de licence à la carte composée de six combinaisons autour de quatre pôles suivant l'usage de votre projet terminal :

- *Attribution* : signature de l'auteur initial (ce choix est obligatoire en droit français) (sigle : BY)
- *Non Commercial* : interdiction de tirer un profit commercial de l'œuvre sans autorisation de l'auteur (sigle : NC)
- *No derivative works* : impossibilité d'intégrer tout ou partie dans une œuvre composite ; l'échantillonnage (*sampling*), par exemple, devenant impossible (sigle : ND)
- *Share alike* : partage de l'œuvre, avec obligation de rediffuser selon la même licence ou une licence similaire (version ultérieure ou localisée) (sigle : SA)

Choisir de créer un logiciel libre peut être motivé par de nombreuses raisons : l'intérêt du partage ou à but pédagogique. On retiendra cependant plus l'envie de pouvoir construire un lot d'outil pouvant être utilisé sans restriction, pouvant être amélioré par quiconque souhaite y contribuer. Même les grandes sociétés ont bien compris les nombreux avantages que le libre comporte : en effet la communauté du libre a prouvé qu'elle pouvait fournir de très bon résultat. Par exemple Google, géant du web, a lancé Chromium, une sorte de Chrome libre afin que les communautés libre aide au développement de Chrome.

Impossible de parler du libre sans vous parler des plus grand succès de celui-ci. Si vous ne deviez en connaitre qu'un, cela serais le système d'exploitation Linux. En effet, l'informatique telle que l'on la connaît actuellement doit beaucoup au développement de Linux. Voici aussi une liste non exhaustive d'autres projets libres, largement utilisés dans les domaines professionnel ou non : Libre Office, MySQL, Firefox, Apache, The Gimp, VLC, ...

Tout comme les logiciels libres, il existe des « matériel libre » (Open Hardware). C'est-à-dire que toute l'architecture matérielle du produit peut être consultée et réutilisée. Un des exemples les plus connus d'Open Hardware est l'Arduino. Il existe de nombreux projets dérivés et inspiré des cartes Arduino, sans pour autant qu'elle soit produite par Arduino Software.

En plus de pouvoir être réutilisés, les produits Open Source et Open Hardware peuvent être contrôlés, par exemple pour éviter les contrefaçons. On pourra aussi veiller à ce que le produit soit conforme, en termes de qualité mais aussi de sécurité.

B) Linux et Raspberry Pi

Il existe plusieurs « versions » (distributions) de Linux sur Raspberry Pi. Elles possèdent toutes des avantages et des inconvénients, mais elles ont toutes la même base : le noyau Linux. Cette base commune permet de pouvoir utiliser tous les systèmes Unix (base du noyau Linux) de la même manière. Ainsi, un grand nombre d'opérations élémentaires peuvent être retrouvée sur les distributions Linux sur Raspberry Pi.

Il existe un bon nombre de distribution Linux compatible avec les processeurs ARM et Raspberry Pi. Ceux disponible sur le site officiel permettent d'avoir une certaine garantie de fonctionnement. On retrouve notamment *Raspbian* (dérivée de *Debian*), ou encore *PiDora* (dérivée de *Fedora*). D'autres distributions ne sont pas directement développées pour Raspberry Pi mais pour sa famille de processeur (ARM), ces distributions sont donc compatibles avec le Raspberry. La plus connues est sans doute *ArchLinux ARM* qui est une dérivée d'*ArchLinux*.

Dans le cadre de ce cours, nous utiliserons Raspbian pour des questions de facilité d'installation et d'utilisation. Toutefois, le contenu de ce support de cours peut être appliqué à n'importe quel autre système Linux !

III) Introduction à l'utilisation de SSH

A) Le protocole SSH

La problématique de l'utilisation des commandes Linux sur un Raspberry Pi est l'utilisation d'un clavier. En effet, il est très vite compliquer de mettre un clavier proche de son Raspberry lorsque l'ont fait des applications embarquée (un drone par exemple).

Le problème est d'ailleurs plus ancien, lorsque les ordinateurs n'étaient pas accessibles au grand public et était particulièrement cher pour l'entreprise et les universités. Le besoin d'un accès à distance à une machine capable de faire des calculs s'est vite posé. Un premier protocole de contrôle à distance a vu le jour en 1968 : Telnet. Il a longuement été utilisé jusqu'à l'apparition des problèmes de sécurité réseau et plus généralement d'internet. Les tentatives de piratage sont alors nombreuses à cause de la faible sécurité de Telnet. En 1995 un nouveau protocole est né : le Secure Shell (ou SSH).

