

Systèmes embarqués et logiciel libre

Pierre Ficheux (pierre.ficheux@openwide.fr)

Décembre 2011

- Quelques mots sur le logiciel libre
- Historique
- Freeware, shareware / open source
- Licences GPL/LGPL
- Les principaux projets: X11, GNU, Apache, Linux
- Avantages/inconvénients pour l'embarqué
- Linux: domaines d'application (quand l'utiliser ou pas)
- Exemples réels
- Limitations, autres solution libres pour l'embarqué

- A peu près équivalent à la notion d'*open source*, voir <http://www.opensource.org>
- La confusion vient de la signification anglaise
 - free = libre / gratuit
- Différents types de logiciels
 - Le *freeware* ou *graticiel*: gratuit mais sources *non disponibles*, pas forcément de licence (abandon de la « paternité » du code)
 - Le *shareware*: sources non disponibles, coût modique, licence souvent propriétaire
 - Le *logiciel libre*: sources disponibles, licence *open source*, non liée à la notion de gratuité (on peut vendre un logiciel libre)

- Le logiciel libre n'est pas forcément gratuit
- La migration vers du logiciel libre n'est JAMAIS gratuite => le « coût du changement »
- Les licences, même libres *doivent* être respectées (cf assignation de Free par la FSF)
- Ces licences ne sont pas incompatibles avec la vente de logiciel (embarqué ou non) ni de matériel
- La conception de l'architecture du système est *fondamentale* afin de prévenir les problèmes de licence

- Doivent être compatibles avec le modèle libre, voir <http://www.opensource.org>
- Environ une centaine de licences
- Problème: compatibilité avec les droits nationaux
- Exemples de licences :
 - GPL / LGPL
 - BSD
 - MIT/X11
 - Mozilla
 - FDL (pour la documentation)
 - CeCILL (France)
- Voir http://fr.wikipedia.org/wiki/Licence_libre

- GPL = General Public License
- On la surnomme également « copyleft »
- La GPL v2 (1991) est la plus répandue (ex: noyau Linux)
- La licence s'applique uniquement en cas de redistribution
- Un code source utilisant du code GPL est du travail dérivé et doit être publié
- Publication: celui qui reçoit la version binaire peut obtenir le code source
- Pas de lien (ld) possible entre du code GPL et du code « propriétaire » !

- La GPL est complexe à gérer dans un contexte industriel → création de la LGPL
- Le lien avec du code propriétaire est possible avec la LGPL (*Lesser/Library* GPL) !
- En majeure partie, les bibliothèques système sont diffusées sous LGPL (exemple: GNU-libc)
- Dans le cas d'une application propriétaire il faut donc vérifier qu'aucune bibliothèque « liée » n'est GPL
- Le lien dynamique n'affranchit pas de la licence sauf dans des cas très particuliers



- Nouvelle version sortie en 2007
- Oblige à fournir les éléments pour construire un logiciel fonctionnel => réponse à la « Tivoisation »
- La GPL v2 demande uniquement la publication des sources à celui qui a reçu le binaire
- La GPL v3 ne sera pas utilisée pour le noyau Linux, voir <http://lwn.net/Articles/200422>
- Voir:
<http://www.gnu.org/licenses/quick-guide-gplv3.fr.html>

- Dans l'espace noyau (pilotes), SEULE la GPL s'applique (en théorie) !

You cannot use kernel headers file to create non GPL'd binaries (Linus Torvalds)

- Message *license XXX taints kernel* au chargement
- Certaines fonctions non disponibles si la licence n'est pas GPL
- En pratique: tolérance si le pilote n'a pas été créé pour Linux (cas du portage) => nVidia
- Cependant les pilotes binaires posent des soucis techniques vu qu'un pilote fonctionne pour la version de noyau utilisée pour la compilation

