

OSAE - BuildRoot

Jérémy COUTURE et Jean CAYLUS

Chaque question sera répondue dans un paragraphe

Utilité des modules

- `sed` : (Stream Editor) est un éditeur de texte non interactif, permet de modifier un flux de donnée
- `make` : est un outil pour faciliter la compilation, avec des templates et un tracker de version
- `binutils` : est un ensemble d'outils pour manipuler des binaires de bas niveaux
- `gcc` et `g++` : sont des compilateurs de respectivement du langage `C` et `C++`
- `bash` : (Bourne Again Shell) est un *shell* compatible POSIX
- `patch` : commande qui permet d'appliquer des différences/modifications à des fichiers. Utilisé en général avec la commande `diff` et inclus dans `git`
- `gzip` et `bzip2` : sont des outils pour (dé)compresser des fichiers `.zip` et autres fichiers compressés
- `perl` : permet de lancer et utiliser le langage **perl**
- `tar` et `cpio` : permettent de manipuler des archives `.tar`
- `python` : permet d'utiliser le fameux langage **python**
- `unzip` : permet de dézipper des `.zip`
- `rsync` : est `scp` mais en mieux. Ajoute des barres de chargements, la reprise de téléchargements, synchronisation etc
- `wget` : est un utilitaire de téléchargement (comme `curl`)
- `libncurses-dev` : est une bibliothèque pour créer des interfaces graphiques dans le terminal
- `libssl-dev` : est une bibliothèque qui intègre des outils pour manipuler **OpenSSL** (et donc les protocoles `TLS` et `SSL`)

Utilité des fichiers copiés dans le BOOT

- `MLO` : premier *bootloader*, il charge les composants matériels de base avant de donner la main au boot loader secondaire (`u-boot.img`)
- `u-boot.img` : boot loader secondaire (universal boot), il lance une ligne de commande pour démarrer le noyau Linux, et gère les périphériques avant de passer le contrôle à l'OS (via `zImage`)

- `zImage` : est une image compressée du noyau Linux, elle est chargée par le boot loader secondaire, elle prend le contrôle une fois décompressée
- `uEnv.txt` : définit les variables d'environnement utilisées par le *u-boot*
- `am335x-boneblack.dtb` : définit les périphériques utilisés pour permettre un lancement sans code ni détection de tous les périphériques

La commande :

```
sudo tar -C /media/enseirb/ROOTFS/ -xf output/images/rootfs.tar
```

permet d'extraire l'archive dans la destination voulue (ici dans le ROOTFS), les options `-x` pour extraire et `-f` pour définir le fichier à extraire.

ttyUSB0 vs ttyACM0

Le second est un port standardisé, il possède donc son driver etc, contrairement à `ttyUSB0` qui est une version générique, inconnue et donc n'a **pas** de driver intégré.

TFTP

Pour **Trivial File Transfer Protocol** est une version simplifiée du protocole sécurisé `FTP` (donc de transfert de fichiers) : il n'utilise pas de système d'authentification ou de chiffrement, mais le rend très utile sur des interfaces simples qui ne nécessitent pas de mode sécurisé (on ne passe pas par internet dans notre cas).

Installation et création du service TFTP

D'abord on installe tous les paquets nécessaires :

```
sudo apt-get install tftp tftpd xinetd
```

puis on crée le fichier de configuration qui va définir les paramètres de la liaison, via `xinetd` un service de création de démons simples. Les arguments sont expliqués ci dessous :

- `socket_type` : type de socket. `dgram` utilise le protocole `udp`
- `user` : le user
- `protocol` : protocole utilisé par le service
- `wait` : comportement du service face aux threads (`yes` pour mono thread)
- `server` : chemin et nom du server
- `serverargs` : les arguments passés au server (dans notre cas `-s /tftpboot` indique le chemin du dossier où seront cherchés les fichiers tftp)

Enfin, on redémarre le service `xinetd` en envoyant le signal `SIGHUP` via `killall`. Puis on le start.

Usage du réseau interne

On utilise le mode *bridged* pour que la VM puisse accéder au réseau local et donc être atteignable comme la beaglebone.

Pour cela nous avons configuré le fichier `/etc/networks/interfaces` avec les données indiquées dans l'énoncé (`192.168.1.2` pour la VM et `192.168.1.50` pour la carte).

`TFTP` permet d'échanger sans chiffrement, sans mot de passe, sans authentification. Les commandes sont simples et vont chercher dans le dossier défini par le server. Pour cela, on peut utiliser des chemins relatifs.