

Embedded Linux

Petite introduction pour conférence sommaire
-- PARINUX --

Gilles BLANC, palpatine42@gmail.com

/me

- ingénieur informatique embarqué/temps-réel (Linagora)
- EPITA GISTR 2006
- 100% sous Linux depuis 2004
- libriste et membre Parinux/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 STB, sous Linux

Intro



Avez-vous ?...

- une {free, live, darty, alicia, 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 ?
- un PC tout intégré « easy »-like ?
- du CPL ? un téléphone wifi ?
- bien d'autres choses.....

Bein alors...
Vous avez Linux !

(qui a dit que c'était compliqué,
déjà ??)

(z'en avez p'têtre même plus que de l'aut' machin de
chez Redmond...)

Qu'est-ce que Linux ?

architecture software

- Qu'est-ce qu'un OS ?
 - un noyau (kernel)
 - des logiciels
- les drivers / modules
- l'héritage *NIX, le « tout fichier »
- Linux, GNU/Linux

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 ?

« 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, avions
 - missiles, radio militaire
 - 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
- contrainte de prix de développement (pur et licences)
- contrainte de prix de vente (amortissement et royalties)

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
 - début 84 : lancement du projet GNU par RMS
 - octobre 85 : FSF
 - fin 90's : projet d'OS finalisé (gcc, binutils, glibc, emacs), à l'exception du noyau
 - juin 91 : GPLv2
 - 91 : Linux par Linus Torvalds
 - 2007 : GPLv3
- Étendu du libre : centaine de milliers de projets ; en 2004, ~70% sous GPL

- Au-delà de Linux embarqué
 - outils portables ou adaptables entre OS :
 - gcc, gnat
 - busybox
 - eclipse
 - gdb
 - u-boot
 - freeBIOS !
 - exemple d'utilisateurs propriétaires :
 - WindRiver : VxWorks
 - LynuxWorks : LynxOS
- Au-delà de Linux : eCos (Solaris ?)
- Au-delà du projet GNU : NetBSD

- Les formats ouverts

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

- échanges de données
- stockage de données
- systèmes softwares
- systèmes hardware

- les avantages

- pérennité
- interopérabilité
- coût !

- exemples :

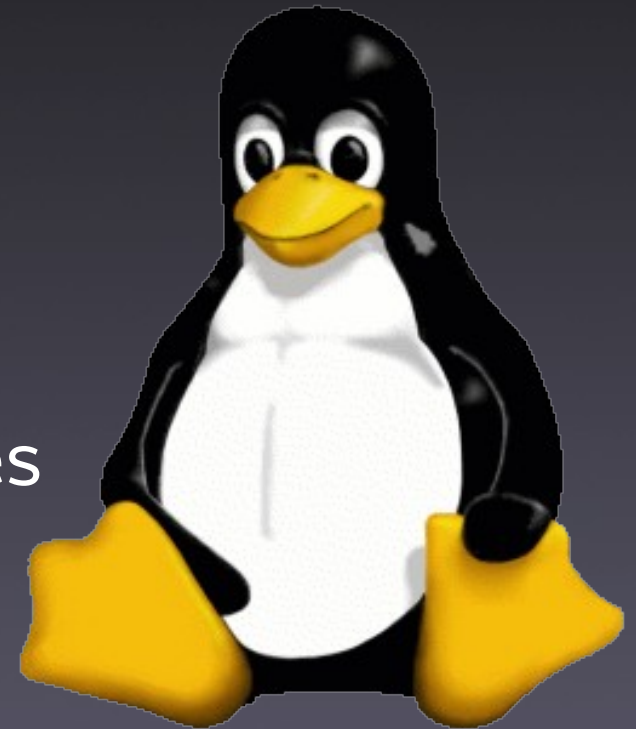
- communication : XMPP
- audio/vidéo : ogg Vorbis/Theora (décodeur Tremor)
- images : png

Pourquoi le libre ?

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

Le noyau Linux

- Pourquoi Linux ?
 - libre : GPLv2, forte évolutivité
 - pas de royalties
 - sources gratuites
 - nombreux points forts techniques
- Présentation générale
- cible/marché
- temps réel : « mou » ou « dur » ?



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
- O(1), préemptible (mais le 2.4 ne l'était pas !)
- beaucoup d'outils disponibles, de libraries
- développement userfriendly (hôte ~ cible)
- TCP/IP (serveurs !), bluetooth, multimédia,... : de base