Le SSH reprend les grands principes du Telnet (pouvoir exécuter des commandes sur un ordinateur distant). Cependant, il ajoute une couche de cryptage des paquets réseaux. Lors de la connexion avec un poste distant, il crée un tunnel de communication sécurisé permettant d'envoyer et de recevoir des données ainsi que des commandes de façon sure.

B) Comment utiliser SSH ?

Tous les systèmes dérivés d'Unix (Linux et Mac entre autres) sont déjà équipés d'un client SSH. Il n'y a donc pas besoin de l'installer si vous utilisez une de ces deux plateformes. Dans le cas où vous utilisez Windows, il vous faudra installer PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) ou tout autre client SSH pour Windows.

SSH fonctionne grâce à une architecture *client-serveur* (comme un site web, de nombreux clients pour un seul serveur). Il faut donc s'assurer qu'un serveur SSH est bel et bien actif sur Raspberry Pi. La plupart des distributions l'activent par défaut (il suffira alors de se connecter avec les identifiants par défaut) mais ce n'est pas le cas de Raspbian. Raspbian propose, lors du premier

lancement, de configurer le Raspberry Pi (il est aussi possible de le faire plus tard grâce à la commande *raspi-config ; Voir Figure 1*).

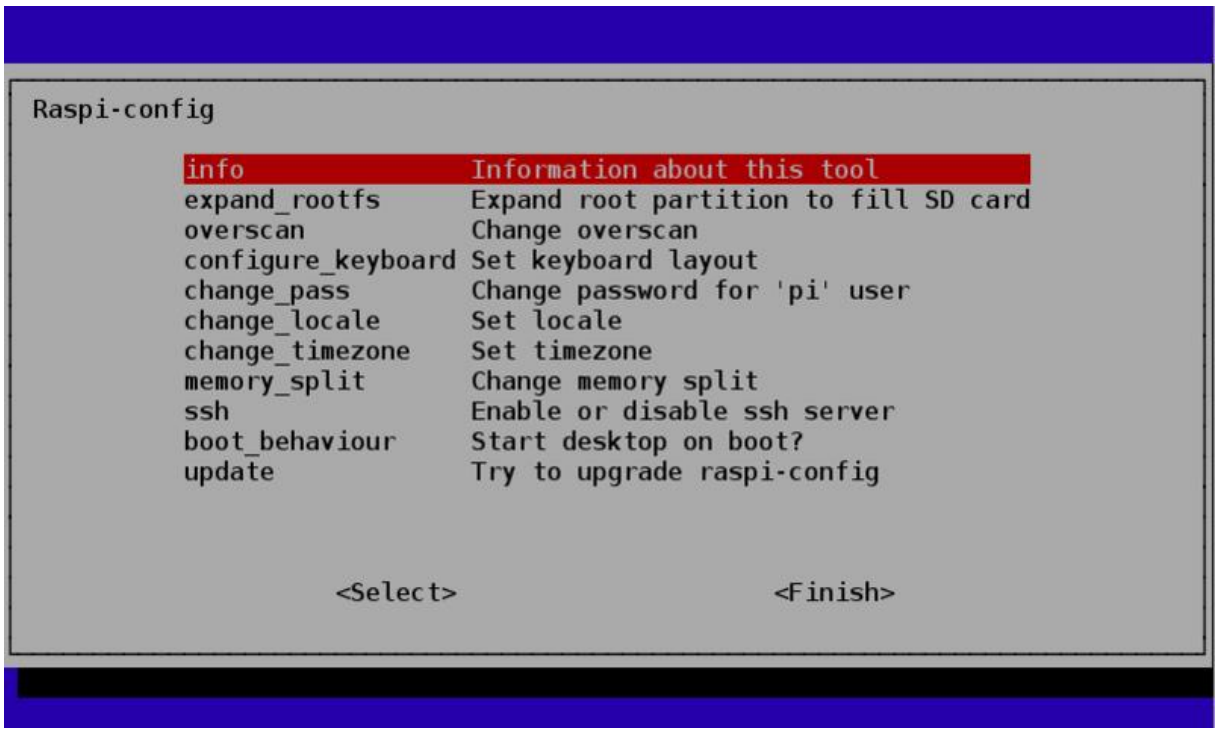

Figure 1. Raspi-config

En ouvrant le menu SSH il est possible d'activer le serveur SSH. Il se lancera alors au démarrage du Raspberry, nous permettant ainsi de pouvoir y accéder par le réseau.

