

Embedded Linux

-- Conférence Paris 8 --

Gilles BLANC, gblanc@linagora.com

Licence

Ces slides utilisent des extraits de documents considérés comme citations courtes. Pour l'ensemble du document, à l'exception des parties précédemment évoquées ainsi que les images et photographies d'illustration, la GNU FDL est appliquée (copies, modifications, traductions autorisées sous réserve de citation de l'auteur)

Copyright (c) 2008 Gilles BLANC.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found on:
<http://www.gnu.org/licenses/fdl.html>

/me

- ingénieur informatique embarquée/temps-réel chez **Linagora**
- EPITA GISTR 2006, 100% sous Linux depuis 2004, libriste et membre actif de Parinux et de l'APRIL
- expériences :
 - Thales Com' : portage d'un simulateur de *poste radio* militaire (HP-UX vers **Linux**), Ada/C
 - Trango : *paravirtualisation* (sous **Linux**)
 - Alcatel : simulateur pour répéteur *DVB-SH* sous **Linux**
 - SAGEM Com : intégration d'un **Linux** pour *module GPRS*
 - Philips : démos pour service *Set Top Box*, sous **Linux**
 - Aldebaran : refonte du bus de communication du *robot Nao* (**Linux**)
 - GCB : **OpenBSD** sur plate-forme *serveur réseau* avec problématique de l'embarqué
 - OKL4 : maître de stage pour un *hyperviseur de paravirtualisation libre* (faisant tourner... **Linux**)
 - CEA : contrôle-commande par **logiciels libres** (Tango/Python/QT/Linux) pour banc de test

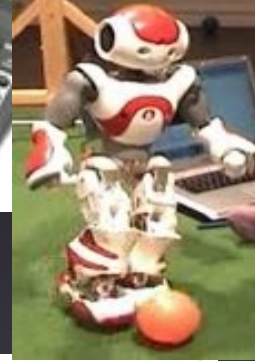
(Groupe) Linagora

- **Société de Services en Logiciels Libres**
 - Pôles Java, Sécurité, Studio, etc : R&D au forfait
 - Intégration logicielle (F/OSS)
 - Support assurance logicielle (OSSA)
 - Formations
 - Hosting (NetAktiv)
- **Société d'édition logicielle**
 - Messagerie : OBM (AliaSource)
 - Solutions : LinPKI, LinSM, etc
 - Tosca
 - ...
- **Un pôle embarqué**

(Groupe) Linagora

- Paris, Toulouse, Lyon, Marseille, Bruxelles, Londres, San Francisco
- 130 personnes (une soixantaine sur Paris)
- Forte croissance continue
- Business model sur l'OpenSource
 - Pas de licences à payer
 - Intégration (nbre de postes à prendre en compte)
 - ou développement sur demande
 - Assurance OpenSource
 - Suivi/formations

Introduction



Avez-vous ?...

- une {free, live, darty, alice, c, neuf}box ?
- un lecteur dvd/divx ?
- un disque dur multimédia ?
- un assistant personnel/lecteur multimédia portable avec wifi ?
- un routeur wifi avec administration html ?
- du CPL ? un téléphone wifi ?
- bien d'autres choses.....

Eh bien alors...
Vous êtes envahis de **Linux** !

(c'est fou, non ?)

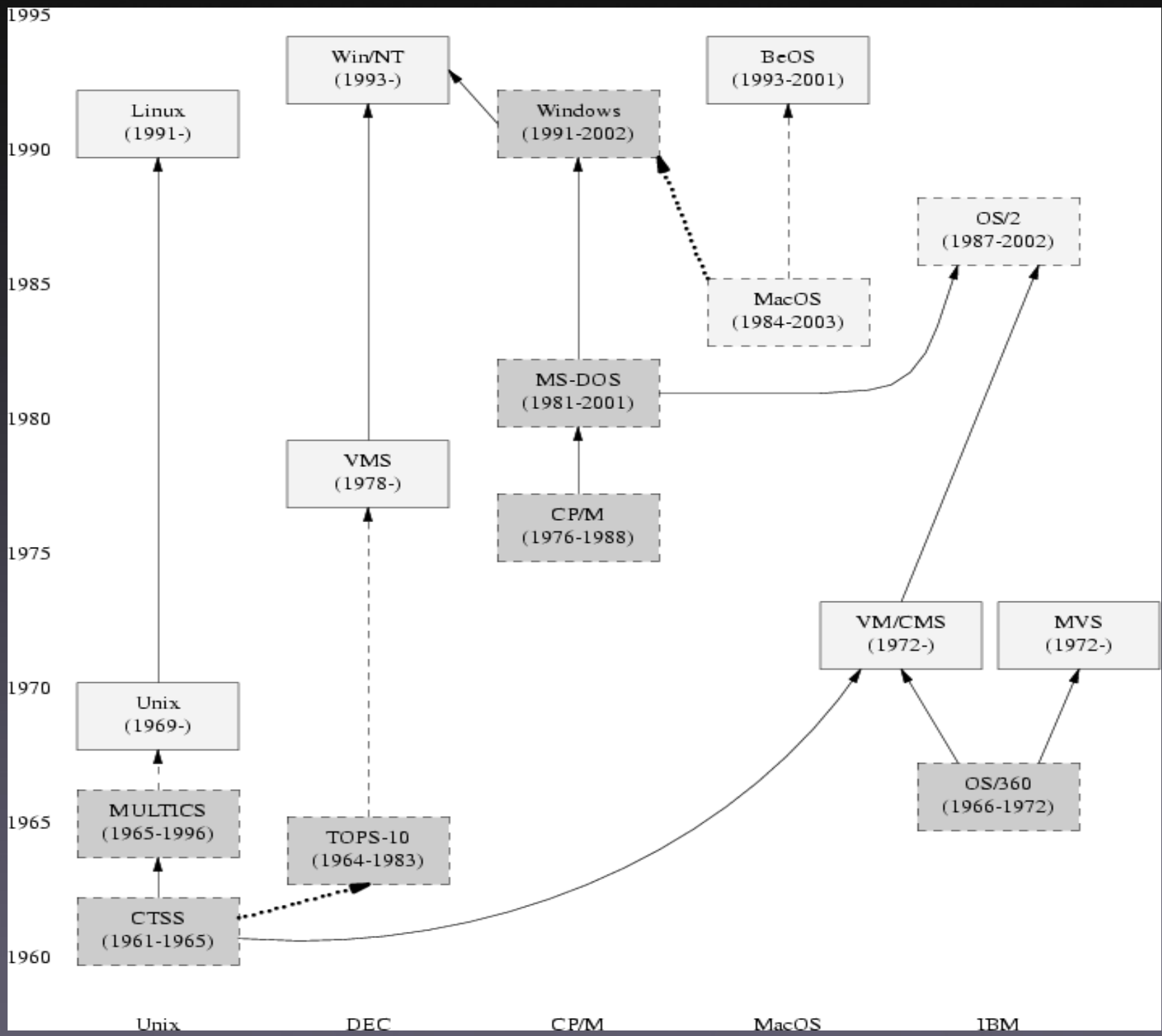
(et même les windowsiens n'y couperont pas !)

Qu'est-ce que Linux ?

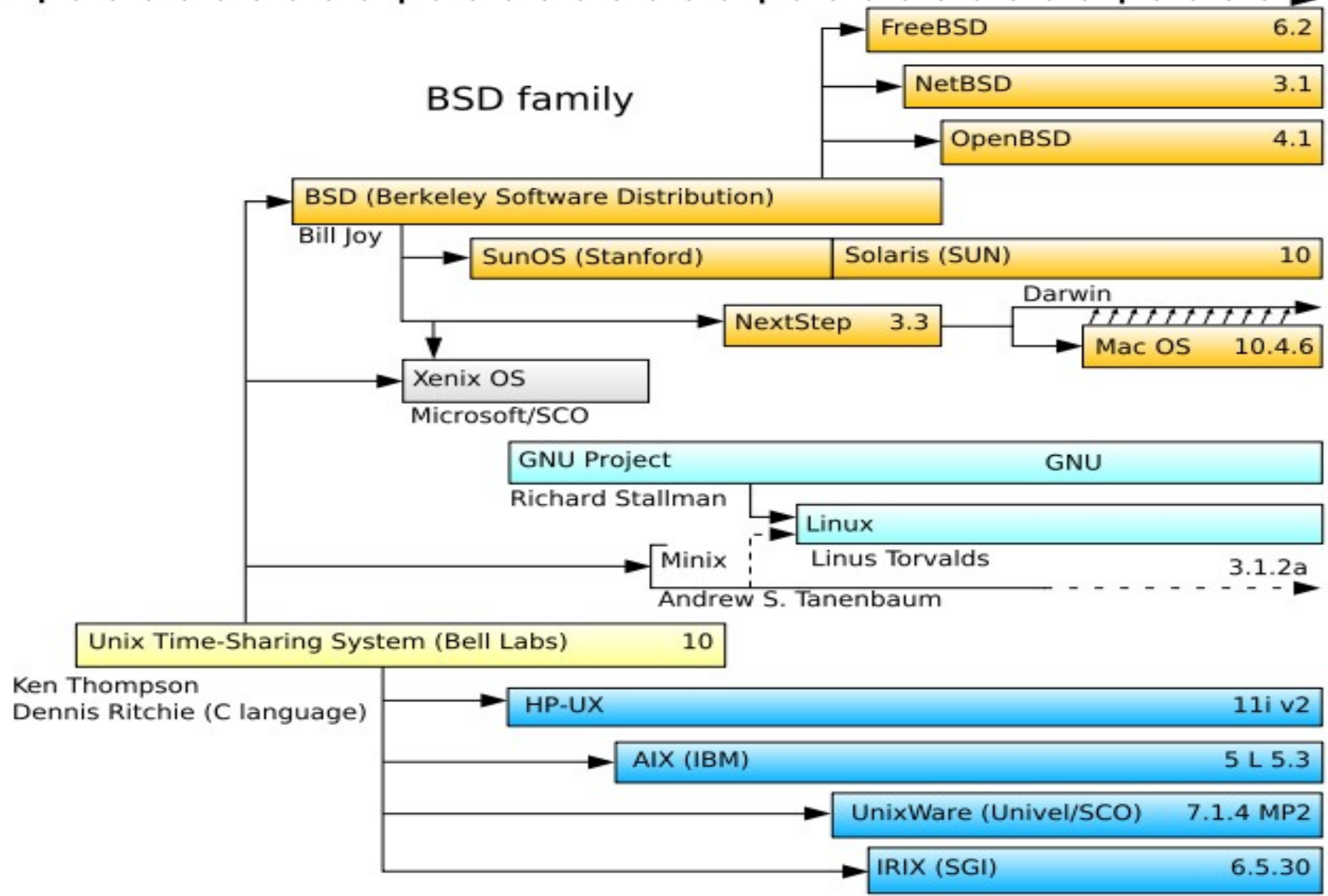
architecture software

- Qu'est-ce qu'un OS ?
 - un noyau (kernel)
 - interaction avec le matériel (interruptions, mémoire)
 - gestion basiques, « appels systèmes »
 - sécurité
 - gestion de « tâches »
 - des programmes/logiciels
 - un tas de configuration
 - des outils d'installation, administration, etc