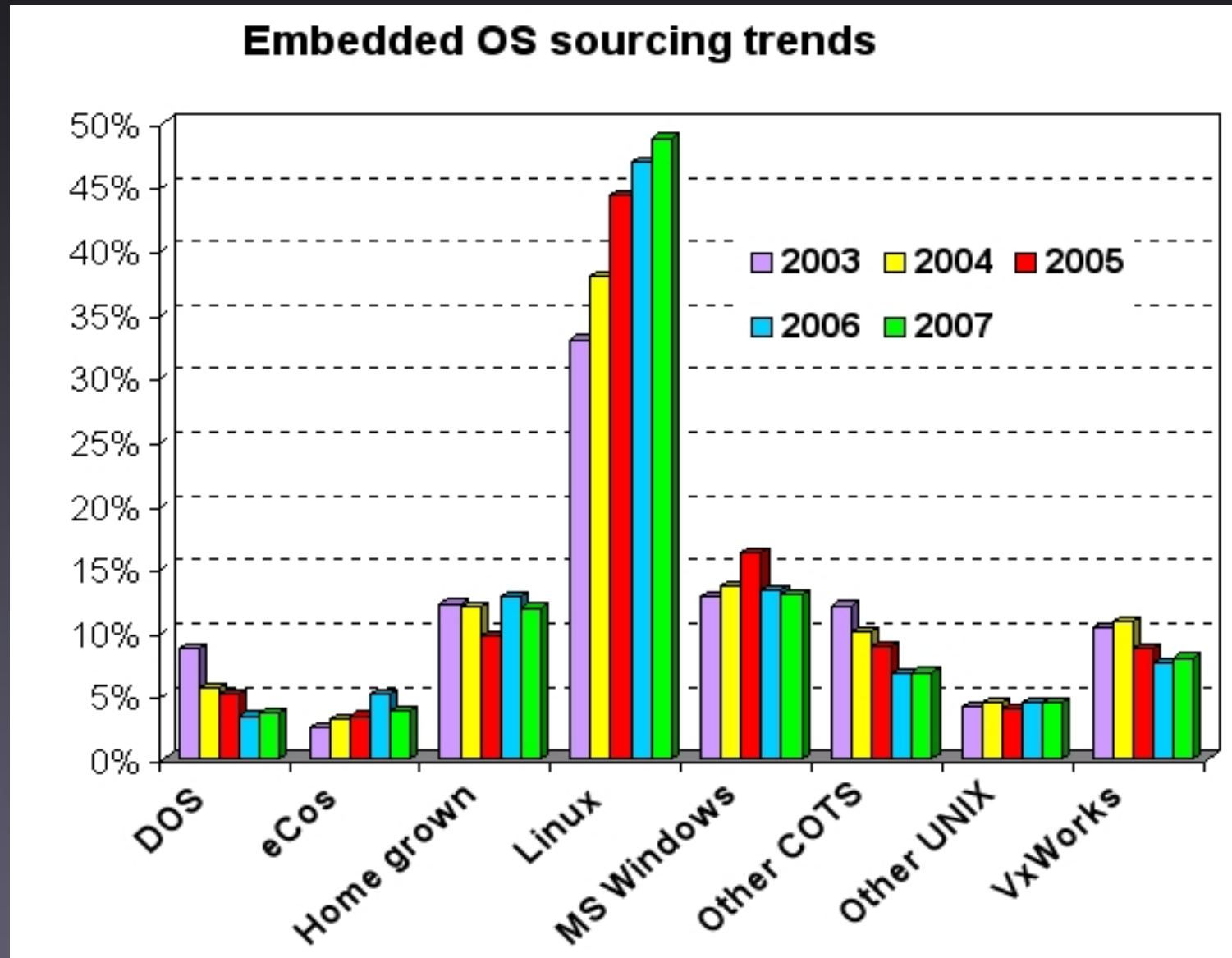
Get the facts

Dans l'embarqué

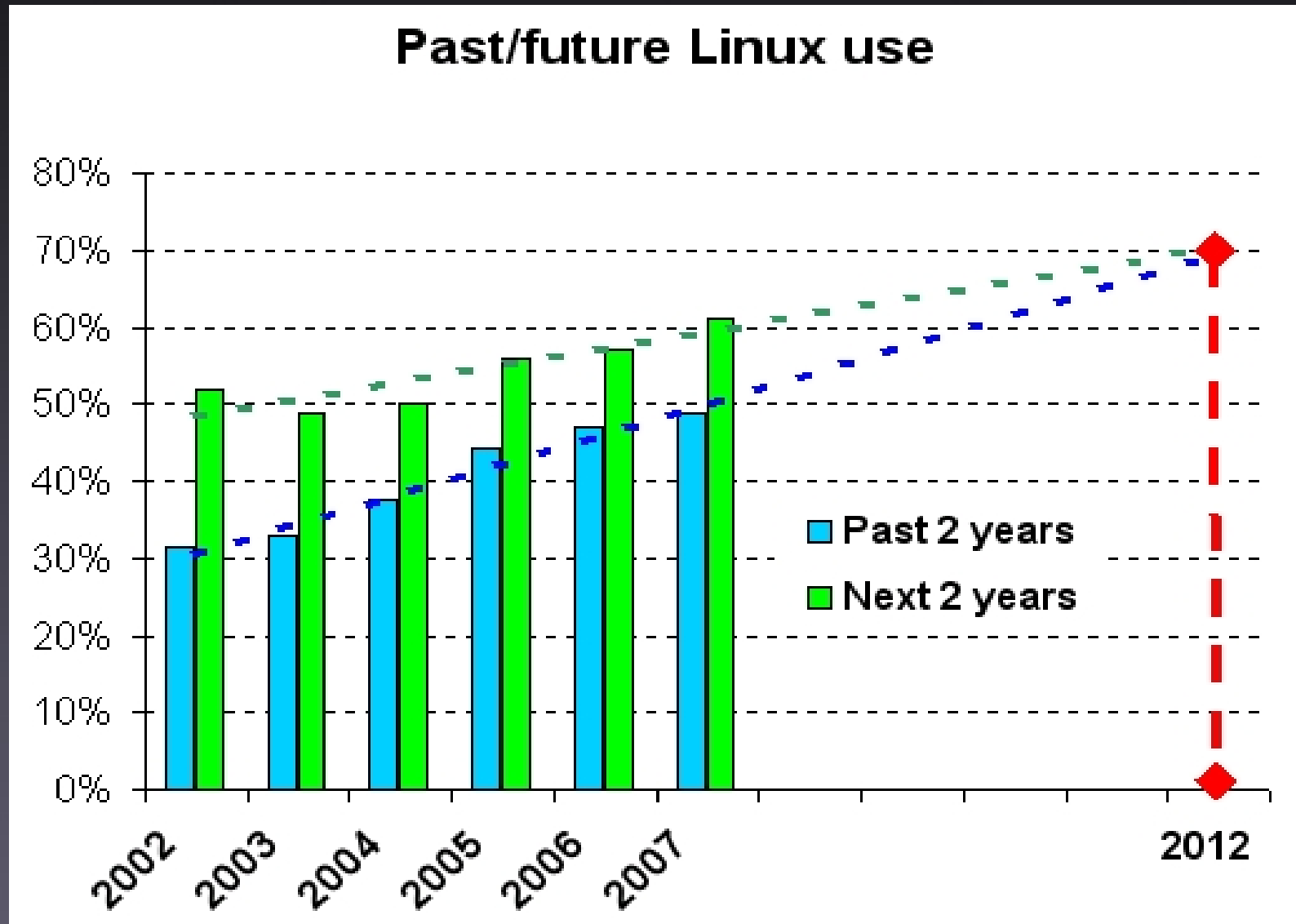
- peu de données 100% fiables
 - culture du secret industriel
 - parfois militaire
 - projets en cours, non finalisés
- études : « snapshot of the embedded Linux market »
<http://www.linuxdevices.com/articles/AT7065740528.html>
- les salons confirment un intérêt croissant et soutenu