Une fois le serveur activé, il est possible de se connecter via SSH sur le Raspberry Pi. Pour les utilisateurs de Linux ou de Mac, il suffit de lancer un invité de commande et de lancer la commande suivante :

ssh user@hote par exemple : *ssh martin@raspberry*

Pour les utilisateurs de Windows (avec PuTTY) au lancement de PuTTY vous obtiendrez une fenêtre représentée Figure 2. Il vous suffit de taper *user@hote* dans le champ *Host Name.*

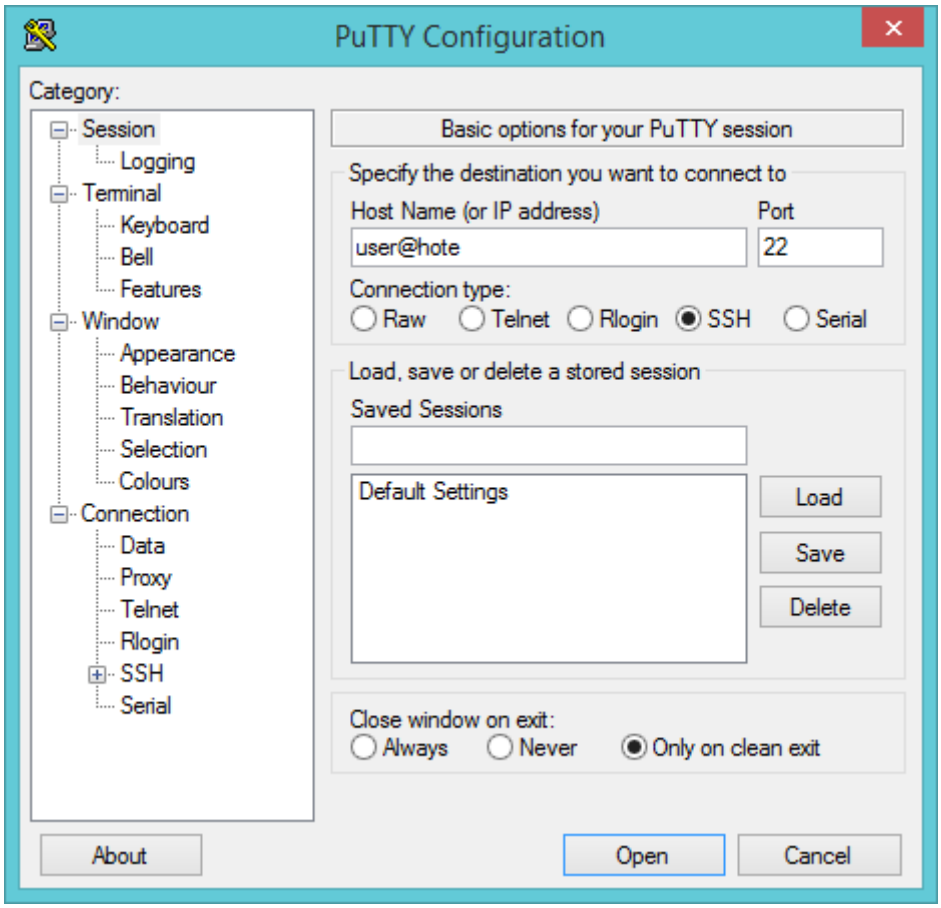

Figure 2. PuTTY

Que ce soit pour Windows ou les systèmes Unix, champ *hôte* correspond à l'identifiant réseau du Raspberry Pi. Cela peut être représenté par le nom donné à la carte (par défaut *raspberrypi*) ou son adresse IP.

Pour les identifier si vous disposez d'un écran branché sur votre Raspberry, exécutez la commande *ifconfig (voir Figure 2)*. On remarque dans l'encadré [1], à la deuxième ligne, un champ *inet addr* qui correspond à l'adresse IP sur votre réseau local. L'encadré [2] représente le nom d'hôte (ou nom de machine) de votre Raspberry Pi.