- Il existe des milliers (à la louche) d'OS (ou au moins de kernels)
- De grandes familles se distinguent parfois, mais énormément de propriétaires obscurs oubliés
- Même wikipedia est très incomplet
http://en.wikipedia.org/wiki/List_of_operating_systems
- Le petit nouveau pas si nouveau : Linux
 - ou plutôt GNU/Linux
 - l'héritage *NIX (et le « tout fichier »)
 - kernel « monolithique » mais modules/drivers



1970 1980 1990 2000 Time



System III & V family

Historique

- « Linux is evolution, not intelligent design », Linus Torvalds
- 1991: première annonce sur usenet (Freax)
- 1992: v0.96 GPLed pleinement fonctionnelle
- 1996: v2.0
- 1999: entrée en bourse de RedHat
- 2001: v2.4
- 2003: v2.6

http://en.wikipedia.org/wiki/History_of_Linux

GNU/Linux, les distributions



- organisation générale d'un système GNU/Linux
- qu'est-ce qu'une distribution ?
- les différentes distribs (<http://distrowatch.com>)
 - Slackware
 - Debian (-like : Ubuntu, Mepis, etc)
 - Red Hat (-like : Fedora, Mandriva, etc)
 - SuSE : [open]SUSE
 - les autres : Gentoo (BSD-like), LFS, etc
 - les lives : Knoppix (CD/DVD), Damn Small (clef usb)
- les différences et similitudes, les philosophies, et les influences dans l'embarqué/temps réel

Qu'est-ce que l'embarqué ?

(sait-on jamais...)

« L'avenir est dans l'embarqué »

- L'informatique embarquée est partout :
 - téléphones mobiles, assistants personnels
 - modem, routeurs, STB, lecteurs DVD/divx
 - mais aussi :
 - machines à laver
 - satellites, fusées, avions, voitures
 - missiles, radio militaire, radars
 - etc
 - nouveaux marchés : lecteurs multimédia portables, ...
- marchés gigantesques, ignorés du grand public

Les contraintes propres à l'embarqué

- contrainte de **taille mémoire**
- contrainte de **temps de réponse**
- contrainte de **fiabilité**, voire de **sécurité**
- contrainte de ressource d'**énergie/autonomie**
- contrainte d'**architecture matérielle**
- contrainte de **prix de développement** (pur et licences)
- contrainte de **prix de vente** (amortissement et **royalties**)
- contraintes... **juridiques** !

Le pendant Temps Réel (RT)

- tout système embarqué ne doit pas forcément être temps réel, et vice-versa... mais quand même
- temps réel « mou » et temps réel « dur »
- ce qui existe à l'heure actuelle
 - VxWorks, LynxOS, QNX, threadX, RTX, Nucleus, VRTX, WinCE, OS20/OS21, RTKe, et des centaines d'autres !!
 - point commun : licences abominables, royalties, performances et capacités en contradiction
 - il faut que ça cesse ! La réponse : **Linux**...

Qu'est-ce que Linux
embarqué ?

GNU/Linux dans l'embarqué et le temps réel

- Un peu d'histoire (encore) :
 - 84/85 : lancement du projet GNU par RMS, puis FSF
 - fin 90's : projet d'OS finalisé (gcc, binutils, glibc, emacs), à l'exception du noyau
 - juin 91 : GPLv2
 - **1991** : Linux par Linus Torvalds
 - **2001/2002** : premiers projets de GNU/Linux dans l'embarqué (télécom)
 - **2007** : sortie de la GPLv3 (not. en réponse à l'embarqué : « TiVoization »)
- Étendu du libre : centaine de milliers de projets ; en 2004, ~70% sous GPL

- Pas seulement Linux : « libre embarqué »
 - outils portables ou adaptables entre OS :
 - gcc, gnat : compilateurs
 - busybox : utilitaires
 - eclipse, gps : IDE de développement
 - gdb : debugging
 - OpenOCD : J-TAG
 - u-boot : bootloaders
 - et même... freeBIOS !
 - exemple d'utilisateurs propriétaires :
 - WindRiver : VxWorks (Workbench)
 - LynuxWorks : LynxOS (compatibilité binaire !)
- Au-delà de Linux : eCos, RTEMS, OKL4
- Au-delà du projet GNU : NetBSD (OpenBSD)

- Les formats ouverts

- « il voit des formats partout ! » (® www.formats-ouverts.org)

- échanges de données
- stockage de données
- systèmes softwares
- systèmes hardwares

- les avantages

- pérennité
- interopérabilité
- coût !

- exemples :

- communication : XMPP
- audio/vidéo : ogg Vorbis/Theora (décodeur Tremor)
- images : png

Pourquoi le libre ?

- nombreux avantages pour l'industriel :
 - technicité (à la Torvalds)
 - coût (à l'entrepreneuriat) : mais libre n'est pas gratuit ! (heureusement : c'est notre boulot)
 - pas de royalties (pbm majeur en embarqué)
 - pérennité/éthique (à la RMS)
- licences : GPL, LGPL, BSD, Apache, etc
- suprématie de la GNU GPL : avantages et conséquences (copyright/copyleft)

Le noyau Linux

- Pourquoi Linux ?
 - libre : GPLv2, forte évolutivité
 - pas de royalties
 - sources gratuites
 - nombreux points forts techniques
- cible/marché
 - du très embarqué...
 - ... au Consumer Electronic
- temps réel : « mou » ou « dur » ? Linux peut être adapté pour répondre aux deux !



Les points forts de Linux

- faible empreinte mémoire p/r aux capacités (not. pour les drivers, exigence pour toute intégration)
- efficace, optimisé, fortes performances ; uptime
- userland efficace (et support Java)
- modulaire, paramétrable très finement
- énorme support matériel, portabilité
- $\Theta(1)$ $O(\log n)$, préemptible (mais le 2.4 ne l'était pas !)
- beaucoup d'outils disponibles, de bibliothèques
- développement userfriendly (hôte ~ cible)
- TCP/IP (serveurs !), bluetooth, multimédia,... : de base

Get the facts

Dans l'embarqué

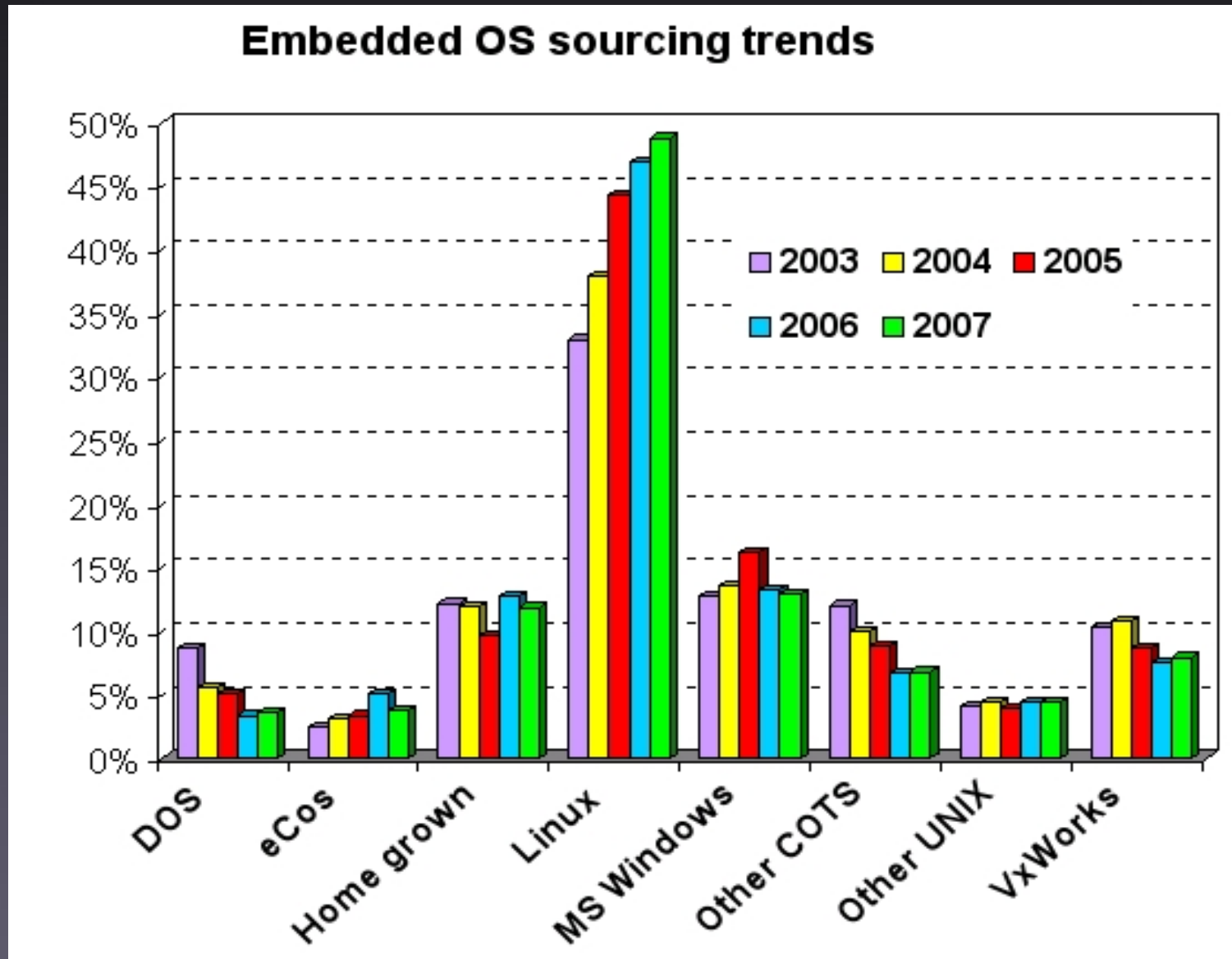


- peu de données 100% fiables
 - culture du secret industriel
 - parfois militaire
 - projets en cours, non finalisés
- étude à suivre : « snapshot of the embedded Linux market » (d'autres études confirment)