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



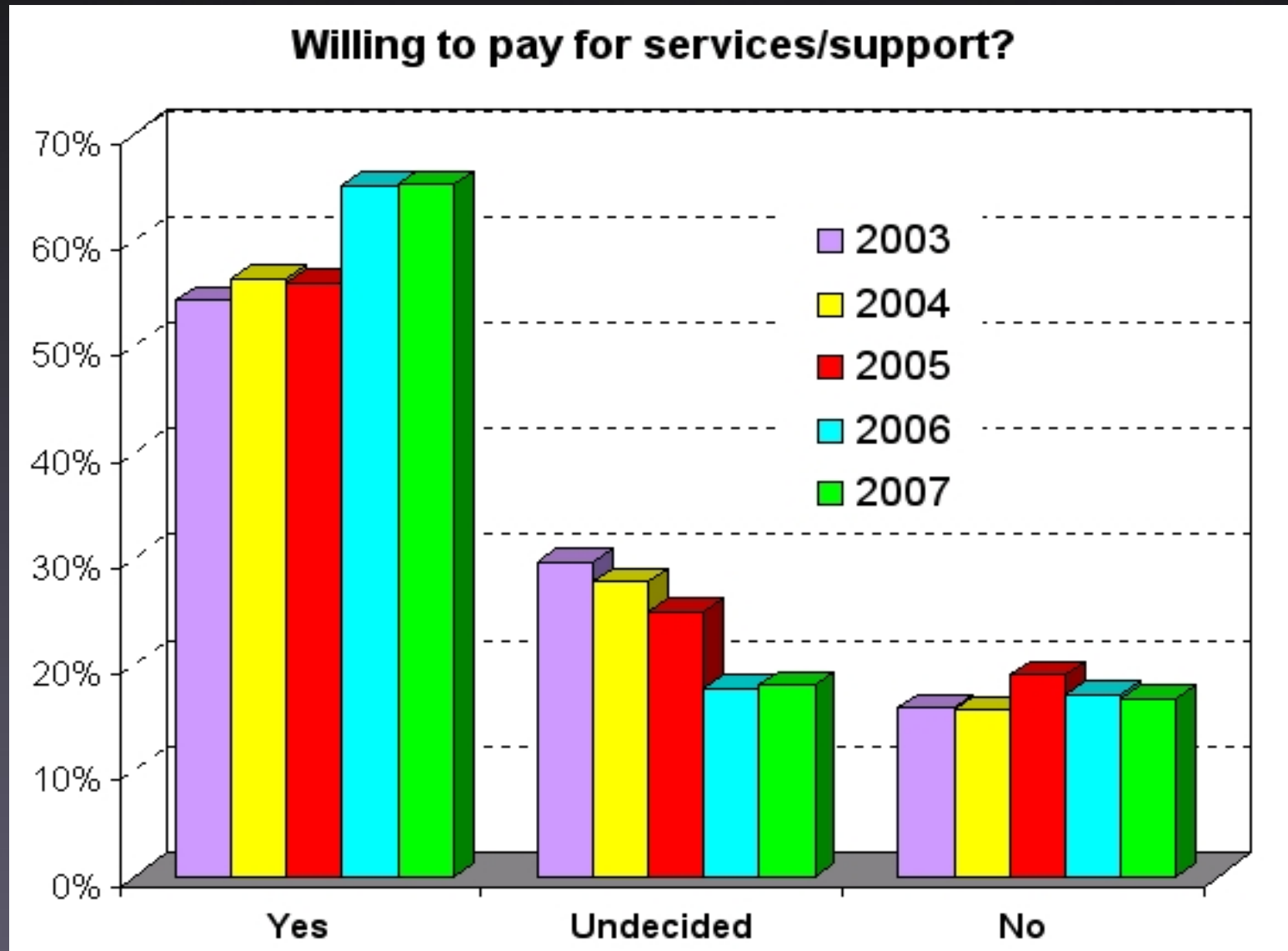
Actual and planned Linux use may converge by 2012



