

Migrer vers un environnement open source

Pierre Ficheux (pierre.ficheux@openwide.fr)

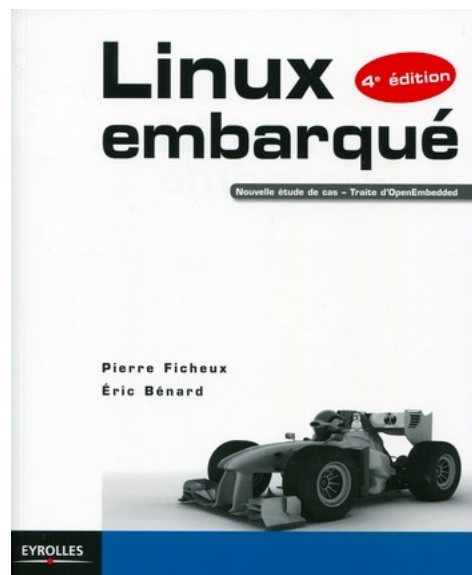
Octobre 2012

- Présenter les « meilleurs » choix possibles lors de la migration d'un système embarqué propriétaire vers un environnement « open source »
 - Matériel
 - Choix du système d'exploitation
 - Aspects légaux (licences)
 - Outils
 - Sous-traitance
 - Recrutement
 - Formation
- Système embarqué = matériel ET logiciel !

- SSII/SSL créée en septembre 2001 avec Thales et Schneider
- Indépendante depuis 2009
- Environ 100 salariés sur Paris, Lyon et Toulouse
- Industrialisation de composants open source
- Trois activités :
 - **OW** Système d'Information (Java/PHP)
 - **OW** Outsourcing: hébergement
 - **OW** Ingénierie: informatique industrielle



- Ingénieur Arts et Métiers + Sup'Aéro
- Utilisateur de logiciels libres depuis 1989
- Utilisateur de Linux depuis 1992
- Auteur des 4 éditions de l'ouvrage « Linux embarqué » (éditions Eyrolles), 4ème édition parue en juin 2012
- Auteur GNU Linux Magazine, Open Silicium
- CTO Open Wide Ingénierie, enseignant EPITA, ENSEIRB