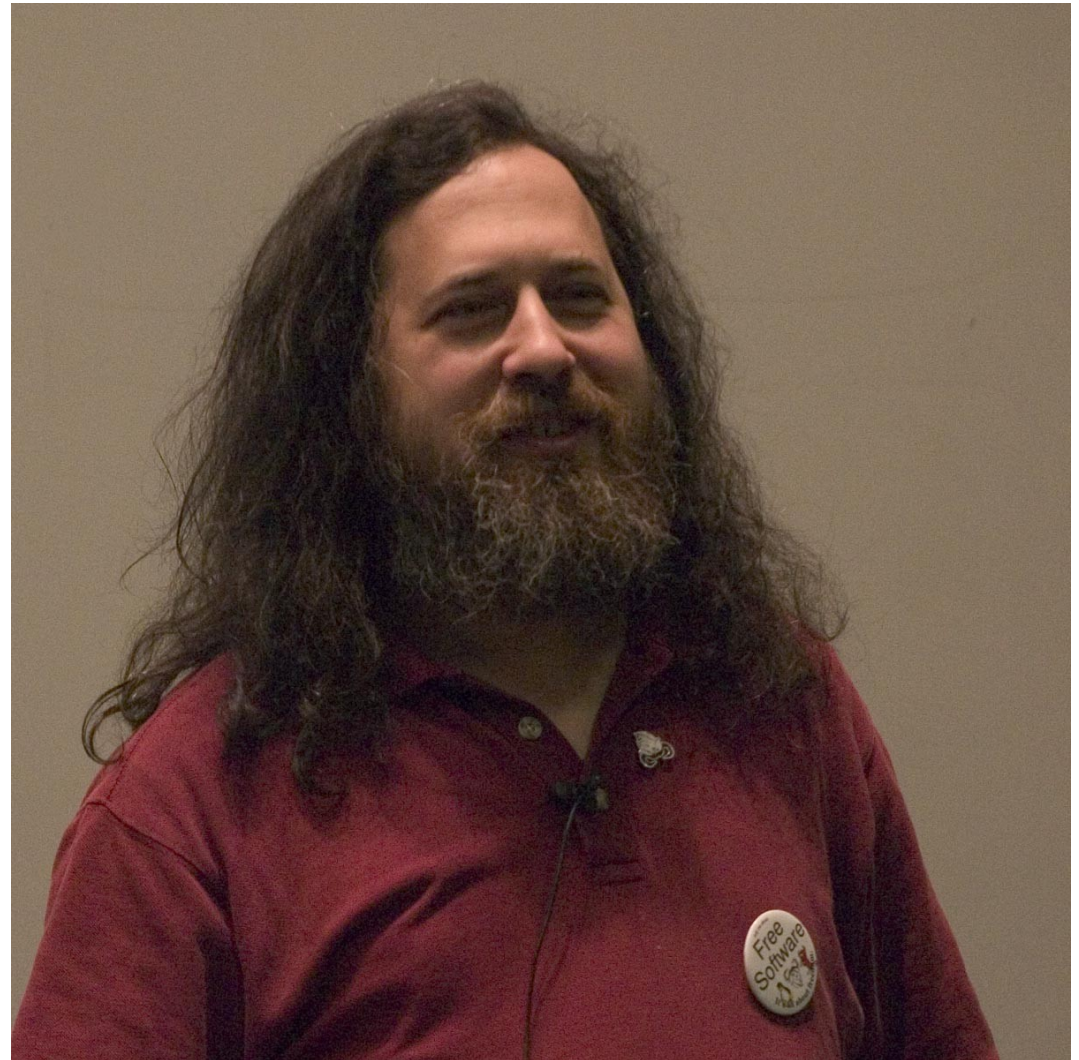
- X Window System (X11), MIT (MIT/X11)
- GNU (GPL/LGPL)
- FreeBSD (BSD)
- Apache (Apache)
- Linux (GPL)
- Eclipse (EPL)
- Mozilla / Thunderbird (MPL)
- WebKit (BSD/LGPL)
- Qt (LGPL)
- Android (partiellement)