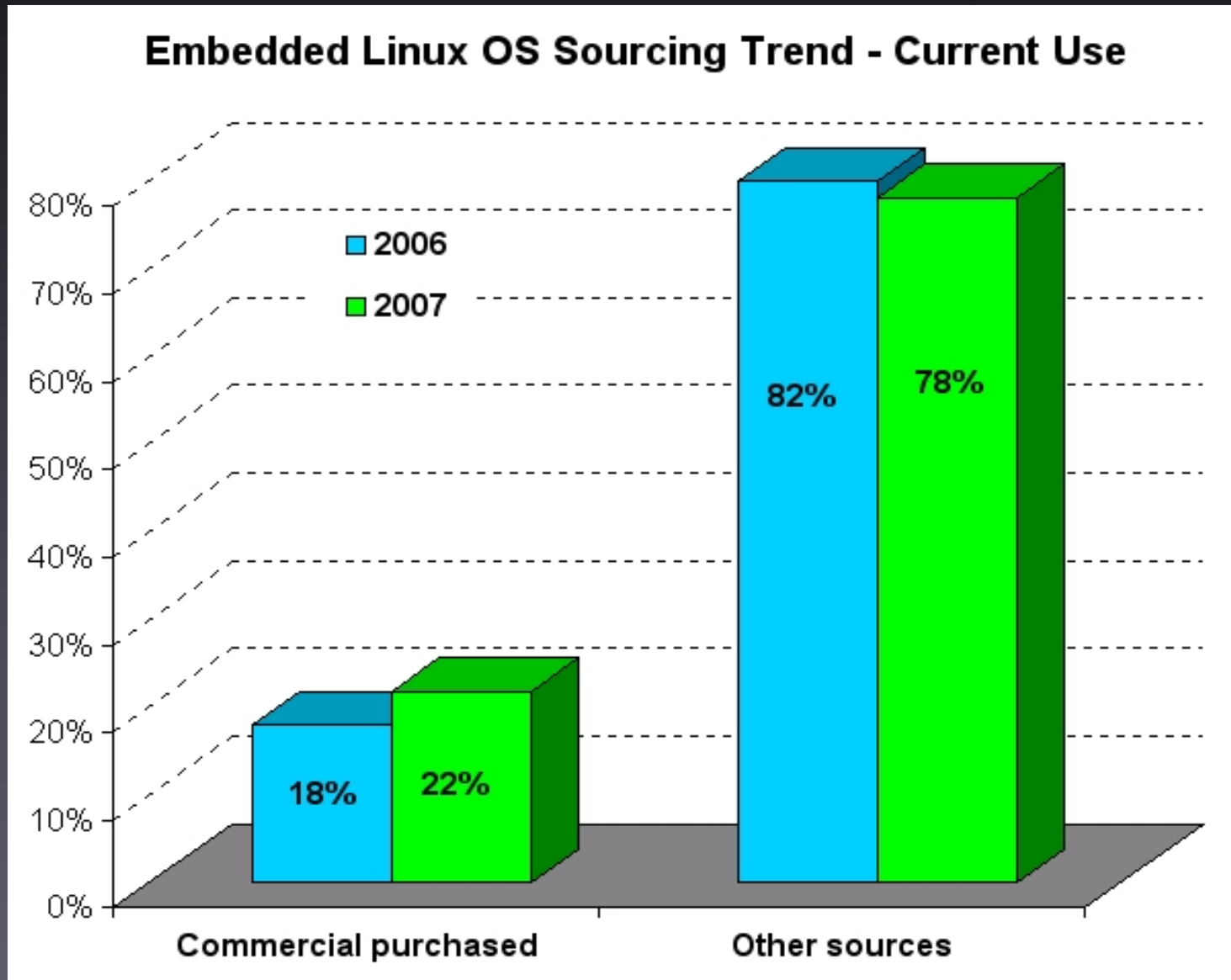
Le choix

- commercial vs gratuit / communautaire
- temps réel mou, patch de préemption, ou temps réel dur ?
- distribution actuelle et tendance
- l'avantage et l'écueil du packaging

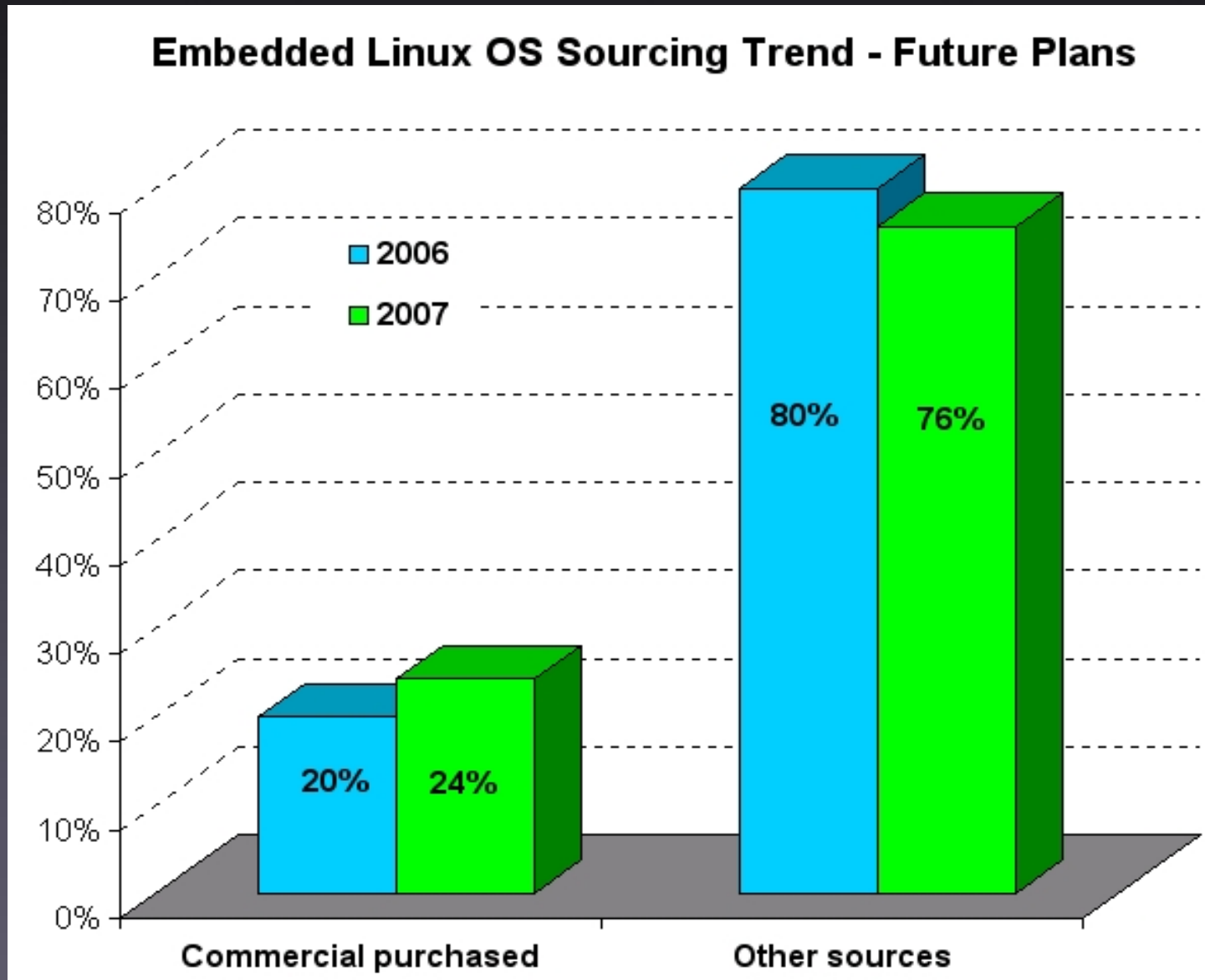
Would you consider paying for support or services?