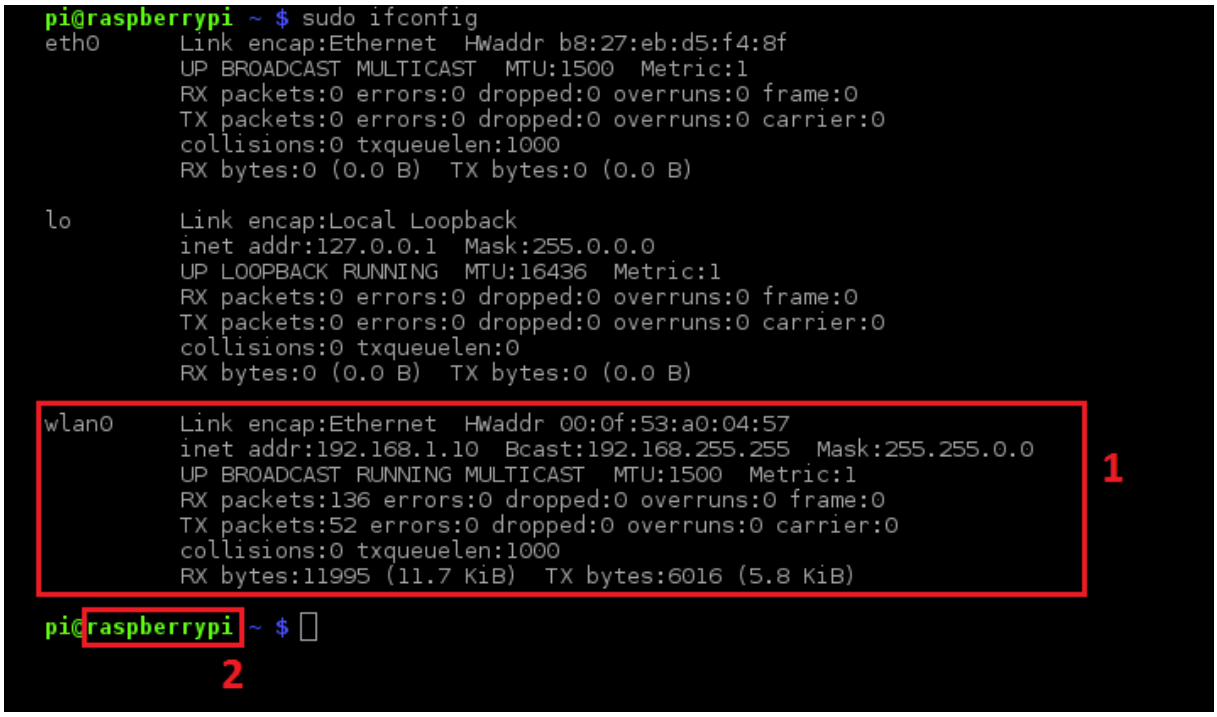

Figure 3. *ifconfig*

Dans le cas où vous n'avez pas d'écran connecté, vous pouvez accéder à l'interface de votre routeur/box et rechercher une liste des appareils connectés. Vous pourrez alors y retrouver le nom de votre Raspberry Pi ainsi que son adresse IP associé.

Le champ *user* correspond à l'utilisateur avec lequel on va se connecter sur le Raspberry Pi. L'utilisateur par défaut est *pi*.

Une fois la commande exécutée, il suffira dans rentrer le mot de passe de l'utilisateur pour se connecter sur le Raspberry Pi.

IV) Contrôle de la Raspberry Pi grâce aux commandes UNIX

A) Un peu d'aide ?

Une des commandes les plus essentielles sur Linux est *man*. Le *Man* (comprendre le « manuel ») est un annuaire répertoriant toutes les commandes d'un système Linux et leurs utilisations. Ainsi, lorsque l'on souhaite connaitre le comportement ou les paramètres d'une commande, on exécutera la commande *man* suivis du nom du programme à exécuter (par exemple : « *man ls* »).

