

I.1 – Vocabulaire et concepts

Christophe BLAESS - janvier 2020

Le projet Yocto utilise une terminologie assez spécifique et peu intuitive (*recipes*, *layers*, *distro*, *bitbake*...) que nous allons examiner rapidement ici.

Yocto est un environnement de production (*build system*) permettant de produire des **images** de systèmes Linux embarqués.

Image et contenu

Une **image** est simplement un gros fichier que l'on peut transférer dans la mémoire persistante du système embarqué (*eMMC*, *SD*, *Flash*, disque...) afin de faire fonctionner l'environnement Linux.

Contenu

Une image regroupe plusieurs éléments que l'on retrouve dans la plupart des systèmes embarqués :

- **Un *bootloader*** (chargeur) comme U-Boot, Barebox, Grub qui est lancé dès la mise sous tension du système et s'occupe de placer le noyau Linux en mémoire et de l'exécuter.
- **Le noyau (*kernel*) Linux**, véritable cœur du système d'exploitation qui pilote les ressources matérielles (processeur, mémoire, périphériques) pour les mettre à disposition des applications. Sur certaines architectures, le noyau est accompagné d'un ***device tree***, un fichier de description du matériel présent lorsque ce dernier n'est pas détectable automatiquement.
- **Le *root filesystem*** souvent abrégé *rootfs*, une arborescence contenant l'ensemble des fichiers nécessaires pour le fonctionnement du système (exécutables, bibliothèques, fichiers de configuration, scripts de lancement, fichiers de données, etc.). Le *rootfs* est généralement représenté par une archive «.tar» ou une image binaire du contenu de la partition principale.

En outre, Yocto permet de produire **un dépôt (*repository*) de packages binaires** que l'on peut installer par la suite sur un système existant (à la manière des distributions Linux habituelles) mais cette particularité n'est pas très exploitée pour les systèmes embarqués courants.

Bitbake et recettes

Pour produire une image, Yocto s'appuie sur un **moteur de compilation** nommée **Bitbake**. Cet outil, initialement inspiré de la distribution Gentoo, a été intégré dans Open Embedded, puis dans Yocto.

On peut l'imaginer comme une sorte de make très évolué, qui lit des fichiers de recettes (*recipes*) et les interprète pour construire les différents éléments du système embarqué. Une **recette** est un simple fichier de texte (avec l'extension « **.bb** » pour évoquer Bitbake) contenant les méthodes et les éléments de configuration nécessaires pour compiler et installer un package.

Il existe des recettes pour réaliser de nombreuses tâches, depuis la compilation du moindre petit utilitaire jusqu'à la génération d'un système complet, en passant par la configuration et compilation d'un noyau Linux.

Layers et distributions

Pour pouvoir organiser les recettes et les manipuler de manière aisée, on les regroupe en *layers*. **Un layer** est un répertoire muni d'un préfixe particulier (« meta - ») contenant un fichier de description et des recettes aux domaines d'action connexes.

Pour produire un système, il est nécessaire de savoir quelles recettes compiler. En effet un *layer* peut contenir plusieurs dizaines d'applications, et toutes ne nous intéressent pas forcément. On précise le contenu désiré dans une recette d'**image**.

La liste des images proposées, des *layers* disponibles, des recettes sélectionnées et leur configuration par défaut sont regroupées dans **une distribution (*distro*)**. La distribution de référence proposée par le projet Yocto s'appelle **Poky**.

Une chose importante : Yocto nous propose (et nous impose même un peu) de travailler proprement, en ne modifiant jamais le contenu d'un *layer* téléchargé. Pour modifier l'influence d'une recette fournie par un *layer*, nous aurons la possibilité de la **surcharger** (dans un fichier **.bbappend** indépendant), afin de changer son interprétation par bitbake. Cette approche nous permettra de garantir la pérennité dans le temps de notre travail de production. Nous reviendrons ultérieurement sur

ce comportement.

Dans les prochaines séquences, nous utiliserons `bitbake` pour produire un système minimal pour plusieurs cibles embarquées, en utilisant les *layers* par défaut de Poky. Pour certaines cibles, il nous sera nécessaire de télécharger des *layers* en complément de ceux fournis initialement par Yocto.



Ce document est placé sous licence [Creative Common CC-by-nc](#). Vous pouvez copier son contenu et le réemployer à votre gré pour une utilisation non-commerciale. Vous devez en outre mentionner sa provenance.



««

sommaire

»»