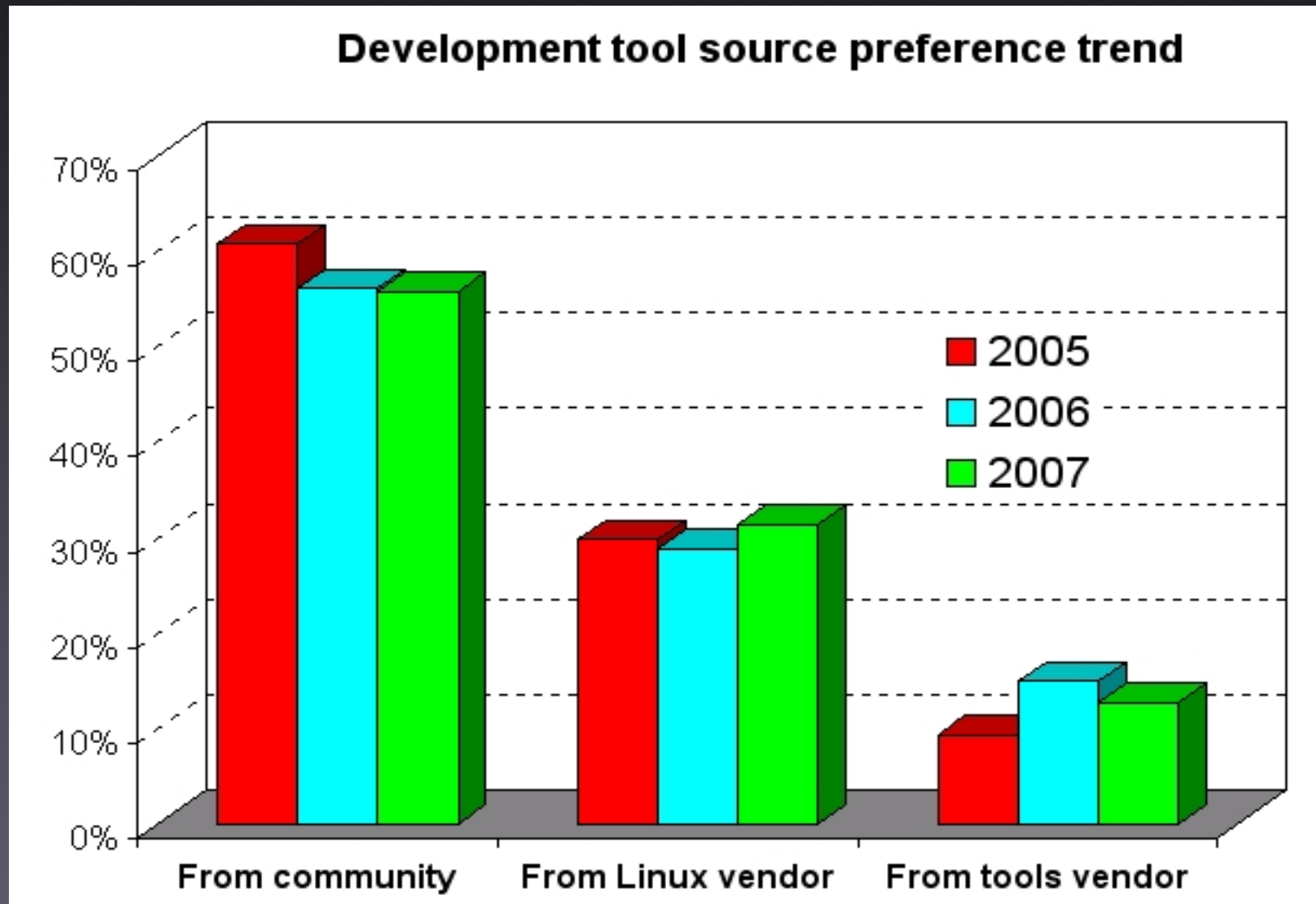
How did you obtain your (company's) embedded Linux software?



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

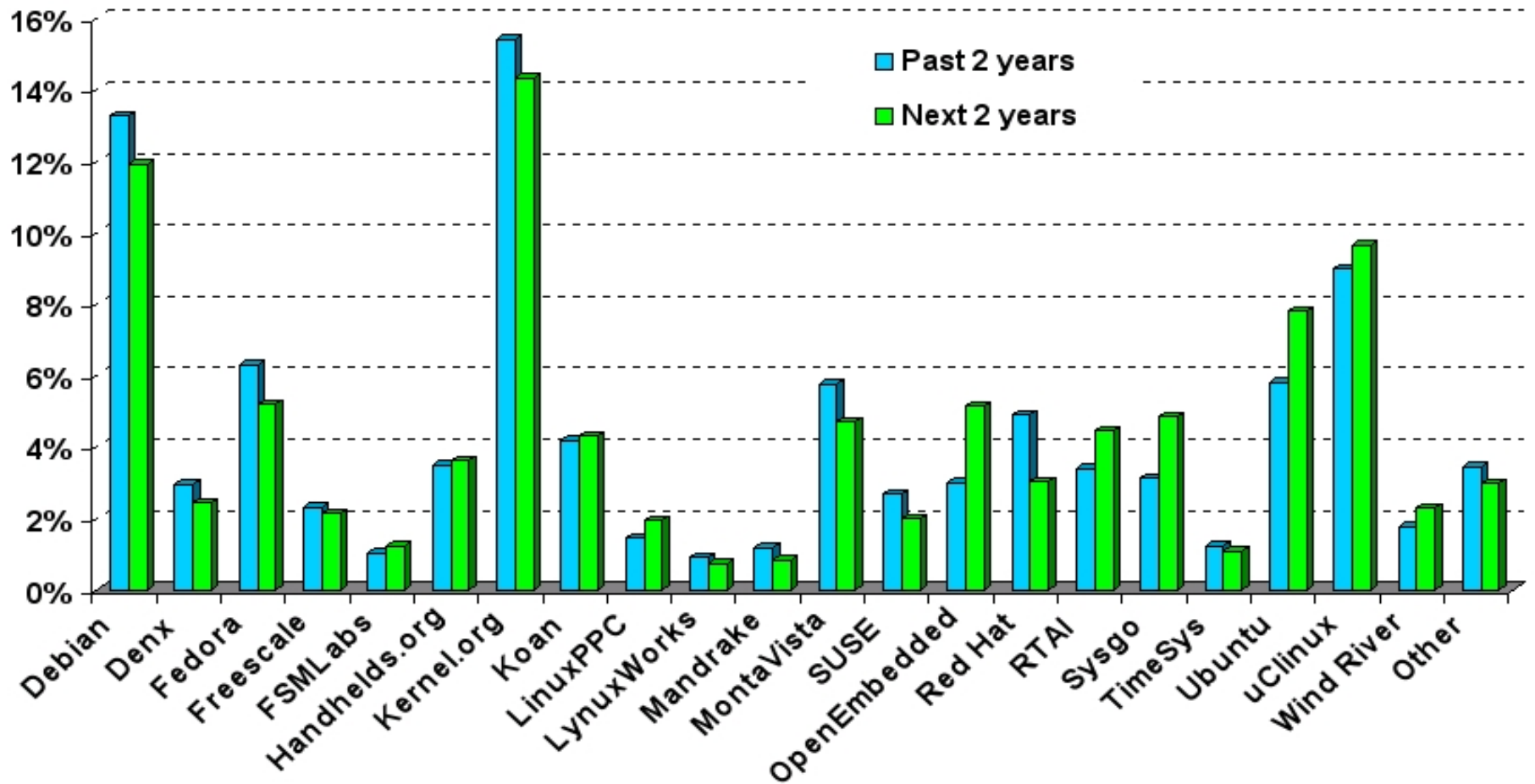
- out of the box
 - debian, RedHat/Fedora
- ready to embed
 - openEmbedded, WindRiver, Sysgo, LynuxWorks, etc
- Real Time
 - MontaVista, Concurrent Computing, etc
- environnements adaptés
 - Maemo (Hildon), openMoko