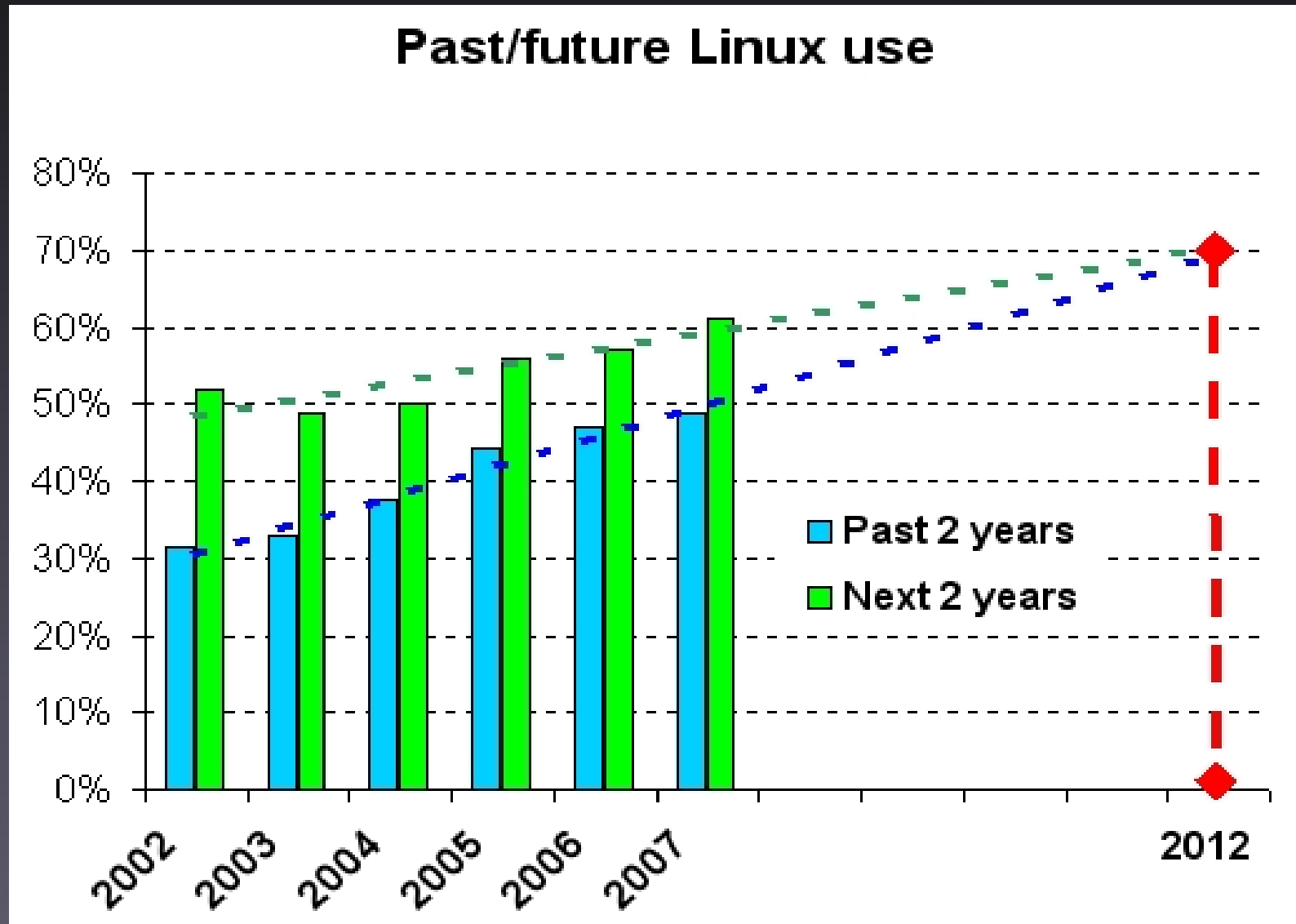
<http://www.linuxdevices.com/articles/AT7065740528.html>

- les salons confirment un intérêt croissant et soutenu (il paraîtrait même qu'on en donne des conférences), Linux fait à présent partie du paysage

Which OSes have been in your (company's) embedded designs during the past two years?



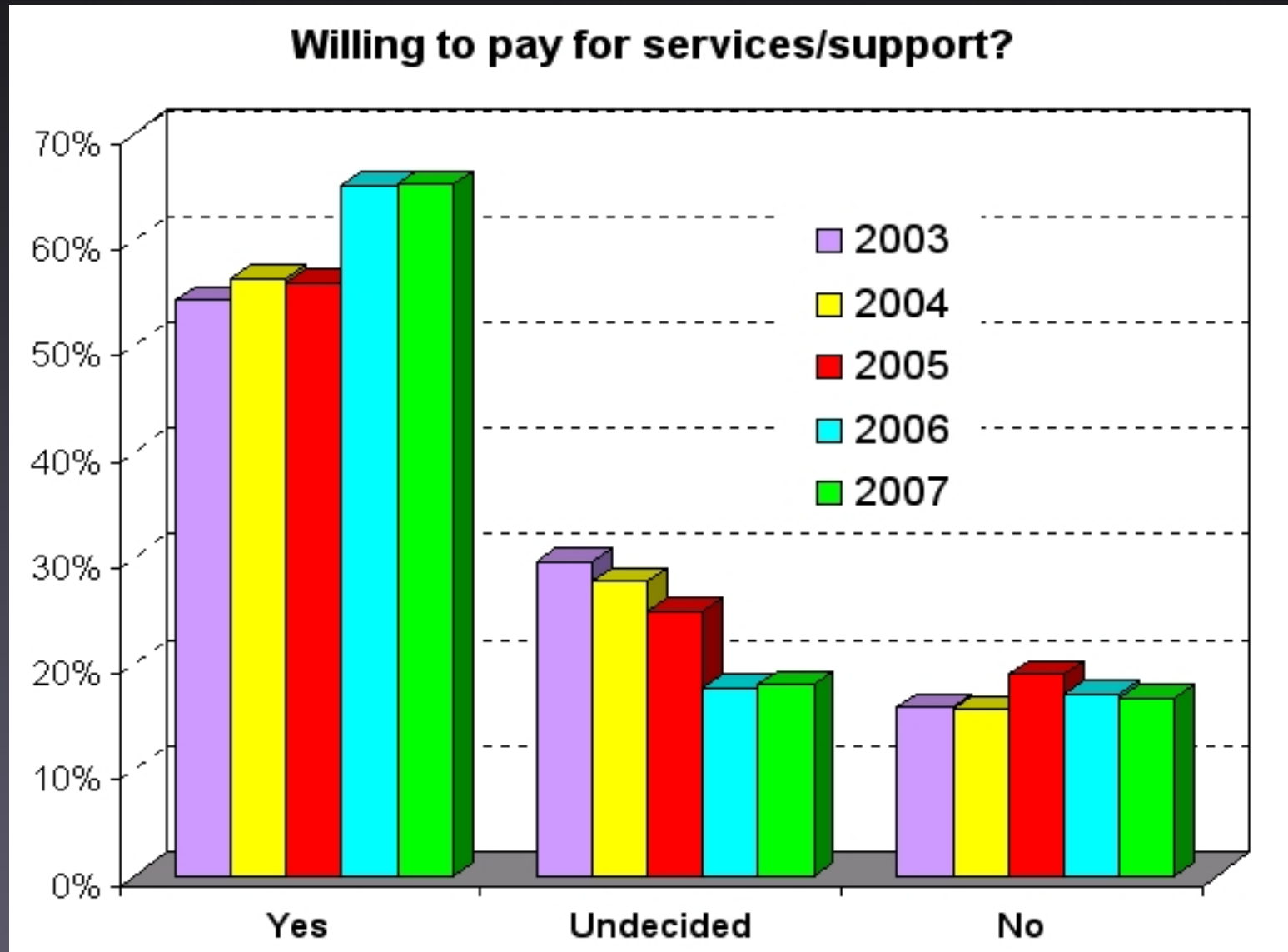
Actual and planned Linux use may converge by 2012



