Première partie : Installer un système d'exploitation et le configurer

Document 1 : Qu'est-ce qu'un raspberry pi ? Le Raspberry Pi est ce qu'on appelle un nano-ordinateur. De la taille d'une carte de crédit, il est équipé du strict nécessaire : Choice of RAM [2GB] [4GB 1GB Ì Le processeur qui anime le More powerful système Raspberry Pi 4 est un processor processeur Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64bit SoC @ 1.5GHz SoC (System-on-Chip, tout le système sur un seul circuit). Cela signifie que la plupart des USB-C composants du système (unité Power centrale, coprocesseur supply GIGABIT graphique, matériel audio et ETHERNET vidéo) sont réunis dans un seul composant. Il est accompagné de 2GB de RAM USB 3 MICRO HDMI PORTS LPDDR4-2400 SDRAM Supporting 2 × 4K displays USB 2

Le processeur BCM2711 utilise un jeu d'instructions machine (ISA, Instruction Set Architecture) nommé ARM. (ARMv8)

Le langage machine est le seul langage qu'un processeur puisse exécuter.

Il y a deux grandes familles de processeurs, selon la complexité de son langage machine :

- processeurs CISC (Complex Instruction Set Computer). Exemple : Pentium

- processeurs RISC (Reduced Instruction Set Computer). Exemples : PowerPC, MIPS, BCM2711

Document 2 : Les périphériques

Les indispensables :

Une alimentation

La carte SD sera le disque dur de votre Raspberry Pi, il est donc obligatoire d'en avoir une pour pouvoir installer votre système d'exploitation. Qu'est-ce qu'une carte SD Noobs ? New Out Of Box Software (NOOBS) Cette carte Noobs contient l'installateur de plusieurs systèmes d'exploitation .

Les très utiles pour un premier démarrage :

Le câble HDMI / ou l'écran tactile 7" vous servira simplement à avoir un affichage Un clavier et une souris pour entrer les commandes.

Document 3 : LES DIFFÉRENTS SYSTÈME D'EXPLOITATION POUR RASPBERRY PI

Le système d'exploitation (ou OS pour *Operating System*) est le premier programme exécuté lors de la mise en marche de l'ordinateur. Il offre une suite de services généraux qui facilitent la création de logiciels applicatifs et sert d'intermédiaire entre ces logiciels et le matériel informatique. Un système d'exploitation apporte commodité, efficacité et capacité d'évolution, permettant d'introduire de nouvelles fonctions et du nouveau matériel sans remettre en causes les logiciels.

Le Raspberry Pi a été conçu dès le départ pour faire fonctionner un système d'exploitation appelé GNU/Linux, que nous abrégeons dans la suite en Linux, symbole même du logiciel ouvert.

Composantes du système d'exploitation :

Le système d'exploitation est composé d'un ensemble de logiciels permettant de gérer les interactions avec le matériel. Parmi cet ensemble de logiciels, on distingue généralement les éléments suivant :

Le noyau (Kernel) représentant les fonctions fondamentales du système d'exploitation telles que la gestion de la mémoire, des processus, des fichiers, des entrées-sorties principales, et des fonctionnalités de communication. L'interpréteur de commande (en anglais « shell », traduisez « coquille » par opposition au noyau) permettant la communication avec le système d'exploitation par l'intermédiaire d'un langage de commande,



Le système de fichier (en anglais « file system » FS), permettant d'enregistrer les fichiers dans une arborescence.

Première partie : Installer un système d'exploitation ~30 min

Avant de démarrer la Raspberry Pi, il faut préparer une carte SD et y installer la totalité du système d'exploitation.

Insérez donc la carte SD contenant Noobs dans votre Raspberry Pi et démarrez-le.

Votre Raspberry Pi devrait donc booter sur l'utilitaire NOOBS et vous avez une liste d'OS disponibles. Choisissez Raspbian et cliquez sur « Install ».

Répondez aux questions suivantes pendant que Raspbian s'installe sur votre carte SD (10/15 min).

- 1) Pourquoi un logiciel conçu pour les ordinateurs de bureau et portables PC, n'est-il pas directement compatible avec le Raspberry Pi ?
- 2) Quel élément, en contact avec le processeur sur une carte mère d'un ordinateur, est absent du Raspberry Pi ?
- 3) Pourquoi les systèmes Microsoft Windows et Apple Mac OS X sont-ils dits « fermés » ?
- 4) Exprimer en quelques lignes la philosophie « Open-source ».

Dans les systèmes d'exploitation, on retrouve deux modes d'utilisation du processeur : le mode utilisateur et le mode privilégié.