La solution « facile »

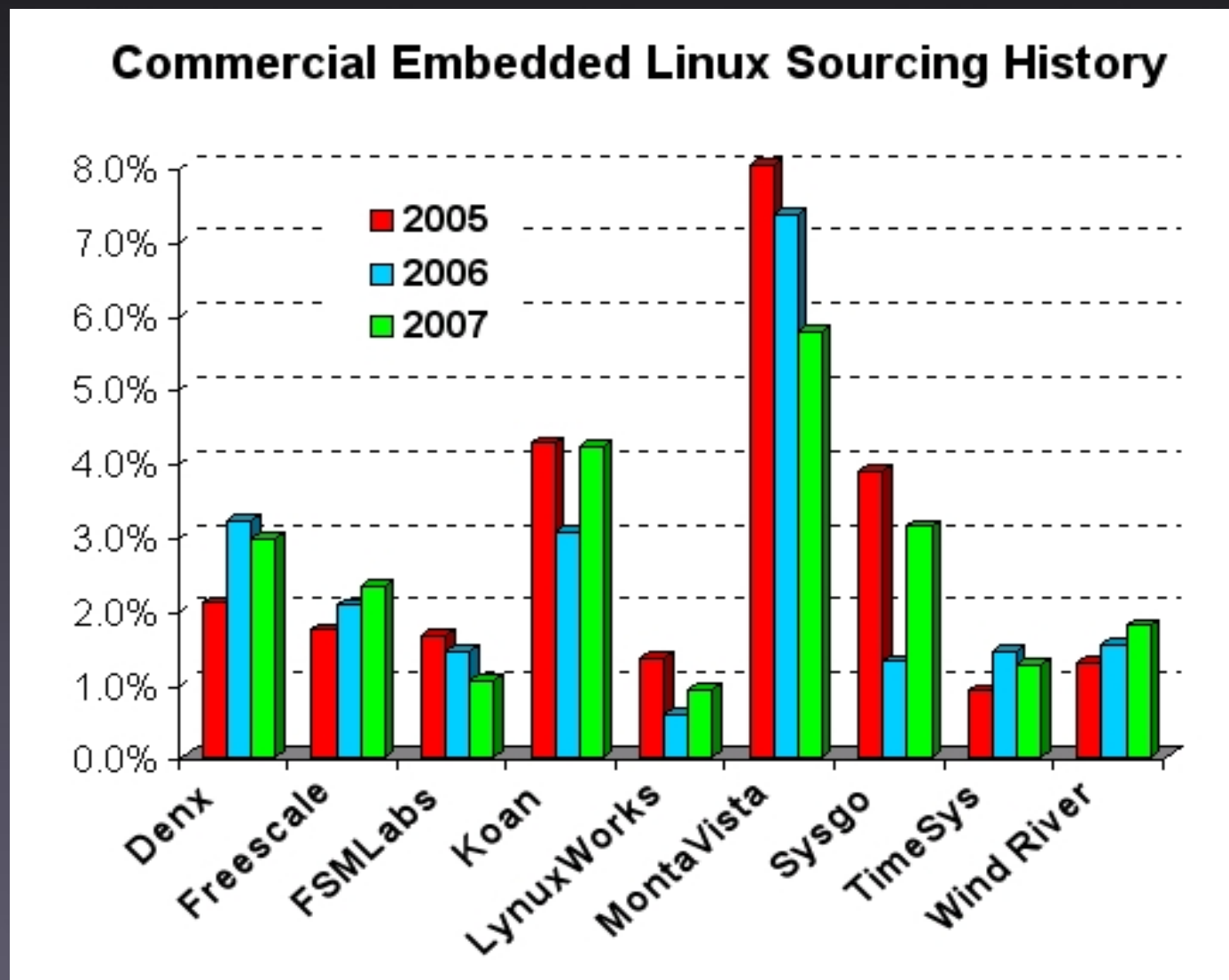
- exemple : sur carte VME compatible (Gaci, Ecrin, Thales, ...)
- distribution standard, « desktop »
- « temps réel mou », noyau préemptif 2.6
- kernel « vanilla »
- Fedora impaire compatible RedHat Enterprise (raisons historiques)
- ou Debian : 11 architectures stables