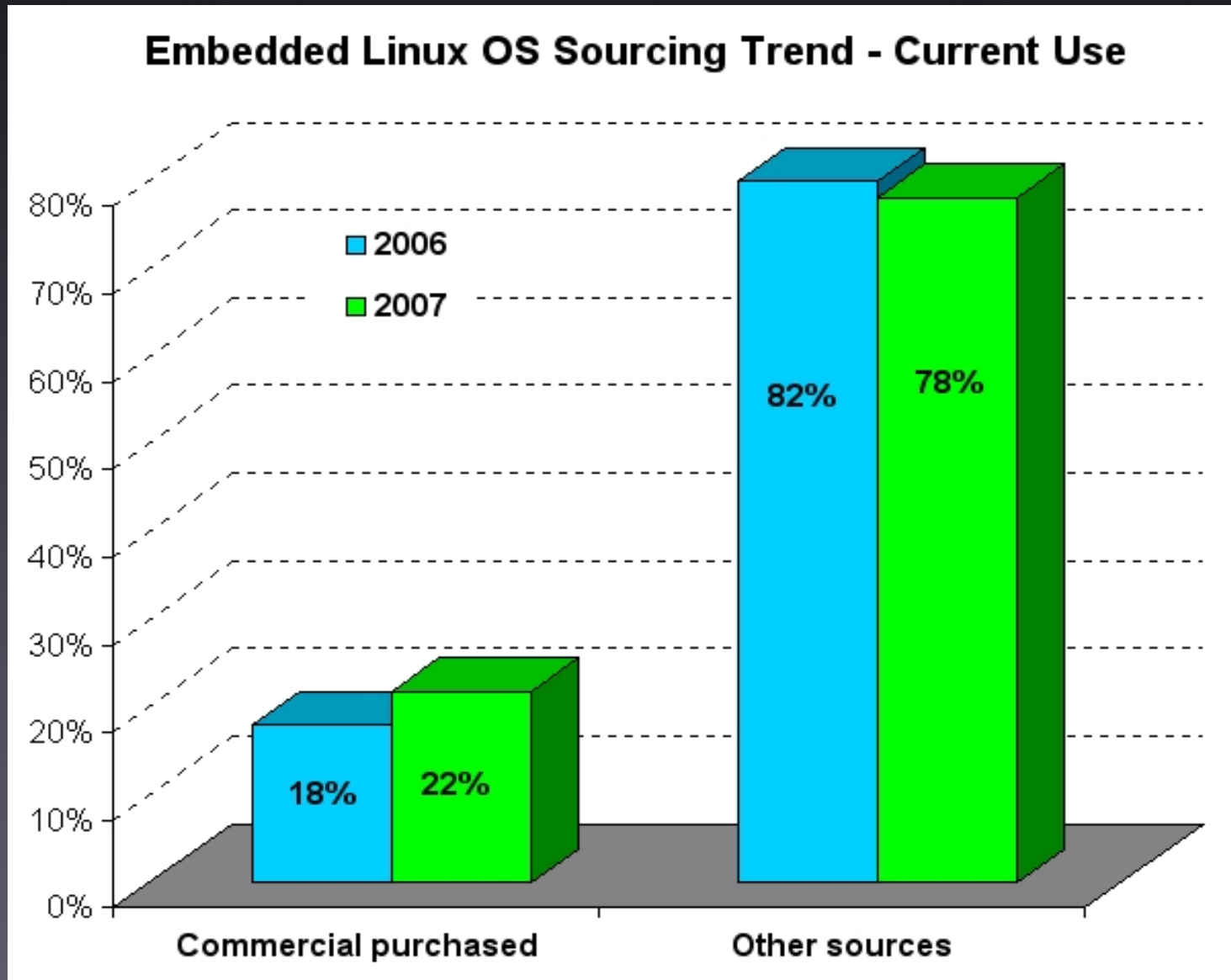
Le choix

- doit répondre à la problématique
- commercial vs gratuit / communautaire
- temps réel mou, patch de préemption, ou temps réel dur ?
- l'offre peut sembler contradictoire
 - difficile pour le n00bs de s'y retrouver
 - et pourtant... l'embarras du choix
- l'avantage et l'écueil du packaging (du boulot pour ingénieur en perspective)
- distribution actuelle et tendance...

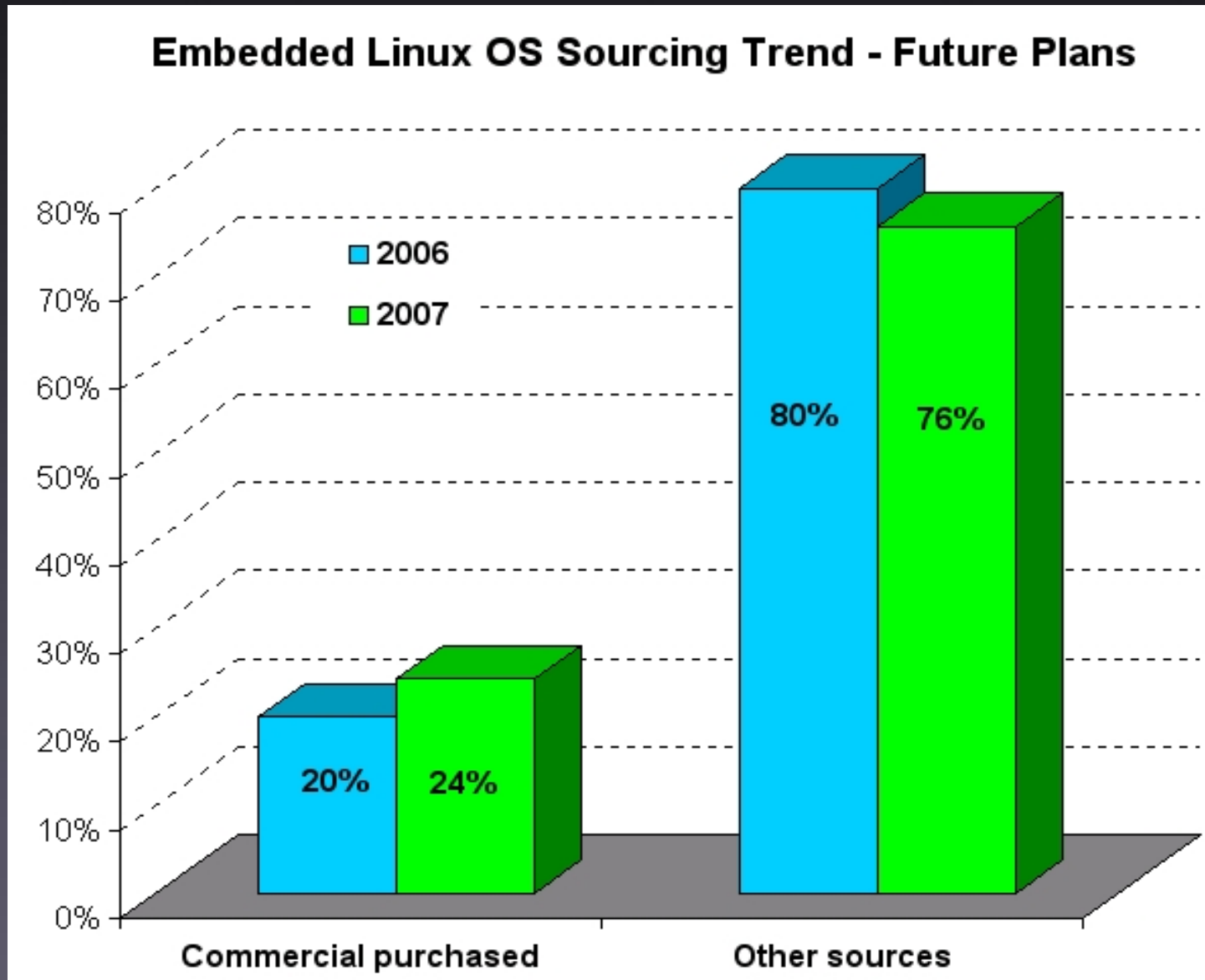
Would you consider paying for support or services?



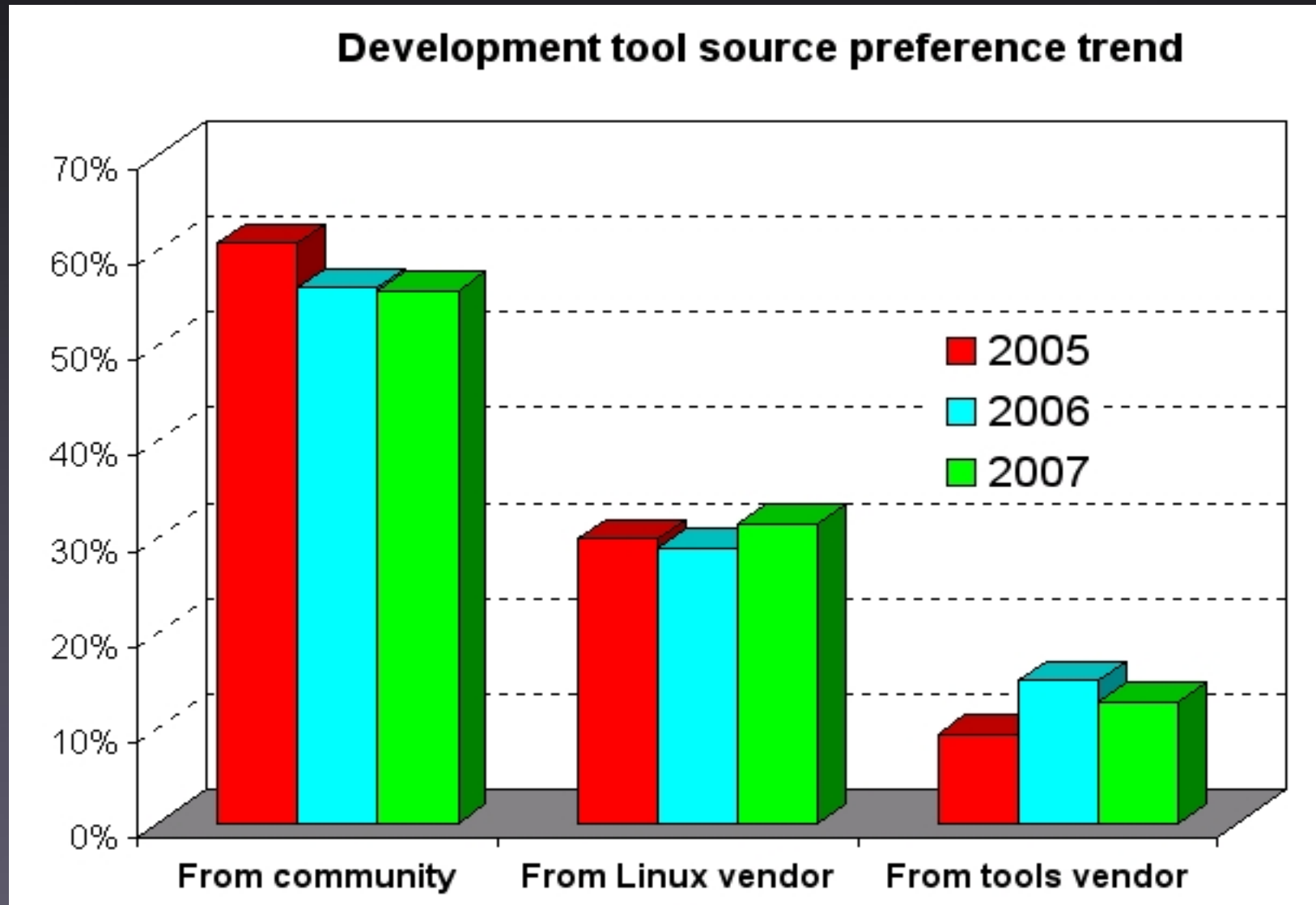
How did you obtain your (company's) embedded Linux software?



How do you expect to obtain your (company's) embedded Linux software during the next 2 years?



What development tools do you expect to use?