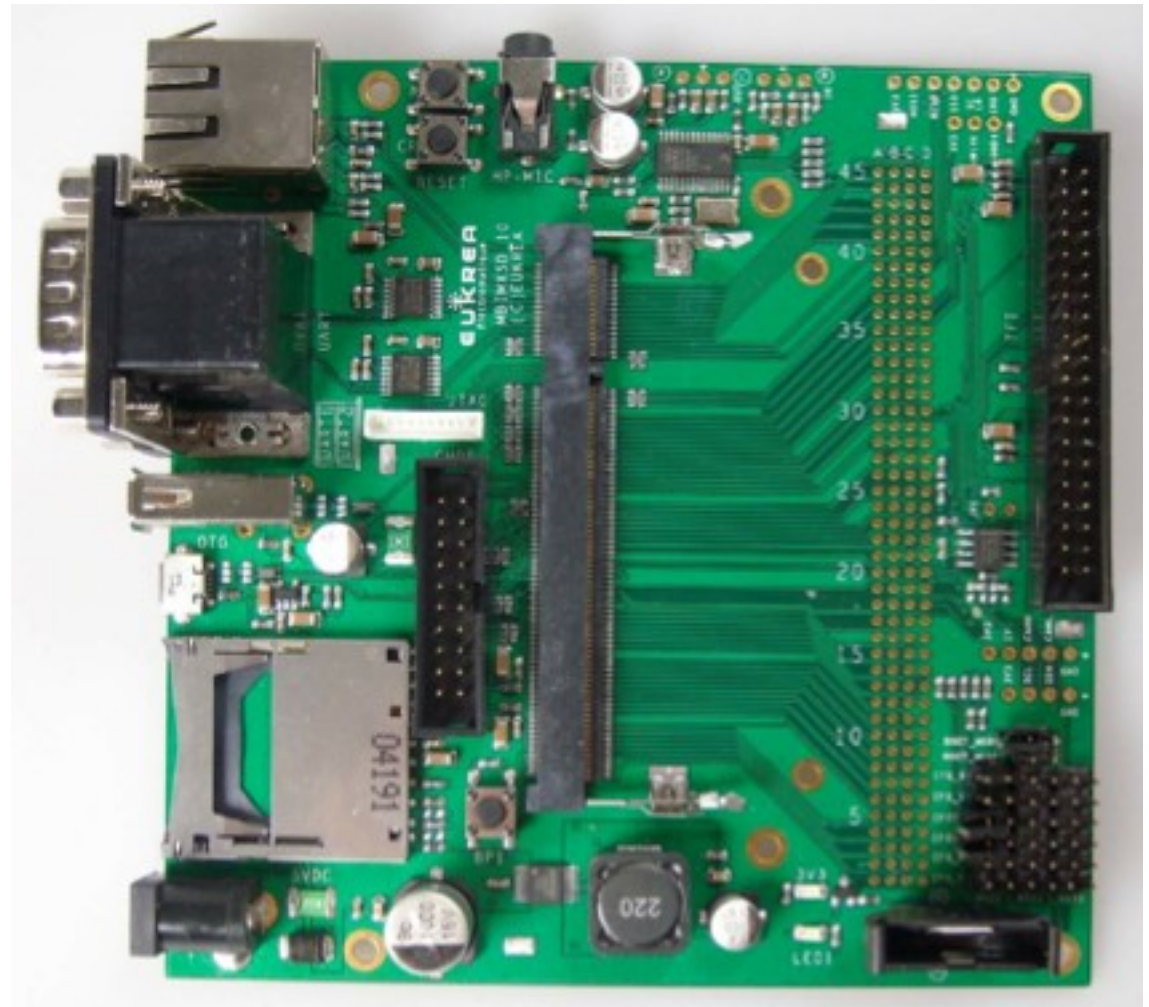


- Compatibilité matérielle → utilisation d'un « vieil » RTOS incompatible avec un nouveau matériel
- Compatibilité logicielle → support de nouvelles fonctionnalités (réseau?)
- Coût de *runtime* et maintenance (ou discontinuité)
- Marché du travail ou de la sous-traitance → trouver un développeur pSOS ou VRTX (?)
- Choix imposés par un donneur d'ordre (exemple : Orange)
- Rénovation d'un produit
- ...
- Bref, on ne migre pas un SE sans raison !

- En général on ne vend pas un logiciel mais un système complet
- Le matériel a une influence très importante sur la réussite du projet
 - Fiabilité
 - Compatibilité logicielle (pilotes)
 - Support
 - Pérennité
- Un choix hasardeux uniquement basé sur le coût immédiat peut avoir des conséquences *dramatiques*
- Si l'ancien matériel est conservé il détermine le choix de l'OS (entre autres)

- Ne pas confondre carte de « hobbyiste » et carte industrielle → éviter les FriendlyARM, Raspberry, etc.
 - Une FriendlyARM 6410 (ARM11) coûte 160€ sur Internet, sans support
 - La même carte conçue en France coûte environ 200€ (Eukréa, Calao, ARMadeus, ...)
 - De nombreux SE sont de « petites » séries
- Privilégier les constructeurs qui :
 - garantissent la pérennité des composants (Freescale : 15 ans)
 - fournissent un support logiciel de qualité → utilisation du noyau Linux « mainline » et d'outils standards (Buildroot, OE)
 - proposent un support local (Europe, France)

- Dans la plupart des cas : ARM ou x86
- x86 :
 - ☺ : support logiciel, simplicité d'utilisation, + d' outils
 - ☹ : consommation, coût
- ARM
 - ☺ : consommation, modularité, encombrement
 - ☹ : support logiciel (parfois noyau non standard), - d'outils de mise au point (Valgrind)
- On ne conçoit (presque) plus une carte mère → privilégier l'approche « modules » (SODIMM) interchangeables qui permet l'évolutivité de la carte (ARM9, ARM11, Cortex, ...)
- Utilisation des SoC (FPGA) → Armadeus



- Un OS était-il utilisé ?
- Un OS est-il nécessaire ?
- Un OS est nécessaire lorsque le système est :
 - Communicant (protocoles réseau)
 - Susceptible d'évoluer
 - Multi-fonction, de nombreux périphériques
- En cas de conservation du matériel, attention à :
 - l'empreinte mémoire, exemple OS21 → Linux
 - la compatibilité matérielle (discontinuité du support)
- Concevoir une nouvelle carte n'est pas si onéreux avec un module, si l'on conserve les périphériques métier

- Linux n'est pas le seul OS industriel libre :
 - empreinte mémoire (16Mo RAM, 4/8 Mo flash)
 - mise au point complexe (mémoire)
 - Certification difficile (15 M lignes pour le noyau ☹)
 - Absence de TR en standard
- Autre possibilités
 - RTEMS (400 K lignes de codes pour l'ensemble ☺, quelques dizaines à quelques centaines de Ko d'empreinte flash)
 - eCos, FreeRTOS
 - Disposer d'une API POSIX !
- Attention au support, formation, recrutement → Linux est le meilleur choix sur ces points
- Attention aux licences

« Embedded is not special » (D Woodhouse, Intel)

« be part of the community »

- Garder le contact avec les communautés (cf les grands éditeurs du logiciels libre) → reverser
- Choisir des communautés actives et pérennes :
 - RTEMS
 - PREEMPT-RT
 - Xenomai
 - OE
 - Buildroot

- La plupart des OS libre ne sont pas « domaine public »
→ licences GPL, LGPL, BSD, ...
- Les pilotes Linux sont en théorie GPL
- En espace utilisateur, vérifier les licences des composants → problème si l'un d'entre eux est GPL
- Le problème des licences peut être complexe
 - GPL (noyau Linux)
 - LGPL (Qt)
 - GPL avec exception (RTEMS)
- La GPL n'est pas forcément l'ennemie de l'industrie, elle garantit la pérennité de composants *essentiels* (Linux, RTEMS, GCC, ...)
- Un problème légal découle souvent d'une mauvaise conception

- Exécutif TR open source sous GPL avec exception
- Créé en 1984 pour l'armée américaine
- Maintenu par OAR (US)
- Communauté réduite mais active
- Déjà largement utilisé dans le spatial : EADS, ESA, CNES, NASA, ...
- API POSIX
- Support réseau
- Niveau de support variable suivant les architectures (excellent sur SPARC, mais version « spéciale » Gaisler)
- Les améliorations ne sont pas toujours reversées ☹
- Plus « basique » que Linux !