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

Embedded Linux OS sourcing trends



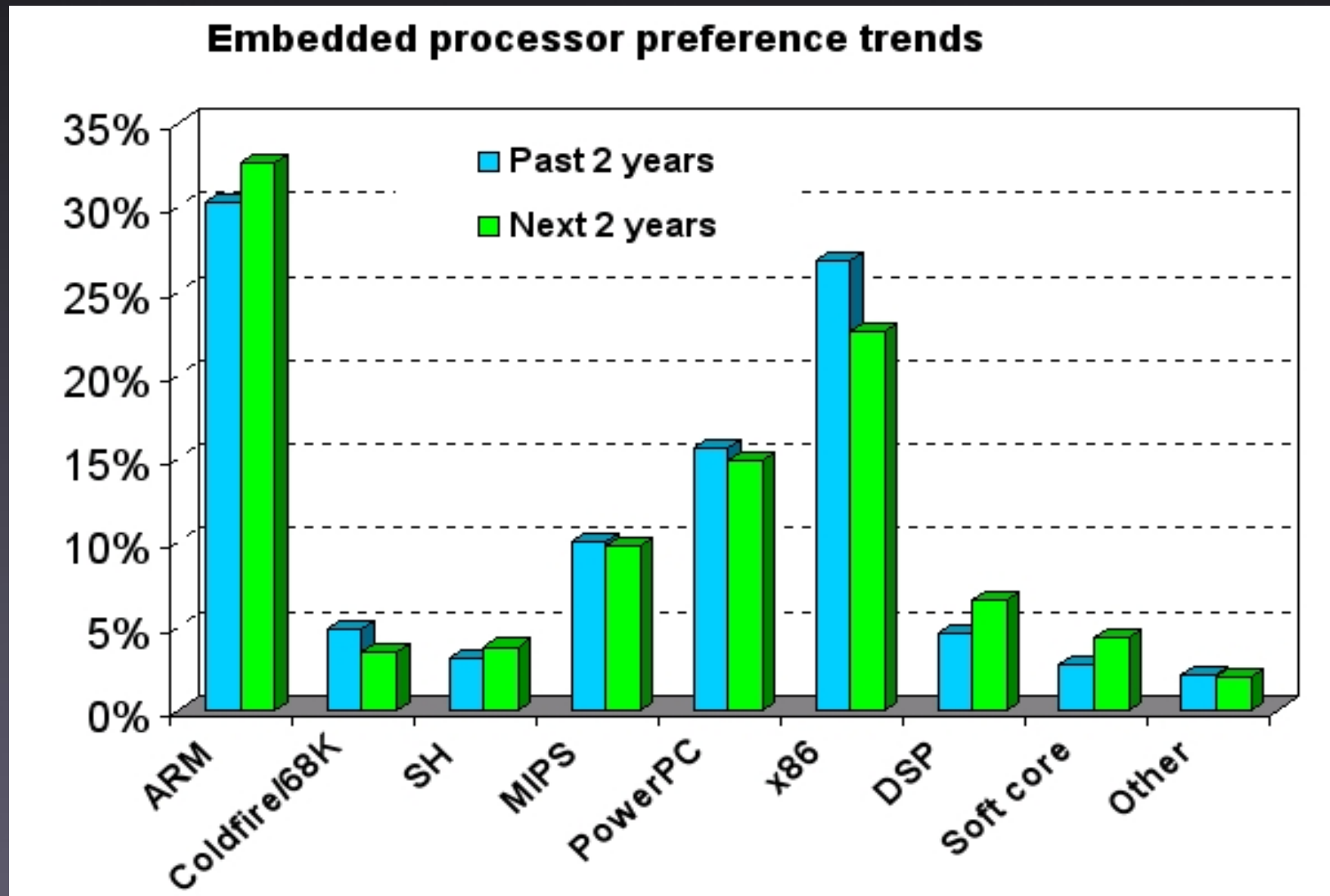
What sources of Linux were in your (company's) embedded designs during the past two years? -- historical trend



Les cibles

- ~75 architectures CPU gérées
 - 32bits minimum
 - MMU requise (sauf pour μ Clinux)
- les principales en TR/embarqué :
 - (Strong)ARM/Xscale: communication (portables) (TI OMAP), assistants personnels (ARM9 DragonBall)
 - 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

L'existant

- 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, 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

- Qtopia : le Qt embedded
 - trolltech, licence Qt (GPL si projet libre, commercial sinon)
 - 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)
 - Sugar (OLPC)
- Android (« gPhone »)

Conclusion

Avantages

- matière première gratuite, abondante
- code de qualité, ouvert
- plate-forme de dev idoine (tester sur place)
- support matériel (si si !)
- 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)
- 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

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
(ie le savoir-faire)
- beaucoup de compétences requises, de la patience, des talents en haute cuisine

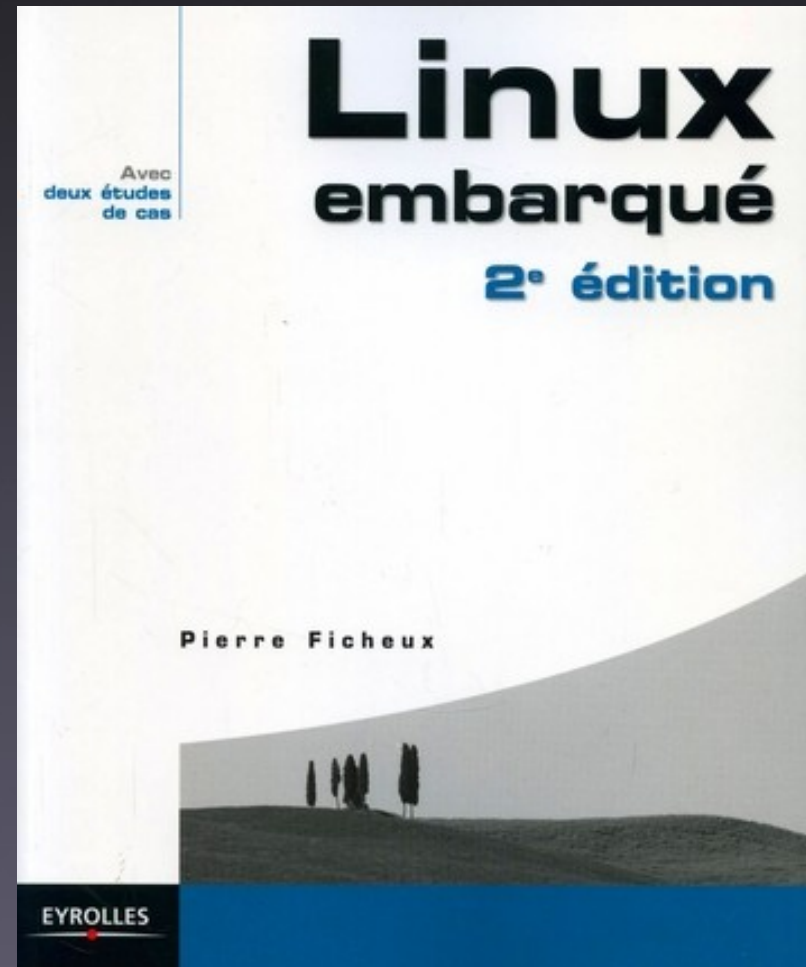
Futur proche et lointain

- amélioration des performances
- (para)virtualisation
- drivers userspace / API générique ?!
- toutes les STB ? (m\$ bouge encore !)
- les téléphones
- à la conquête du monde