L'existant

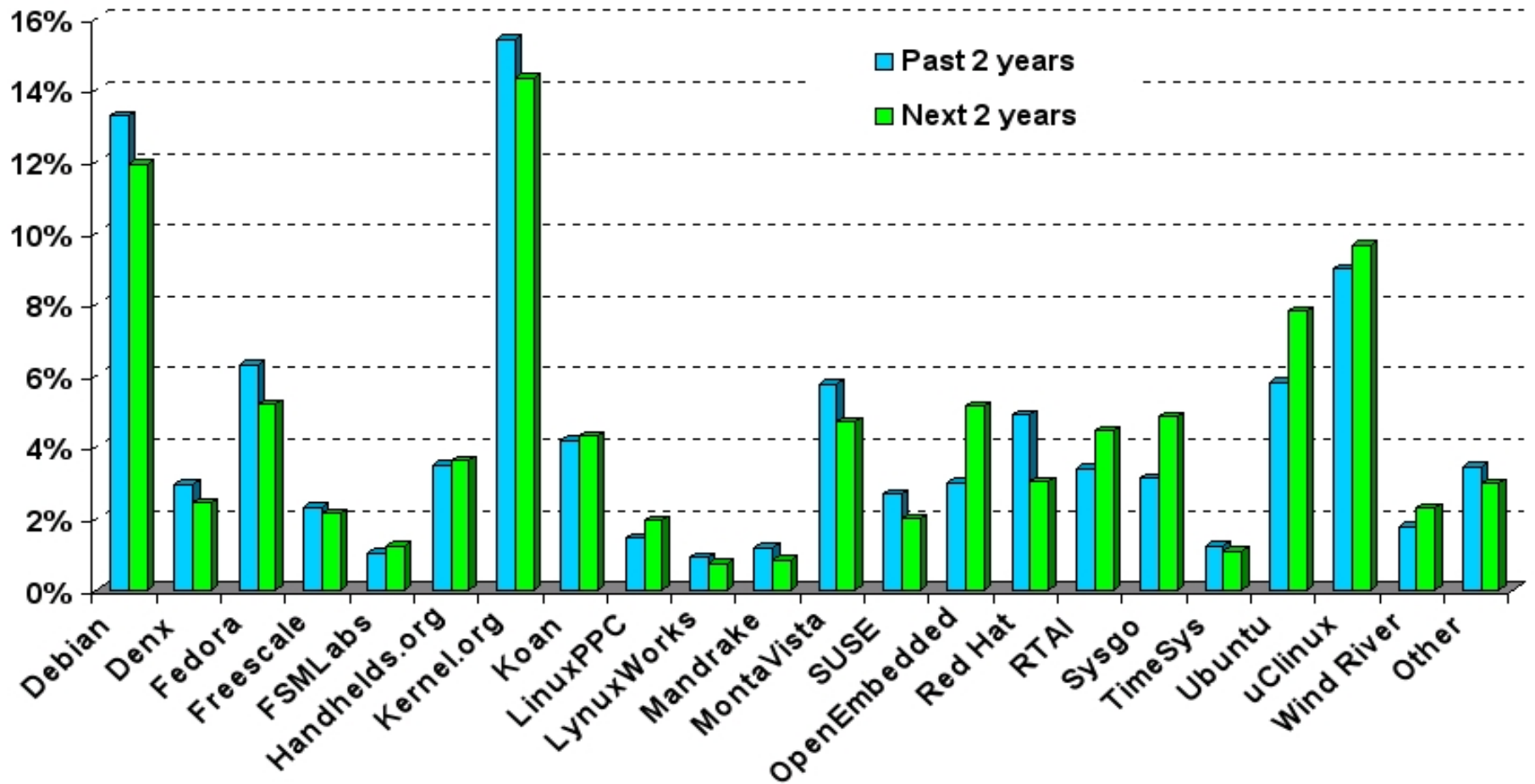
- « out of the box »
 - Debian, RedHat/Fedora
- ready to embed (le « prêt-à-embarqué »)
 - OpenEmbedded, WindRiver, Sysgo, LynuxWorks, etc
- Real Time
 - MontaVista, Concurrent Computing, etc
- Environnements adaptés/à adapter
 - ex: Maemo (Hildon), openMoko, Sugar, Moblin, Android

La solution « facile »

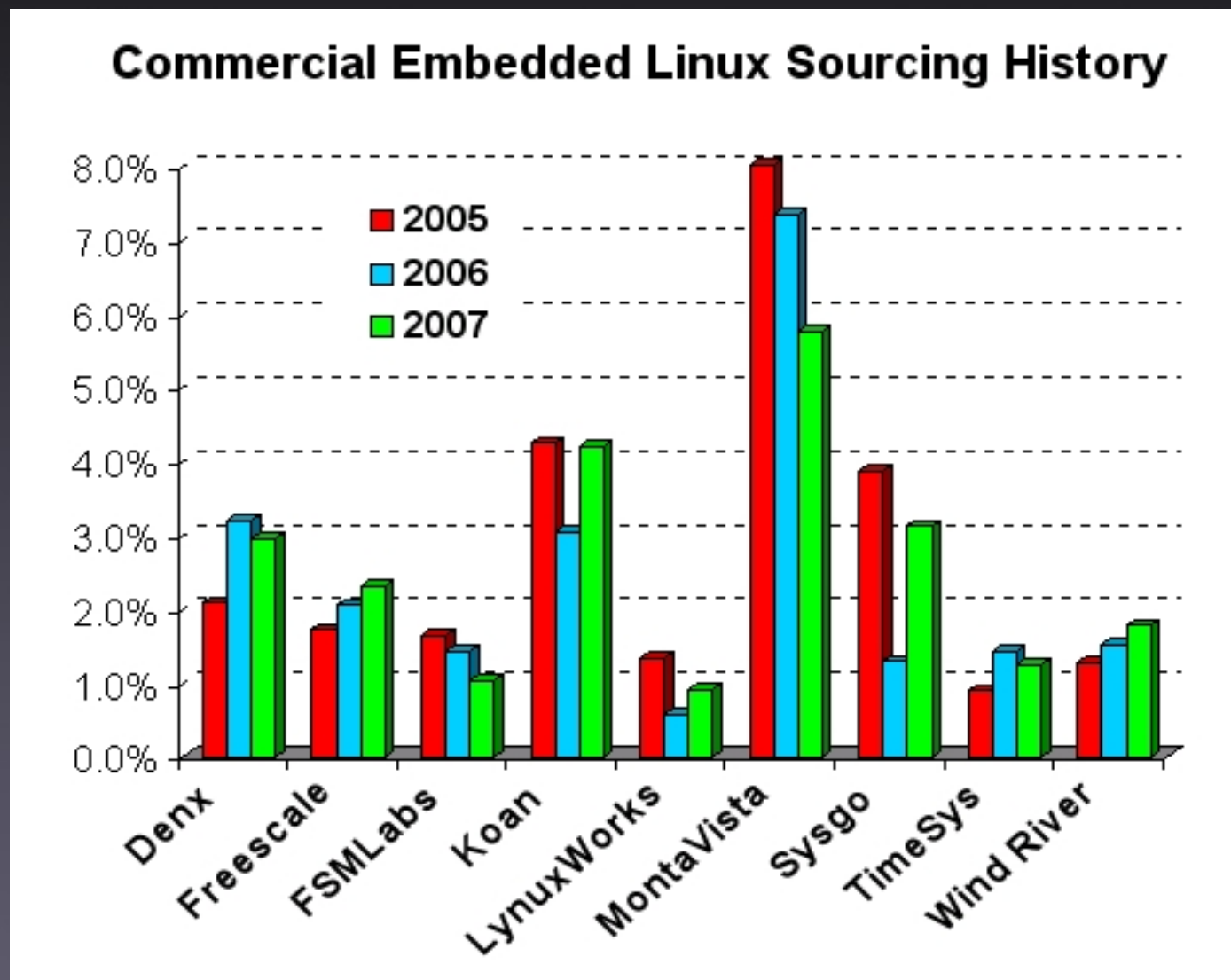
- exemple : sur carte VME compatible x86/PPC/PowerQUICC (Gaci, Ecrin, Thales, ...)
- distribution standard, « desktop »
- « temps réel mou », noyau préemptif 2.6
- kernel « vanilla »
- Fedora impaire compatible RedHat Enterprise (raisons historiques)
- ou Debian : 11 architectures stables

What sources of Linux were/will be in your (company's) embedded designs during the past/next two years?

Embedded Linux OS sourcing trends



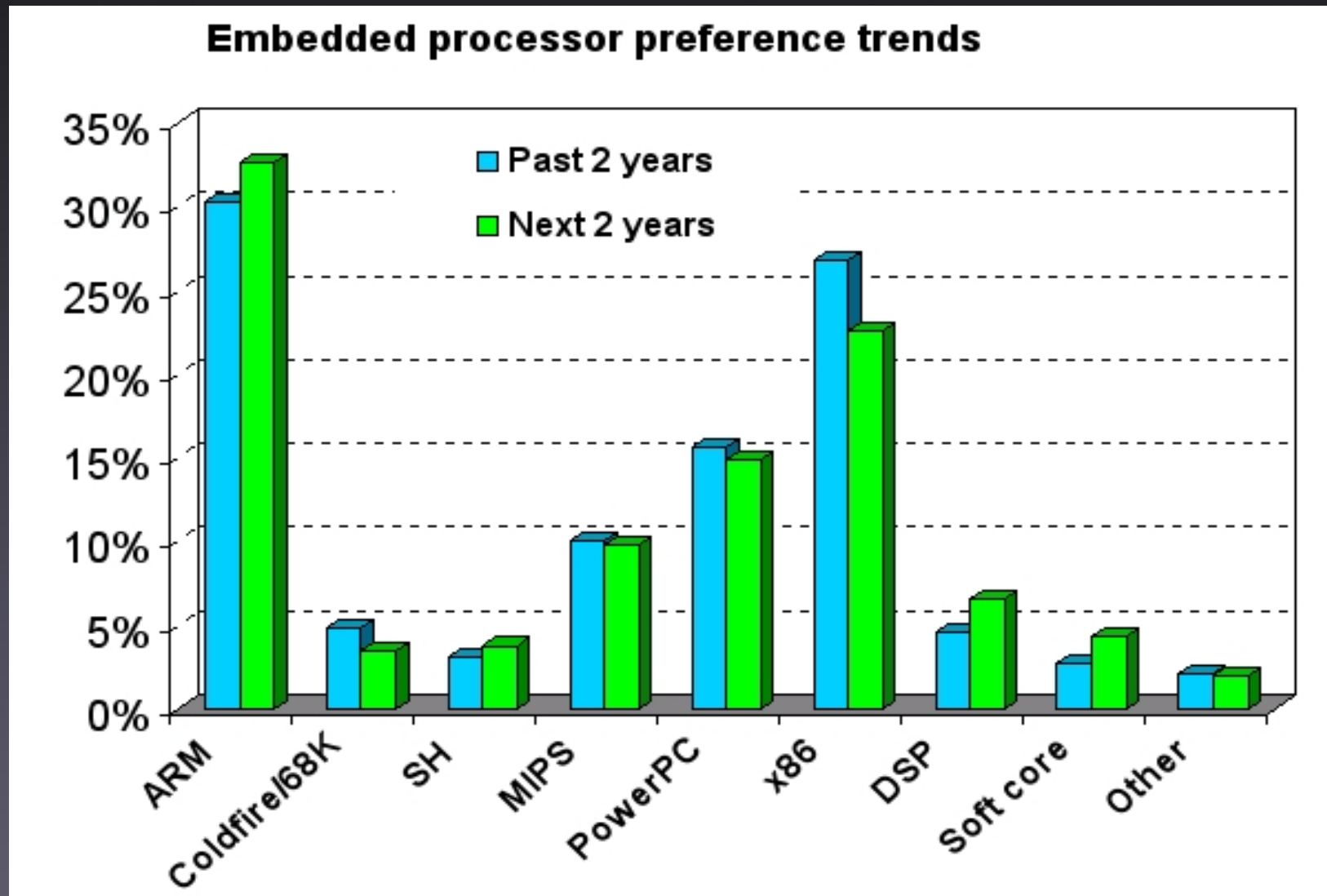
What sources of Linux were in your (company's) embedded designs during the past two years? -- historical trend



Les cibles

- ~75 architectures CPU gérées
 - 32bits minimum
 - MMU requise (sauf pour μ Clinux)
- les principales en TR/embarqué :
 - ARM/Xscale: communication (portables) (TI OMAP), assistants personnels (ARM9 / Cortex-A8)
 - MIPS: routeurs/modems (*BCM6348*)
 - SH4: Set Top Boxes, platines divx (*ST7100*)
 - PPC: réseaux soutenus (*PowerQUICC II*)
 - x86: le reste du monde (*PC104, AMD Geode*)

Which OSes were/will be in your (company's) embedded designs during the past/next two years?