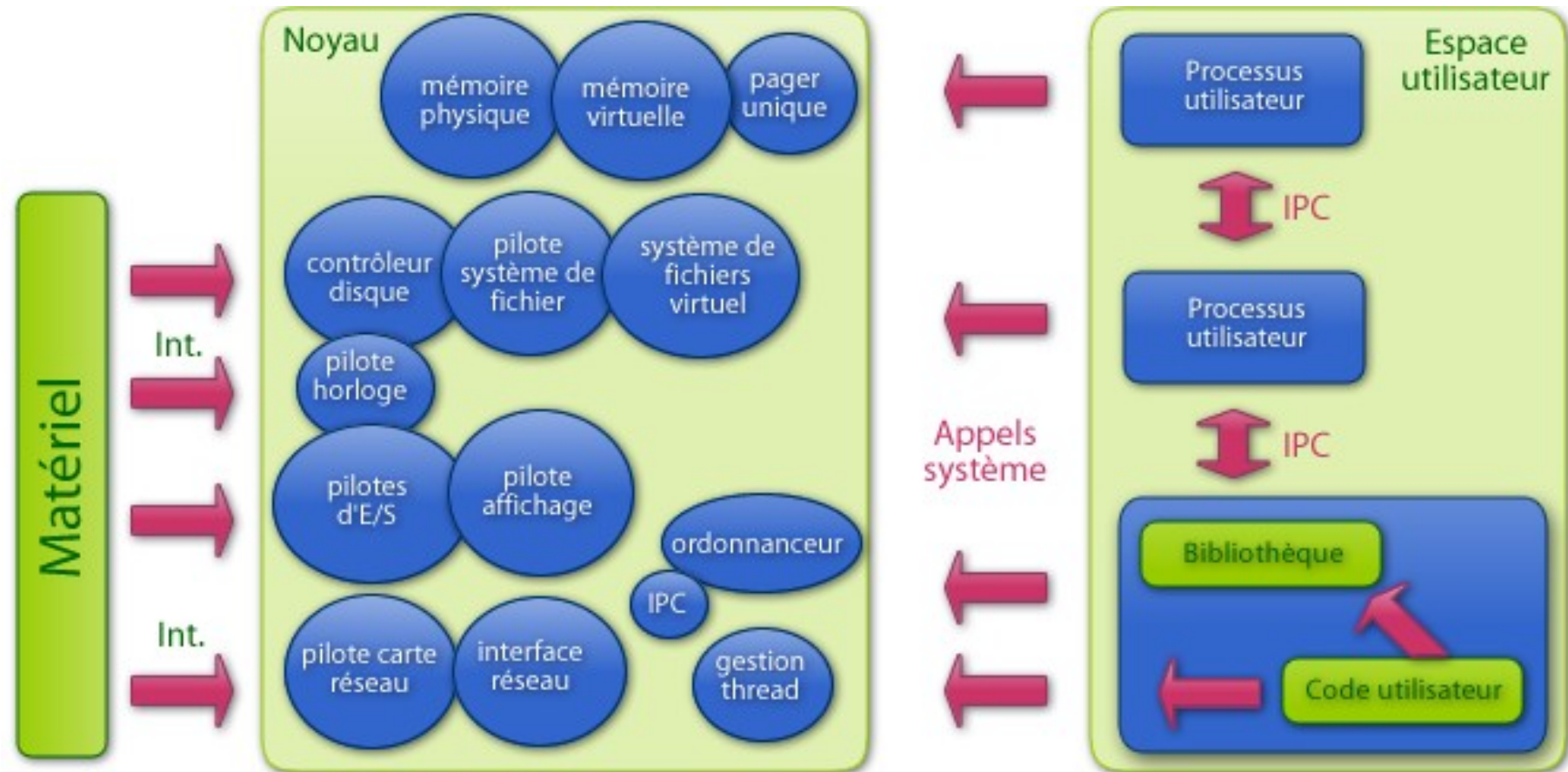
- Linux est une implémentation libre d'UNIX diffusé sous licence GPL
- Inspiré des deux versions: AT&T et BSD
- Les principes d'UNIX sont respectés:
 - Simplicité, modularité, respect des standards, ouverture
 - Deux espaces de mémoire: noyau / utilisateur
 - Le noyau permet d'accéder au matériel (pilotes, appels système)
 - Tout composant est un fichier: répertoires, périphériques, éléments de communication, etc. (organisation arborescente)
 - Puissance de la ligne de commande (shell et expressions régulières)

- Noyau monolithique (en un seul fichier) + modules dynamiques
- Création des processus via `fork()/exec()`
- Multi-threading natif (NPTL depuis 2003)
- Nombreuses piles réseau (IPv4, IPv6, Ethernet, etc.)
- Organisation des fichiers arborescente à partir de la racine (/), montage et démontage *logique* (`mount`)
- Notion de super-utilisateur (*root*), groupes, et utilisateurs