Aller plus loin
(par soi-même)

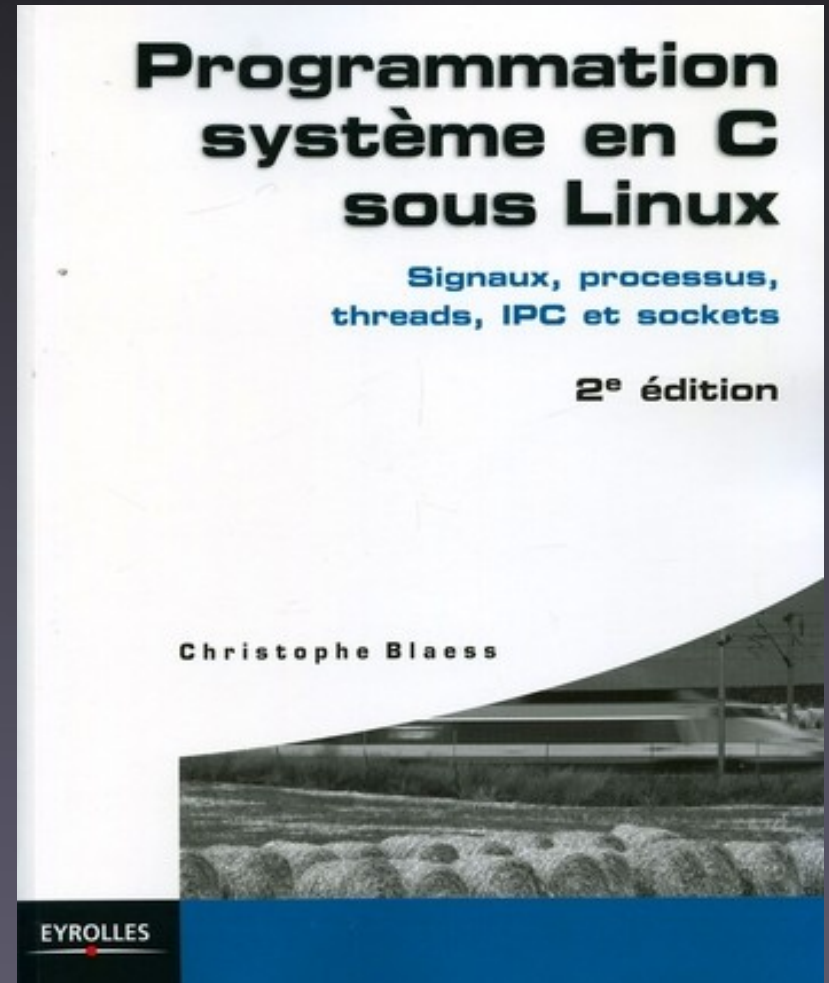
Linux Embarqué

- Pierre Ficheux
- édition Eyrolles
- 330 pages
- la bible absolue
- comment construire une distrib' complète



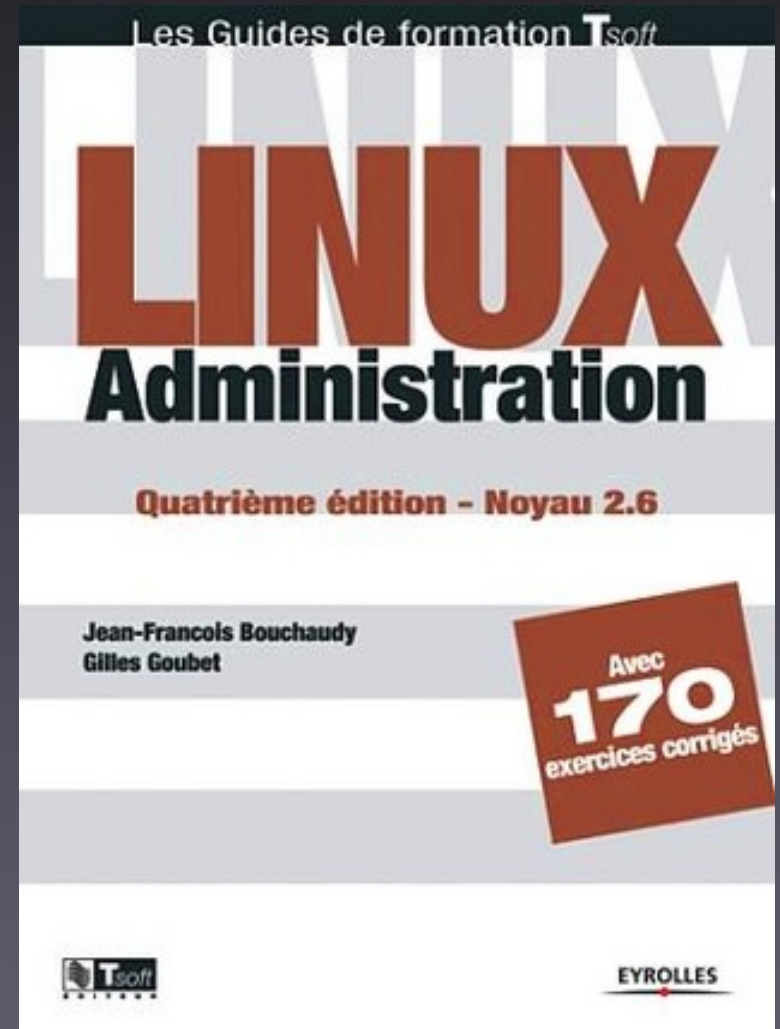
Programmation système en C sous Linux

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



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://www.realtimelinuxfoundation.org/>
- <http://www.linuxdevices.com>
- <http://kerneltrap.org>
- <http://www.gnu.org/>
- <http://lxr.linux.no/>
- <http://www.openembedded.org/>
- <http://www.mvista.com/>
- <http://free-electrons.com/>

En kiosque

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