Linux (really) everywhere

- pour lancer des missiles :

<http://www.linuxdevices.com/news/NS4667725014.html>

- dans les télévisions Sony :

<http://www.sony.net/Products/Linux/>

http://products.sel.sony.com/opensource/source_tv.shtml

- des robots maison ou pro :

<http://www.linuxrobots.org/>

<http://www.aldebaran-robotics.com/>

- pour traire les vaches :

<http://www.linuxdevices.com/news/NS4275702675.html>

- pour faire des crèmes glacées :

<http://www.generation-nt.com/linux-open-source-actualite-13082.html>

Quick overview

Les différents projets

- Les projets pour l'embarqué
- Les projets pour le temps réel
- Les projets libres et communautaires
- Les projets commerciaux
- Les projets libres lancés par les commerciaux
- Les commerciaux qui participent aux projets libres

Projets communautaires

- OpenEmbedded
 - distribution « source » pour l'embarqué
 - customisable
 - très répandue
- RTAI / Xenomai
 - temps réel dur
 - beaucoup de fonctionnalités
- Busybox, uClibc, u-boot, OpenOCD, etc
 - projets communautaires de softwares et bibliothèques optimisés

Projets commerciaux

« clefs en main »

- deux types : les embarqués, et les Temps Réel (quasiment toujours embarquables)
- Sysgo/ElinOS, LynuxWorks/BlueCat, ST Microelectronics/STLinux 2.0, WindRiver/Real Time Core (ex-RTLinux) : les grands acteurs débarquent
- TimeSys/LinuxLink, Koan Software/KaeilOS
- Concurrent Computing : 30 μ s garanties sur RedHawk
- la référence reste MontaVista

Les environnements pré-intégrés

- Qt embedded (ex-Qtopia)
 - trolltech/Nokia, licence LGPL
 - efficace, simple à mettre en oeuvre
- Gnome Mobile *<http://www.gnome.org/mobile/>*
 - Maemo (N770, N800, N810)
 - Debian/scratchbox puis openEmbedded
 - interface basé sur Hildon, GTK+ allégé et adapté
 - openMoko (téléphone Neo1973/Freerunner)
 - Sugar (OLPC)
- Android (Google, licence permissive Apache)
- OpenedHand/PokyLinux, Moblin (Linux Foundation), etc

Les intégrateurs

- Linux ne s'intègre pas tout seul (loin de là)
- Même si c'est plus simple a priori (à nuancer)
- Les SSI répondent au cas par cas, souvent en prestation : pas de garantie de résultat
- L'évolution tend vers le forfait pour les nouveaux projets
- **Linagora** est là pour ça ! ;) (et la concurrence en France n'est pas bien organisée/nombreuse, à vrai dire)

Conclusion

Avantages

- matière première gratuite, abondante
- code de qualité, ouvert
- plate-forme de dev idoine (tester sur place)
- support matériel (si si ! Du jamais vu, même)
- architecture sympathique
 - userland, modules, ça change du kerneland
- aide et documentation (même si...) : communauté (IRC, forums, pages man et info, listes de diffusion)
- ouverture (parfois meilleure que les NDA !)

Inconvénients

- coûts cachés ou « le libre n'est pas gratuit »
- très bons développeurs/intégrateurs nécessaires (rare et cher ; en théorie ; mais tant mieux pour nous)
- besoin de temps (notamment une partie incompressible de compilation)
- être root et avoir une connexion internet
- bien dimensionner, ne pas s'embarquer dans n'importe quoi
 - un noyau 2.4 vanilla pour du TR
 - de la mémoire largement insuffisante (4Mo mini !)

Le travail de l'ingénieur

- des légos
(les paquets, la construction)
- de la super-glue
(les scripts, les programmes persos)
- du scotch
(les trucs et astuces, les rustines)
- des incantations mystiques
(c'est-à-dire le savoir-faire)
- beaucoup de compétences requises, de la patience, des talents en haute cuisine

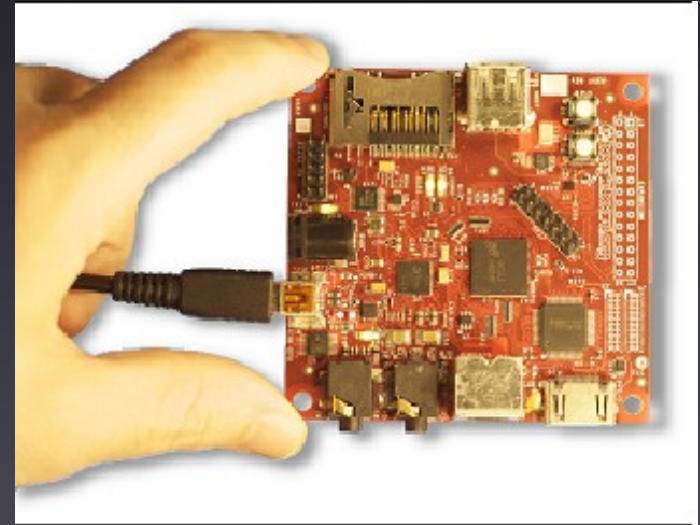
Futur proche et lointain

- amélioration des performances (préemptibilité, scheduling, support matériel, etc)
- (para)virtualisation
- développement durable (le retour de la domotique), certification
- drivers userspace / API générique ?!
- toutes les STB ? (m\$ bouge encore !)
- les téléphones !
- à la conquête du monde

Aller plus loin
(par soi-même)

La beagle board

- Carte de dev par TI
- Très ouverte
- <http://beagleboard.org/>
- Attention à la connectique



Only **\$149**

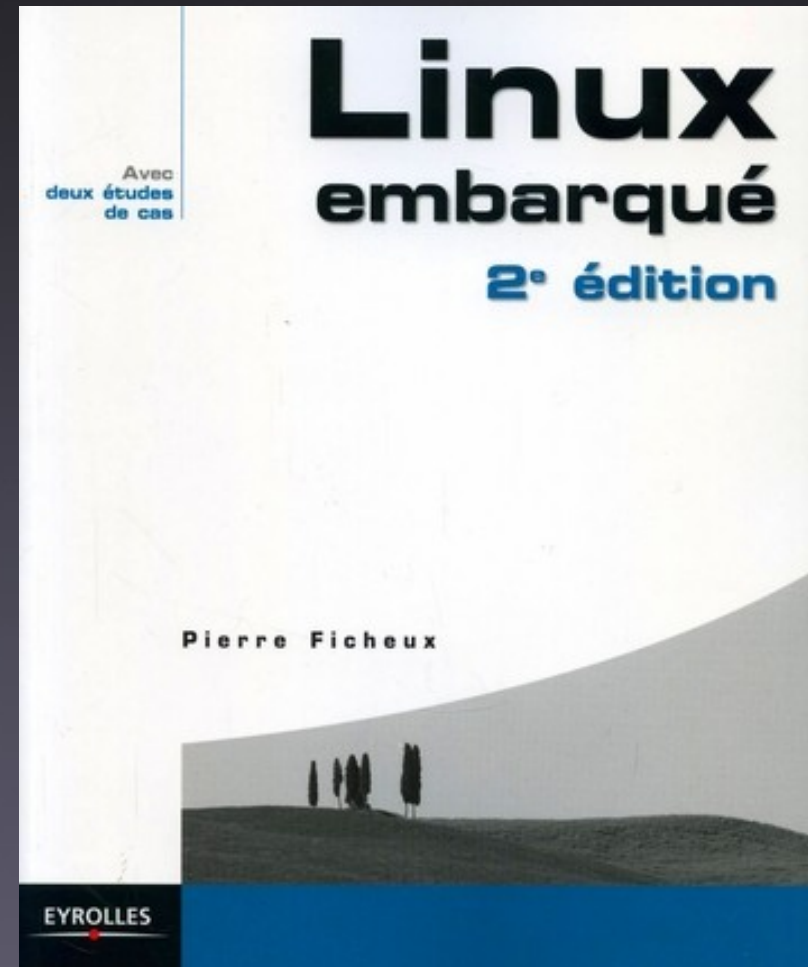
USB-powered **Beagle Board** delivers laptop-like performance and expansion

- Based on **OMAP3530 processor** featuring ARM® Cortex™-A8
- High-speed USB 2.0 OTG port for use as power source or expansion
- Flexible expansion through USB and standard PC peripherals
- Access to active and growing community development

Available now from:  

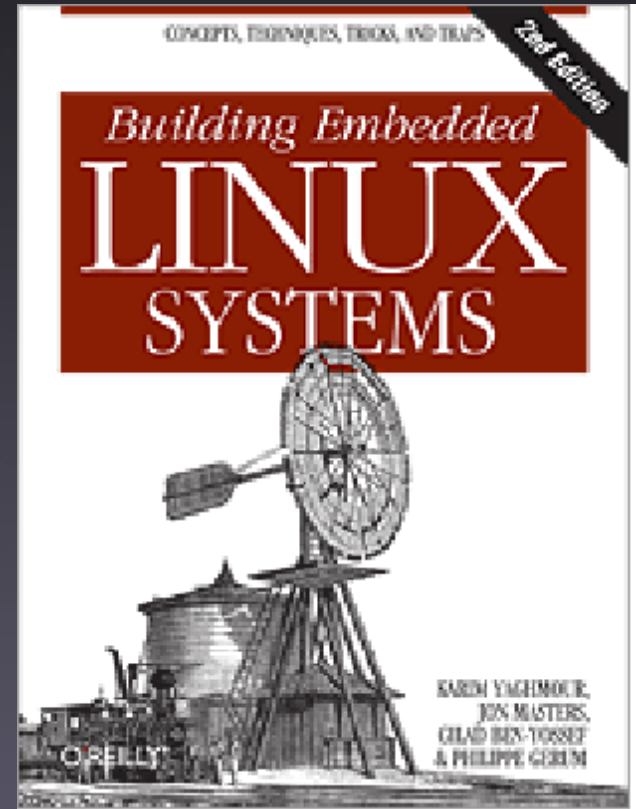
Linux Embarqué

- Pierre Ficheux
- édition Eyrolles
- 330 pages
- l'incontournable... en Français
- très bon exercice de vulgarisation, simple à lire