- Extension TR « dure » de Linux
- Simple à installer
- Mêmes API que Linux (POSIX)
- Nécessite des capacités matérielles (TSC)
- Garantit des temps de réponse entre 50 et 200 μ s sur x86 et ARM/Cortex
- Cible : systèmes industriels, bancs de test

- Extension TR « dure » de Linux
- API spécifiques mais POSIX, VxWorks, VRTX, ... → outil de migration d'application TR vers Linux
- Nécessite des capacités matérielles (TSC) pour obtenir les meilleurs résultats
- Garantit des temps de réponse autour de 10 μ s (x86)
- Pilotes spécifiques (RTDM) ☹
- Cible : systèmes TR embarqués à fortes contraintes

- On ne peut pas ne pas l'évoquer !
- Standardisation de l'IHM 😊
- Pas de support TR 😞
- Lourd à manipuler, des heures de compilation :
 - « The source download is approximately 8.5GB in size »
 - « If you are running Linux in a virtual machine, you will need at least 16GB of RAM/swap »
- Utilisable dans un contexte industriel après modification (stages EPITA)
- Un choix envisageable pour les terminaux / IHM

- Qt: Plate-forme de développement croisée C++
 - Pas uniquement une IHM → QThread, QSql
 - Acquis par Nokia en 2008 puis par Digia en 2012
 - Une référence dans le monde de l'embarqué
 - Un peu lourd mais peut fonctionner dans un environnement émulé par QEMU
 - Excellente documentation
 - Inclu dans Buildroot, OE
- EFL (Enlightenment) est plus léger mais moins de documentation
 - Très bonne performances
 - Portage aisé sur OS POSIX
- Utilisation industrielle des 2 environnements par OWI

- Le passage vers l'environnement libre peut rendre le choix complexe
- Les outils propriétaires sont ciblés, documentés
- D'excellents outils libres d'intégration, développement
 - OpenEmbedded, Buildroot, OpenWrt
 - Eclipse/CDT
 - QEMU
 - Git
- Les outils de mise au point ne sont pas toujours disponibles sur toutes les architectures (ex : Valgrind)
- Privilégier les outils *officiels* (mainline)
 - Ftrace / Kernelshark
 - Perf

- Très différente de la culture industrielle courante (surtout en France...)
 - Respecter les licences libres même si il n'y pas (ou peu) de gendarme
 - Participer à des projets open source pour alimenter le système (sur le temps de l'entreprise)
 - Accepter un état d'esprit raisonnablement geek: télé travail, horaires un peu décalés
 - Accès Internet « ouvert » (grands groupes)
- Certaines sociétés (US) ont bien intégré le modèle :
 - SFR → OpenWrt
 - Montavista & Wind River → OE, CDT, KGDB
 - Free (?)
- Le libre doit être intégré comme une culture et non uniquement comme un avantage financier

- Le choix du logiciel libre facilite la sous-traitance → imposé par des donneurs d'ordre (Orange)
 - OS et outils connus
 - Développement collaboratif par défaut
 - Pas de coût de licence
- Deux types de prestataires
 - Beaucoup d'indépendants
 - Très peu de SSII spécialisées
 - Privilégier ceux qui sont actifs dans la communauté libre et non les opportunistes
- Il est aisé de se présenter comme « expert Linux » après avoir suivi une formation 😞

- Attention au piège du « tout gratuit »
 - Free software = logiciel libre
 - Free software != logiciel gratuit
- La formation est un gain de temps important par rapport à l'énorme quantité d'information disponible
- Peu de prestataires en France (4 ou 5?) → attention à la qualité du contenu

- L'information industrielle avait été balayée des cursus supérieurs au profit des technologies web
- Réapparition de spécialisations :
 - EPITA / GISTRE
 - ENSEIRB, filière électronique
 - INSA
 - ...
- Les filières « électroniques » sont parfois mieux adaptées que les « informatique » qui font surtout du Java !

- www.openembedded.org
- <http://buildroot.uclibc.org>
- <http://www.xenomai.org>
- <http://www.rtems.com>
- <http://wiki.rt.kernel.org>
- https://kernel-recipes.org/?page_id=412
- <http://www.eukrea.com>
- <http://www.armadeus.com>
- <http://qt.digia.com>
- <http://trac.enlightenment.org/e/browser/trunk>
- <http://source.android.com/source/downloading.html>

Questions ?