Le Man possède une mise en forme (ou plutôt une syntaxe) particulière. Les crochets sont utilisés pour spécifier un champ facultatif dans la commande. Si l'on reprend l'exemple de la commande *ls* on obtient ceci : **ls** [OPTION]… [FILE]…

Les points de suspension indiquent que l'on peut rajouter plusieurs paramètres du même type (par exemple ici des options) à la suite. Ainsi on peut avoir une commande *ls* construite de cette façon : *ls –l –a* ou encore *ls –la*

Chaque champ souligné est considéré comme obligatoire. Attention, cela peut être combiné avec les crochets ! Par exemple, l'option *--color* de la commande *ls* peut être accompagnée d'un filtre. On aura alors la syntaxe suivante : [--color[=WHEN]] qui peut être traduit par : l'option facultative *–color* **peut** être suivie par un symbole = lui-même suivis par un filtre (WHEN) qui lui est **obligatoire** (mais seulement si on met le symbole =).

Enfin, un dernier point concernant la syntaxe, mais cette fois ci des commandes elles-mêmes. Certains symboles sont utilisés sur toutes les commandes. Parmi eux, deux sont seulement présent dans le terminal à titre indicatif c'est le # et le $. Le # informe que l'on va exécuter une commande alors que l'on est l'utilisateur courant, tandis que le $ informe que l'on va exécuter une commande en mode super utilisateur (ou administrateur).

Un autre symbole, est lui très utilisé dans les commandes courantes, c'est le * (joker). Il permet d'indiquer à la commande que l'on souhaite que la suite d'un mot soit complétée automatiquement. Par exemple, si l'on souhaite supprimer tous les fichiers d'un répertoire on pourrait utiliser la commande *rm ** ou encore supprimer tous les fichiers finissant par *.txt* comme ceci : *rm *.txt*. Attention à l'utilisation du joker, il peut être très dangereux dans certains cas puisque qu'il n'est pas explicite.

B) Se déplacer dans l'arborescence des fichiers

Une des actions de bases sur tous les systèmes d'exploitation est la navigation au sein d'un système de fichier. Le système de fichier Linux se présente comme une arborescence, c'est-à-dire qu'un dossier contient d'autres sous-dossier qui contiennent eux même des fichiers etc. Le dossier d'origine d'un système Linux (appelée la *racine*) à pour chemin (ou adresse) *« / »*. L'adresse d'un fichier ou d'un dossier commence donc systématiquement par « */* » par exemple « */home* », cette représentation s'appelle un chemin **absolu** (car référencé depuis la racine).

Il existe aussi des chemins relatifs, qui eux dépendent du dossier dans lequel l'utilisateur se trouve. Par exemple, si l'on suppose que l'on se situe dans le dossier « */home* » et qu'il contienne un fichier « *info.txt* », le chemin absolu de ce fichier est « */home/info.txt* » et le chemin relatif **par rapport** à « */home* » est « *info.txt* » (cette fois ci sans le « */* ») ou encore « *./info.txt* » (où le « *./* » représente le répertoire courant).

Deux commandes sont principalement utilisées sous Linux pour se déplacer dans l'arborescence. La première est *ls* (abréviation de *list* pour *list directory*) qui permet d'afficher le contenu d'un dossier. Voici sa syntaxe : **ls** [OPTION]… [FILE]…
Il existe diverses option pour modifier l'affichage de cette commande (par exemple l'option « *-l* » pour afficher le contenu sous forme de liste). Si aucun dossier n'est spécifié, cette commande affichera le contenu du répertoire courant, sinon, c'est le contenu du dossier spécifié qui sera listé.

La deuxième commande est *cd* (abréviation de *change directory*) permet de changer le répertoire où se situe l'utilisateur. Elle s'utilise comme ceci : **cd [-L | -P] [**directory**]** (comprendre « soit **–L** ,soit **–P** » pour **[-L | -P]**). Si aucun dossier n'est spécifié, l'utilisateur retourne automatiquement dans son répertoire personnel (généralement */home/nomUtilisateur*, référencé

comme « **~** »). Il est possible de spécifier n'importe quel type de chemin (relatif ou absolu) à cette commande.

Enfin une dernière commande peut être utile lorsque que l'on navigue dans l'arborescence des fichiers, c'est *pwd*. Elle permet d'afficher le répertoire dans lequel se trouve actuellement l'utilisateur (elle s'utilise le plus souvent sans paramètres).