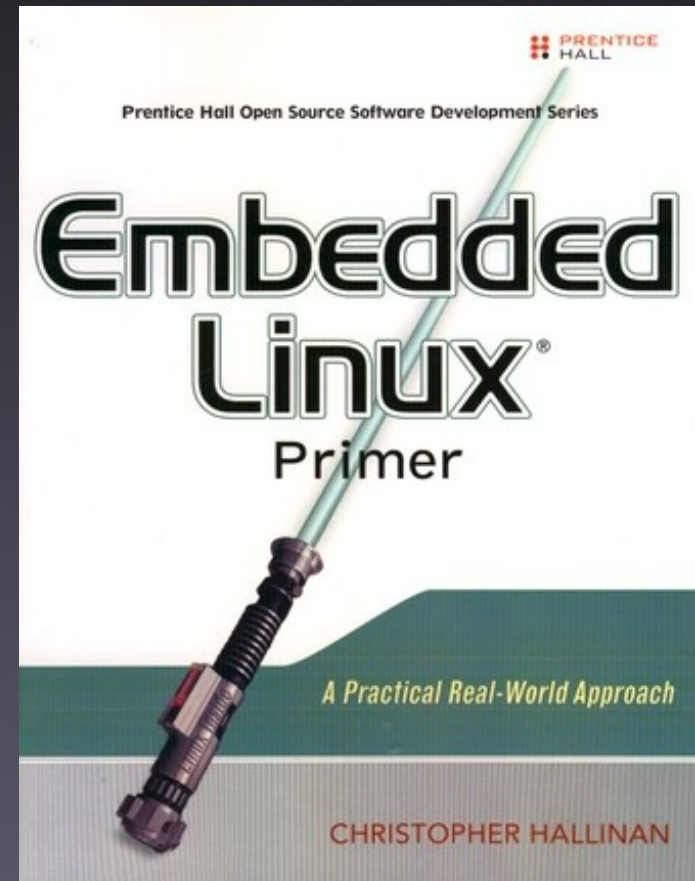
Building Embedded Linux Systems

- K.Yaghmour, Jon Masters, Gilad Ben-Yossef, Philippe Gerum
- édition O'Reilly
- 464 pages
- + technique
+ austère
++ profond
+++ complet
+++ mieux pour le travail d'ingénieur
- 2008, nouvelle édition



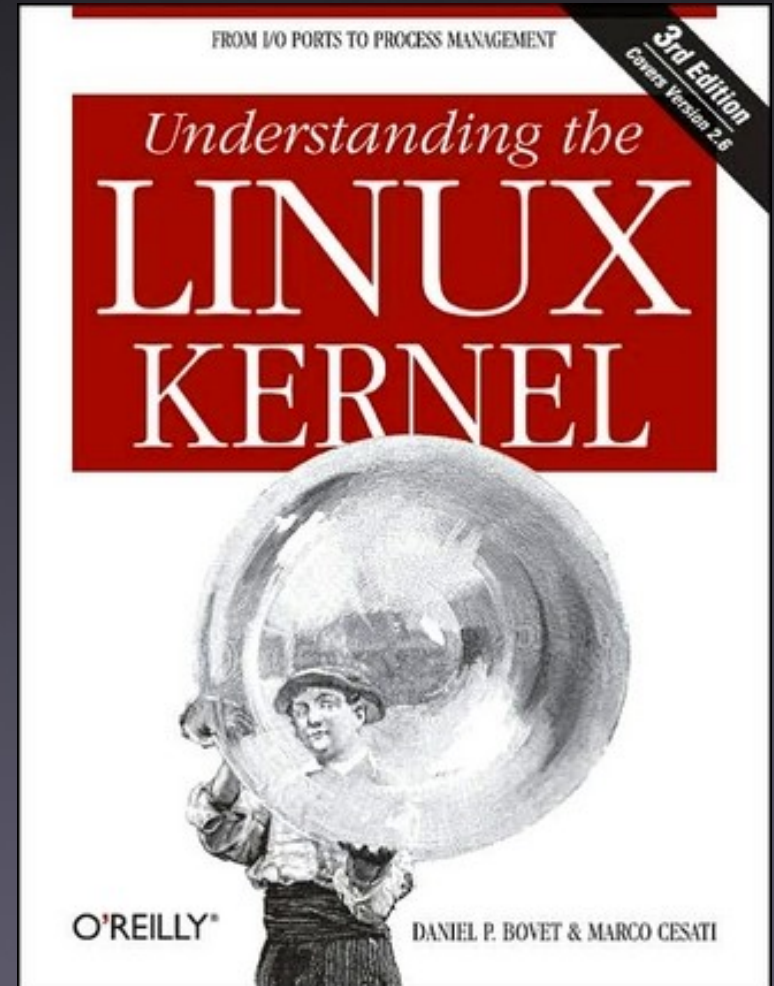
Embedded Linux Primer

- C. Hallinan
- édition Prentice Hall
- 540 pages
- apparemment rudement bien et très complet (non testé)



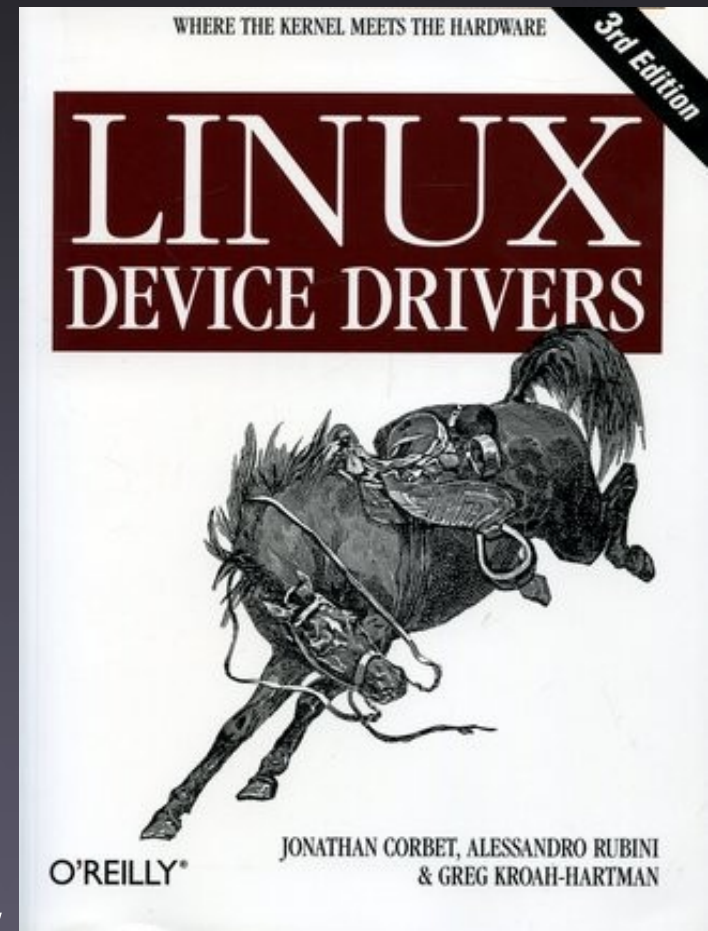
Linux Kernel

- Daniel P. Bovet,
Marco Cesati
- O'Reilly
- 920 pages
- la référence absolue
et l'un des seuls rares
ouvrages existants
(mises à jour constantes)



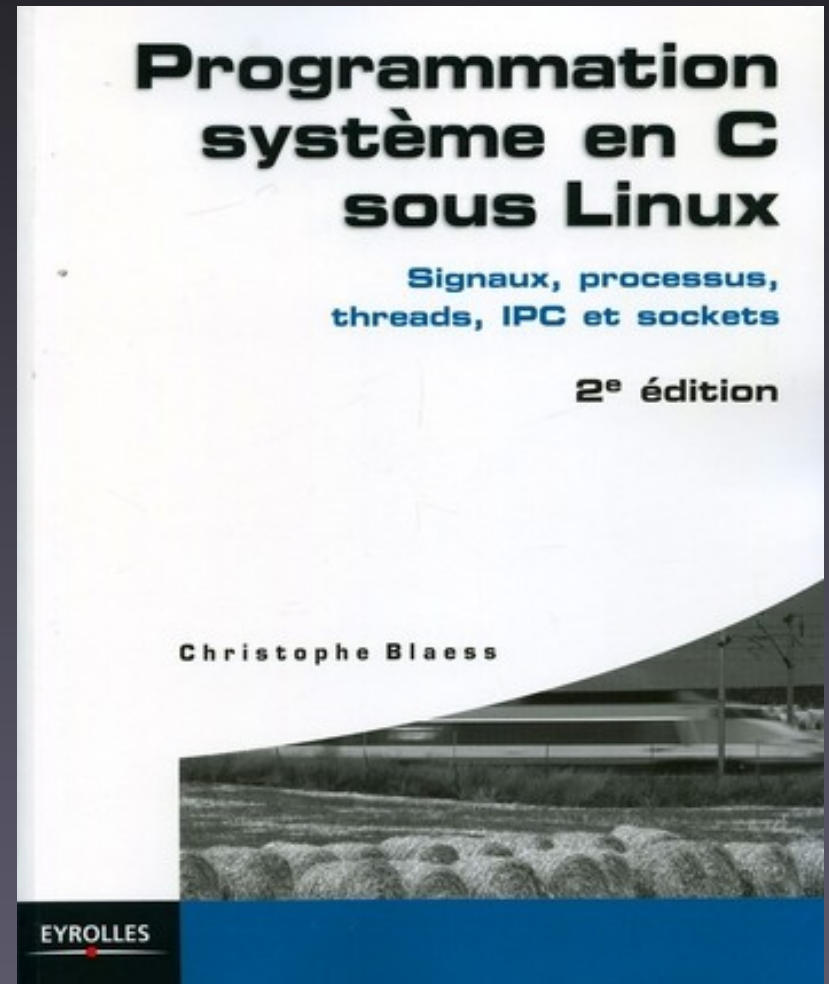
Linux Device Drivers

- Jonathan Corbet,
Alessandro Rubini,
Greg Kroah-Hartman
- O'Reilly
- 614 pages
- la référence en terme
de développement de
drivers
- <http://lwn.net/Kernel/LDD3/>



