

LINUX/GNU

ÉVALUATION DES AVANTAGES ET DES
INCONVÉNIENTS DE YOCTO PAR RAPPORT À
BUILDROOT

YOCTO

Colin BAUMGARD

Paul-Antoine LE
TOLGUENEC

May 3, 2020



Contents

1	Introduction	2
2	Yocto: pour qui ? pourquoi?	3
2.1	Les outils disponibles pour construire son OS aujourd'hui . . .	3
2.2	Yocto : un fonctionnement plus flexible	4
3	Construire son OS avec Yocto	5
3.1	First step	5
4	Discussions et Conclusion	5

List of Figures

1	crédit: Yocto project	4
---	---------------------------------	---

List of Tables

1 Introduction

Dans le monde de l'embarqué, il est souvent utile de créer son propre système d'exploitation. En effet, les contraintes imposées par le système sont diverses et variées en fonction du projet. La construction d'un OS peut être longue et délicate. C'est pourquoi différents framework sont nées pour faciliter cette construction d'OS personnalisé. Celui auquel nous allons nous intéresser est le projet Yocto. La création d'un système d'exploitation personnalisé sur Linux conçu pour les spécificités d'une carte embarquée et les exigences du futur produit peut s'avérer difficile, mais le projet Yocto a pour mission d'aider. Le projet Yocto est un projet collaboratif en exploitation libre géré par la Fondation Linux. Mais la question qu'il convient de se poser est que dois-je utiliser pour l'OS de mon système. Dans cet article, nous allons tenter d'évaluer les avantages et inconvénients des différents projets en fonction des différents systèmes. Puis nous allons synthétiser le déroulement de la construction de l'OS pour les deux projets. **Repo GitHub**

<https://github.com/Paul-antoineLeTolguenec/Linux-GNU.git>

2 Yocto: pour qui ? pourquoi?

Pour construire son OS aujourd'hui, il existe de nombreux outils. Un outils (ou plutôt ensemble d'outils) très utilisé est buildroot. Mais il en existe plein d'autre. Dans le monde de l'industrie le projet Yocto est de plus en plus présent et nous allons voir pourquoi.

2.1 Les outils disponibles pour construire son OS aujourd'hui

Comme nous l'avons dit, les outils permettant la création d'OS sont multiples. Voici une synthèse des principaux :

Do It Yourself : limité aux cas simples

- Avantage = maîtrise totale
- Inconvénient = il faut tout faire

Buildroot : simple mais fonctionnellement moins riches que les autres

- Adapté aux applications enfouies, pas très riches
- Difficile de travailler en différentiel : régénération complète du File System, pas de gestion de paquets
- Basé sur des Makefiles

Scratchbox : riche mais obsolète

LTIB : outil utilisé par Freescale, mais changement en cours au profit du Yocto Project

- Versions logicielles datées (host + target)

OpenEmbedded : Ancêtre commun issu du projet Open Zaurus, toujours actif.

- Base de distributions variées

Et il en existe encore bien d'autre. Mais alors avec cette forte diversité pourquoi choisir YOCTO ? Les avantages sont multiples bien sûr, mais le plus important est son côté modulable. Aussi, ce projet permet de mettre en place de manière industrielle des outils de création de distribution Linux embarqué. C'est donc un projet opromisé pour le monde de l'industrie. En effet, du fait de l'évolution des composants d'un système il faut avoir une certaine flexibilité sur l'OS que l'on conçoit. Yocto permet cette flexibilité. De plus, le projet Yocto est sous l'égide de la Linux Foundation ce qui permet une certaine garantie de fonctionnement. On voit donc un peu plus pourquoi Yocto est apprécié des industriels. Mais comme nous l'avons vu ce projet ce distingue notamment de par sa flexibilité. Nous allons maintenant expliquer comment cette flexibilité est permise.

2.2 Yocto : un fonctionnement plus flexible

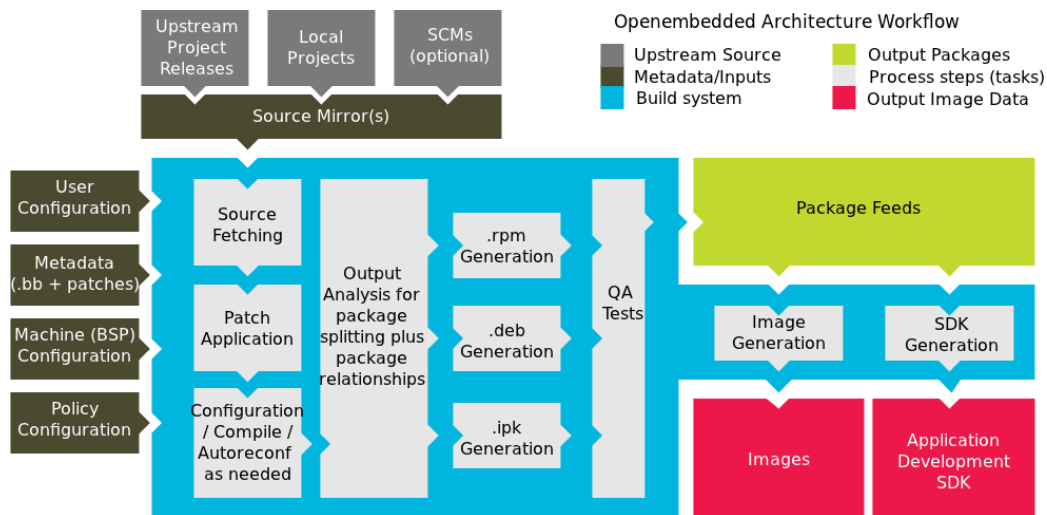


Figure 1: crédit: Yocto project

Yocto intègre des parties développées conjointement pour OpenEmbedded, notamment BitBake, OpenEmbedded-Core et d'autres métadonnées. Les éléments développés dans le cadre du projet, appelés "meta-yocto" et "meta-yocto-bsp", comprennent l'intégration des éclipses. Ensemble, ils améliorent les outils d'OpenEmbedded, cette plateforme de référence pour la construction de systèmes avancés embarqués dans les HW est connue sous le nom de Poky.

Pour développer des logiciels, nous avons besoin d'une chaîne d'outils (croisés) : les fichiers sources et les instructions sur la façon de les compiler. C'est suffisant pour une source. Pour plus de composants et de dépendances dans la compilation et le temps d'exécution, il faut augmenter la complexité et des étapes supplémentaires. Bitbake est un agent ayant la capacité d'interpréter et d'exécuter les recettes d'amélioration, il calcule la chaîne des tâches nécessaires pour développer l'objectif défini et exécuté.

3 Construire son OS avec Yocto

3.1 First step

4 Discussions et Conclusion

ecris ici

References