C) Droits des fichiers

Le système Linux étant un système multi-utilisateurs, il convient de protéger les accès des fichiers sensibles aux uns et aux autres. Un utilisateur lambda ne doit pas avoir accès aux fichiers de configuration créés par l'administrateur par exemple. Il existe donc un système de droits sur les fichiers, indiquant quel utilisateur à accès à un fichier précis. Pour mettre en place ce mécanisme, les utilisateurs ont été regroupés et différenciés dans des **groupes**.

Les droits des fichiers sont de trois natures : lecture, écriture et exécution. On peut attribuer ces droits à trois catégories de personnes :

- Le propriétaire du fichier (celui qui l'a créé) ;
- Les utilisateurs du même groupe que le propriétaire du fichier ;
- Les autres utilisateurs.

On obtient ainsi une combinaison de trois caractères pour chaque catégorie, **r** pour la lecture, **w** pour l'écriture et **x** pour l'exécution. Sachant qu'il y a trois catégories, on obtient une combinaison de 9 caractères symbolisant les droits d'un fichier. Par exemple :

Propriétaire	Groupe	Autres
rwx	**r-x**	**r--**

Ici on a le propriétaire qui a tous les droits possible, son groupe qui a des accès en lecture et en exécution et enfin les autres utilisateurs qui n'ont que le droit de lire le fichier. On peut aussi représenter les droits des fichiers grâce à trois chiffres (cf. *man chmod*).

D) Liste de commandes utiles

Maintenant que vous connaissez les généralités sur les commandes Linux, voici une liste (non exhaustive) des commandes les plus utiles sur Linux et Raspberry Pi.

Syntaxe	Utilité
touch [-acm][-r REF_FILE**| -t** TIME**]** FILE**...**	Créer un fichier, change la date de dernière modification d'un fichier existant.
mv [-fi] SOURCE_FILE**...** TARGET_FILE	Déplace (ou renomme) un fichier
cp [OPTION]... SOURCE... DIRECTORY	Copie un fichier
rm FILE...	Supprime (définitivement) un fichier
rmdir [-p] DIR**...**	Supprime (définitivement) un dossier vide
mkdir [OPTION]... DIRECTORY...	Créer un dossier
nano [FILE]	Éditeur de texte en console
vim [OPTIONS] [FILE ...]	Éditeur de texte en console
cat [-u][FILE ...]	Affiche le contenu d'un fichier dans la console
tail [OPTION]... [FILE]...	Affiche les dernières lignes d'un fichier
less [FILENAME]	Affiche un fichier au fur et à mesure
chmod [OPTION]... MODE[,MODE]... FILE...	Change les droits d'un fichier. Le mode correspond au droit du fichier (par exemple *chmod rwxrxr fichier.txt*)
scp [[USER@]HOST1:]FILE1 ... [[USER@]HOST2:]FILE2	Copie un fichier depuis (ou vers) un ordinateur distant (par exemple un Raspberry Pi). Si vous souhaitez copier un fichier vers votre Raspberry Pi, utilisez *scp* comme ceci : *scp /home/fichieOriginal utilisateur@raspberryPi:/home/fichierCopie* Dans le cas où vous souhaitez prendre un fichier de votre Raspberry Pi : *scp utilisateur@raspberryPi:/home/fichierOriginal /home/fichieCopie*

Conclusion

On peut conclure en disant que le Raspberry Pi est un ordinateur mono carte très accessible et polyvalent. Il peut être utilisé dans bon nombre de projets qui vont de l'application multimédia à la robotique en passant par la domotique. Il faut cependant garder à l'esprit qu'il existe d'autres ordinateurs mono cartes ayant les mêmes utilités mais avec des caractéristiques différentes (plus puissant, moins cher, etc.).

Le Raspberry Pi utilise une version embarquée du système d'exploitation Linux, ce qui lui permet de garantir la compatibilité avec un grand nombre de logiciel que l'on trouve généralement sur nos ordinateurs classique. De plus l'aspect libre de la carte (que soit coté logiciel ou matériel) rend son utilisation et son développement plus rapide, plus confortable et plus fiable. Il est aussi possible de contrôler le Raspberry Pi à distance via le protocole SSH qui permet d'exécuter des commandes Linux à distance.

Toutefois, toutes ces commandes Linux ne sont pas dédiées au Raspberry Pi, elles peuvent être utilisées sur n'importe quel autre système Linux. On peut ainsi administrer un Raspberry Pi de la même manière qu'un serveur ou qu'un poste de bureau.

Contact : lechalupe.julien@gmail.com