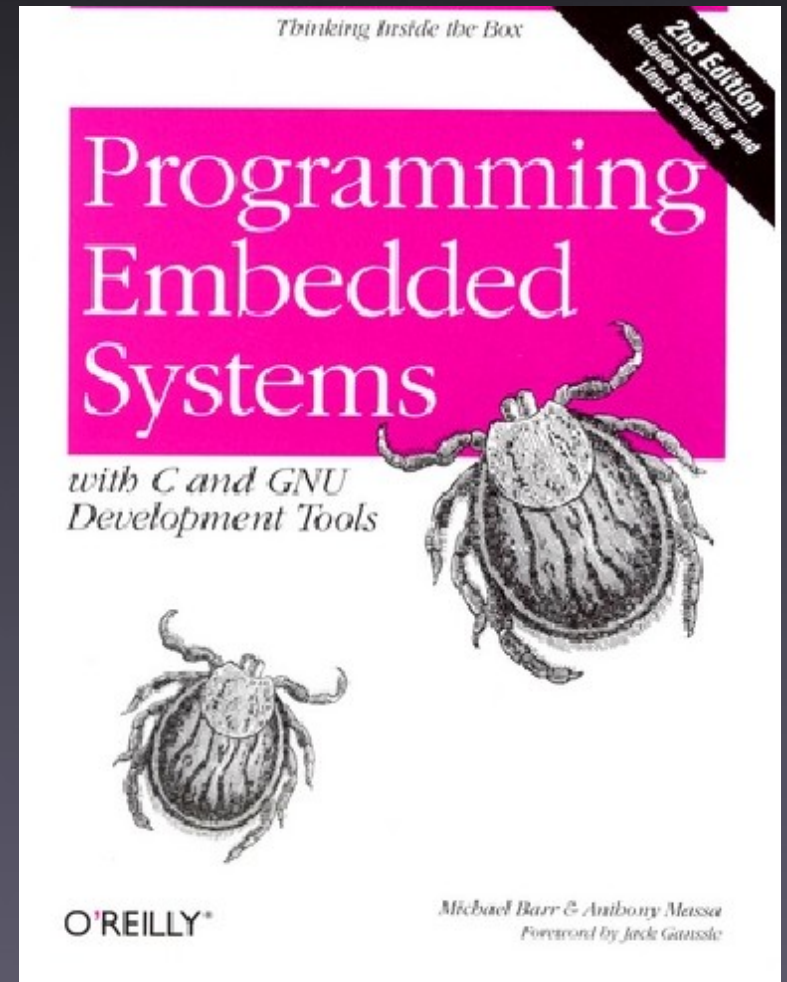
Programmation système en C sous Linux

- Christophe Blaess
- édition Eyrolles
- 964 pages
- un « must-have »
- programmation système avancée sous Linux : signaux, processus, threads, IPC et sockets



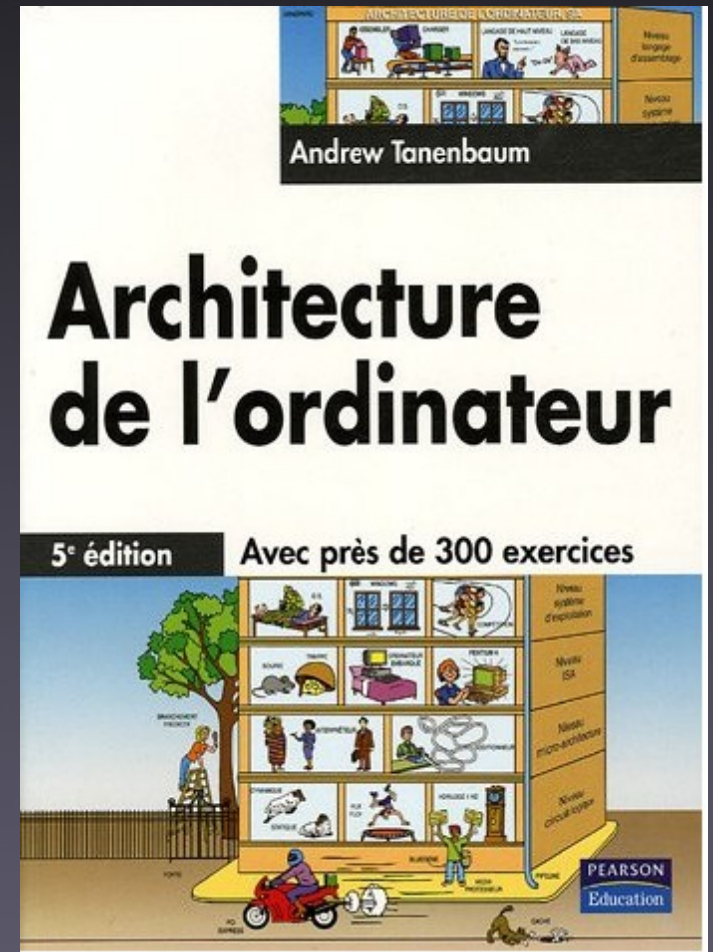
Programming Embedded Systems

- Michael Barr
- O'Reilly
- très bon (mais onéreux pour si peu de pages)
- apprendre à lire le schéma d'un CPU, programmer en milieu contraint et croisé
- Linux/eCos (mais C uniquement)



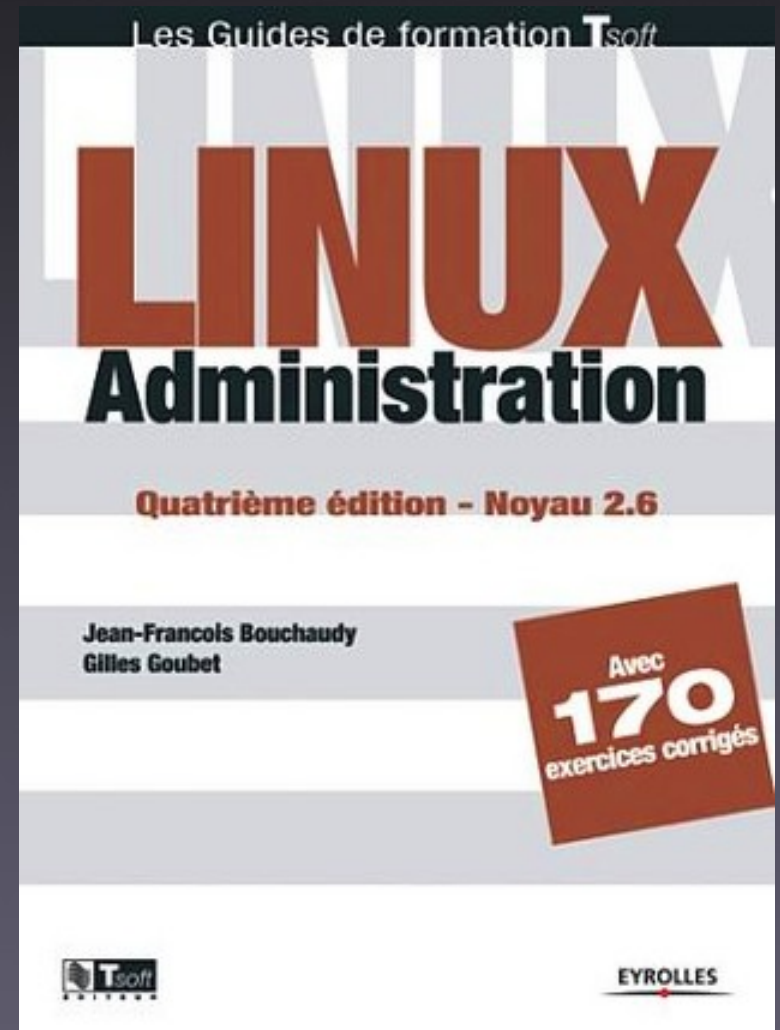
Architecture de l'ordinateur

- Andrew Tanenbaum
- édition Pearson
- les dessous de l'informatique
- comment marche un CPU, l'asm, la pile, les espaces mémoire, etc
- un très grand classique de référence (que personne ne lit...)



Linux Administration

- Jean-Francois Bouchaudy, Gilles Goubet
- éditions Eyrolles
- 936 pages
- tout pour comprendre et administrer un Linux
- pour débutants et experts



Sur le web

- <http://elinux.org/>
- <http://www.linuxdevices.com/>
- http://rt.wiki.kernel.org/index.php/Main_Page
- <http://kerneltrap.org/> -- <http://kernelnewbies.org/>
- <http://lxr.linux.no/> -- http://www.linuxdriver.co.il/kernel_map
- <http://free-electrons.com/>
- <http://www.openembedded.org/> -- <http://www.mvista.com/>
- <http://www.realtimelinuxfoundation.org/>
- etc
- <http://gblanc.blogs.linagora.com/>

En kiosque

- Linux Magazine
 - avec HS spécial électronique
 - de plus en plus de place pour l'embarqué
 - Denis Bodor, Patrice Kadionik, etc
- Linux+ DVD
- Linux pratique
- Planet Linux

En cours

- C'est tout de même le plus simple
- Parce que les DIF sont aussi rares que chers
- Mais **Linagora** intervient aussi les droits à la formation (et on est très bien reçu) :
 - Linux temps réel
 - Linux embarqué
 - Programmation C avancée sous Linux

Demo time

Le matos

- Une carte de développement
 - Embest SBC2440
 - ARM920t S3C2440
- Des câbles
 - J-TAG
 - série/USB
 - RJ45
- Un PC
 - Gdb/OpenOCD/NFS

L'idée

- On branche un peu tout comme il faut
- On envoie un Linux en mémoire RAM par J-TAG
- On met les registres dans le bon état
- On boote le Linux toujours par debugger
- On va chercher le système de fichier root par NFS
- Et voilà !
- Détails sur <http://gblanc.blogs.linagora/>
<http://gblanc.blogs.linagora.com/index.php/post/2008/10/24/portage-Linux-Emb2440>
<http://gblanc.blogs.linagora.com/index.php/post/2008/08/29/J-TAG-sur-carte-Embest-SBC244>

Questions ?