- Linux est fortement lié au projet GNU (GNU is Not UNIX) de Richard Stallman (MIT, années 80)
- Le nom officiel de l'OS Linux est *GNU/Linux* car « Linux » correspond uniquement à la partie noyau
- Libre de toute licence source par rapport à AT&T
- Internet a également fortement contribué au succès de Linux
- Pour un non initié, la personnalité de Linus Torvalds est plus « rassurante » que celle de R. Stallman :-)

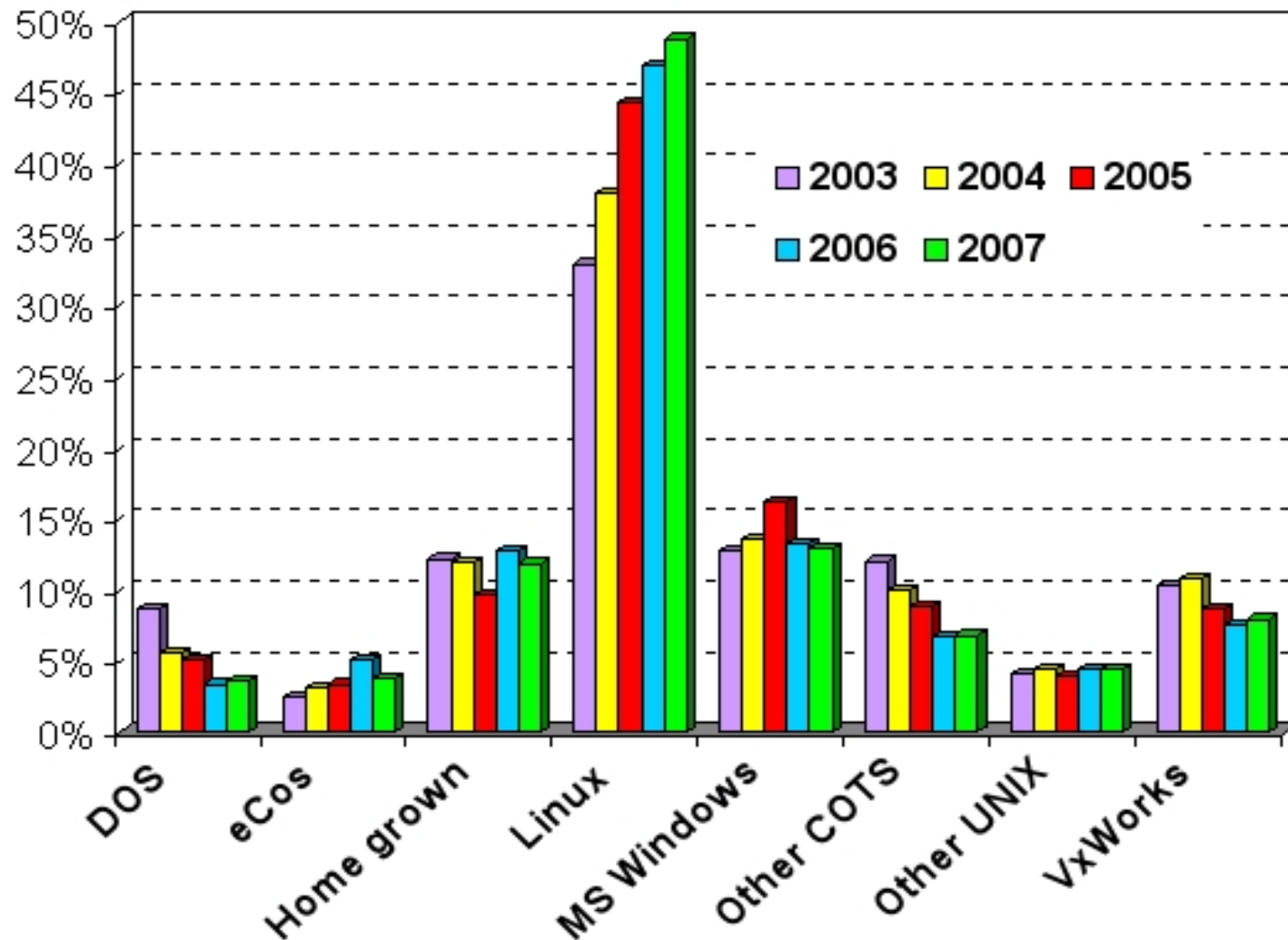
Les *parents* de Linux !





