



Chapitre 2

Méthodologie de développement Linux embarqué

Contenu

- Présente les concepts du développement d'un système Linux du point de vu logiciel
 - Déroulement d'un projet Linux embarqué
 - Environnement de développement
 - Outils de développement

Étapes d'un projet Linux embarqué

1. Définition des **fonctionnalités du système**
2. **Choix de la plateforme** matériel
3. Création de **l'environnement de développement**
4. Configuration et construction du **noyau**
5. Construction et configuration du **système de fichiers racine**
6. Configuration du **chargeur d'amorçage**
7. Développement de **pilotes ou d'applications** supplémentaires



Définition des fonctionnalités du système

- Particulièrement important dans un projet Linux embarqué
 - Déterminant pour le choix du matériel
 - Un changement du matériel peut entraîner des modifications importantes des composants logiciels
 - Diversité et évolution rapide des logiciels Open Source
 - Sans cahier de charge précis il est très facile de se perdre dans l'évaluation permanente de nouveaux logiciels
- Méthodes:
 - Cahier de charges, spécification
 - Use cases



Choix de la plateforme

-




- Con
 - P
 - C
- Chc plat

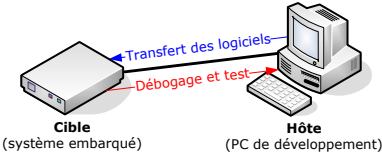




à la


Jürgen Ehrensberger IICT/HEIG Cours MAR 2-Développement 5

Environnement de développement

- Typiquement on ne développe pas directement sur la cible
 - Manque de ressources
- Développement sur un hôte
 - **Compilation croisée**
 - **Transfert des logiciels**
 - **sur la cible lors de tests**



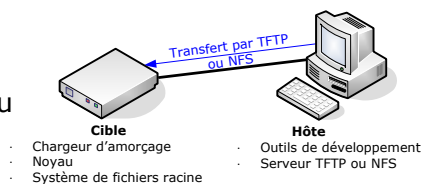

Jürgen Ehrensberger IICT/HEIG Cours MAR 2-Développement 6

Transfert hôte - cible

Transfert du système développé sur la cible

- Connexion Ethernet

- Le chargeur d'amorçage de la cible permet normalement d'utiliser les interfaces réseau et TFTP



- Connexion série

- Transfert de fichiers sur liaison série

- Transfert par un dispositif de stockage amovible

- Disque dur ou mémoire flash amovible
- Inséré et configuré sur l'hôte



Débogage

- Nécessite souvent l'exécution sur la cible
 - Matériel différent sur l'hôte (processeur, interfaces)
- Mais les outils de débogage se trouvent souvent sur l'hôte
 - Cible sans moniteur ou clavier
 - Ressources trop limitées pour installer un débogueur sur la cible

Outils de débogage

- printf ou printk
- GNU debugger GDB, DDD
 - Client sur la cible, serveur sur l'hôte
 - Via l'interface série ou Ethernet
- Débogueur hardware (JTAG)



Outils de développement

C/C++

- Compilateur, linqueur et bibliothèques C et C++
 - Appelés « *Toolchain* »
 - **Compilation croisée**
 - Le compilateur est exécuté sur l'hôte mais génère du code pour la cible (processeurs différents !)
- Compilateur GNU GCC
 - Standard de facto pour le développement Linux
 - Libre et relativement stable
 - Capable de compiler le noyau Linux
- Bibliothèque standard C
 - Implémentation des fonctions comme printf
 - Bibliothèque standard C de GNU : GLIBC



Outils de développement (2)

Java

- Principalement pour le développement d'application et de services
- Problèmes
 - ~~Taille importante de la machine virtuelle~~
 - ~~Le JRE de Sun (Oracle) n'est disponible que pour x86~~
- Utilisation sur plateformes non-x86 / embarquées
 - Oracle Java Embedded
 - Projet Blackdown : Java Linux
 - Utilise les sources de Sun
 - JRE et JDK portés sur plusieurs architectures
 - Compilateur GNU Java (GCJ)
 - Génère du code exécutable, sans machine virtuelle
 - Machines virtuelles libres
 - Kaffe
 - JamVM



Outils de développement (3)

Python

- Beaucoup de paquetages orientés réseaux et services
 - SMTP, HTTP, Telnet, XML-RPC, ... dans la librairie standard de Python
 - Paquetages pour sniffer et analyser le trafic (comme Wireshark)
 - Réalisation de services complexes avec Twisted Python
 - SMTP, HTTP, IRC, SSH, Radius, LDAP, ...
- Permet l'interfaçage avec l'OS à bas niveau

mais ...

- Compilation croisée de l'interpréteur Python est difficile mais possible
- La librairies standard de Python est large
 - Il est possible d'omettre les modules non nécessaires



Outils de développement (4)

Perl

- Souvent utilisé pour la programmation Web
- ~~Ne permet pas de compilation croisée~~
- Versions réduites et portables
 - Microperl et Miniperl

Ada

- Utilisation similaire à C/C++
- Compilateur croisé GNU GNAT



Outils de développement (5)

C#

- Framework .NET disponible pour Windows CE et Windows XP embarqué
- Multiplateforme
- Environnement Open Source : **Mono**
 - Disponible pour beaucoup de plateformes
 - Presque 100% compatible avec .NET
 - Performances inférieures à .NET