- 5) Quelle est la principale différence entre le fonctionnement dans les deux modes ?
- 6) Le noyau du SE tourne dans quel mode, à votre avis ?

Au premier démarrage de Raspbian, vous devriez voir apparaître l'écran appelé « raspi-config ». Si besoin changer la langue du clavier etc ...

Deuxième partie : Gérer les droits et permissions d'accès aux fichiers. ~1h30

Quitter raspi-config et ouvrir une fenêtre de terminal

qu'est-ce que Bash?

Un shell est un processeur de commandes qui vous permet de manipuler votre ordinateur en tapant des commandes. Les commandes sont tapées après "l'invite", qui est un court extrait de texte vous informant de qui et de l'endroit où vous vous trouvez sur le système:



Bash est le nom du langage de programmation et du "shell" qui alimente l'interface texte de votre Raspberry Pi. Chaque fois que vous tapez une commande dans le terminal, vous parlez Bash.

L'humble shell (sh : Bourne Shell) remonte à 1989 et, avant que les ordinateurs ne disposent d'interfaces graphiques, c'était le seul moyen d'interagir avec eux. C'est forcément très puissant et flexible. bash : Bourne Again Shell. Une amélioration du Bourne Shell, disponible par défaut sous Linux et Mac OS X.

Windows a également son shell (cmd.exe) : fenêtre d'invite de commande

Par exemple, vous pouvez utiliser ls (list) pour répertorier le contenu d'un répertoire, mkdir (make directory) pour créer un répertoire ou cd (change directory) pour modifier le répertoire en cours. Remarque : Aucune de ces fonctions n'est bash et Bash ne peut pas faire ces choses tout seul. Ce sont des fichiers exécutables qui résident dans des emplacements spéciaux de votre ordinateur, tels que /bin ou /usr/bin (où bin signifie "binaire" et non pas corbeille!), Afin que Bash sache les exécuter lorsque vous les appelez par leur nom. Aidez-vous de ces notes et de la commande man pour répondre aux questions suivantes. *Gardez la trace de votre travail et de vos réponses*.. (Astuce : la flèche vers le haut vous permet de rappeler les commandes que vous avez tapées plus tôt.)

Lancez le terminal :

- 1) Affichez les utilisateurs connectés en utilisant les commandes who, w et finger.
- 2) À l'aide de man, trouvez la commande qui peut vous donner le uptime de ariel (le temps depuis qu'on l'a redémarré la dernière fois).
- 3) Affichez le répertoire courant.
- 4) Dans votre répertoire « home » (accessible en faisant cd), créez un répertoire cours, et un sous-répertoire nsi à l'intérieur de ce répertoire. Ajoutez un répertoire cours1 à l'intérieur de nsi. C'est ici que nous créerons nos fichiers d'exercice aujourd'hui.
- 5) Déplacez-vous dans ce répertoire. Créez-y un répertoire trash et détruisez-le ensuite.
- 6) Déplacez-vous dans le répertoire /usr/bin et listez-en le contenu. Trouvez les options de ls pour lister les fichiers en ordre lexicographique inverse.
- 7) Listez tous les fichiers dans /usr/bin commençant par t et se terminant par i.
- 8) Placez-vous dans votre répertoire cours/nsi/cours1 sur votre compte et créez-y un petit fichier texte appelé test avec le programme nano. Copiez-y toutes les questions. Sauvegardez votre fichier.
- 9) Comptez le nombre de lignes du fichier avec l'utilitaire approprié.
- 10)Copiez votre fichier test sous le nom test2.
- 11)Renommez test2 en test3. Utilisez my pour ceci.
- 12)Détruisez (attention) le fichier test3.
- 13)Toujours à l'aide de man, trouvez comment créer un fichier zip appelé test.zip qui contiendra les fichiers test et test2.
- 14)Détruisez les fichiers test et test2. Êtes-vous capable de restaurer vos fichiers test et test2 en dézippant votre fichier test.zip ?
- 15)Toujours dans le même répertoire, faites 1s -1 et observez les droits des fichiers et des répertoires. Ils sont classés dans l'ordre {utilisateur, groupe, autres}, r étant le droit de lecture, w le droit d'écriture et x celui d'exécution.
- 16)Ajoutez à votre groupe le droit de lecture du fichier test avec la commande chmod.
- 17)Restaurez les anciens droits d'accès.
- 18)Avec la commande cat, more ou less voyez le contenu du fichier /usr/include/utime.h.

Pour quitter un terminal, faites simplement exit.