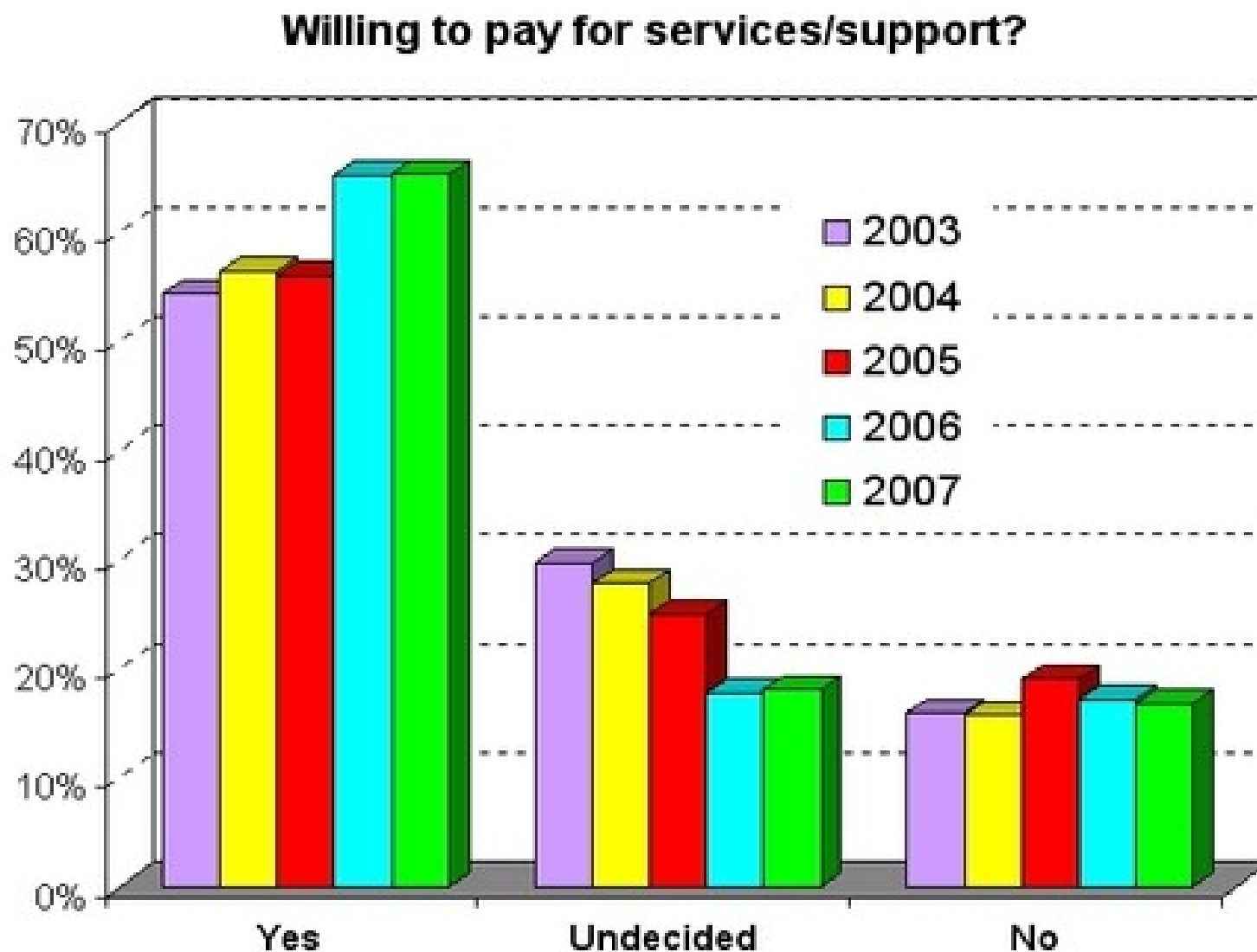
- Avantages
 - Disponibilité du code source: *maintenabilité* dans le temps
 - Redistribution sans *royalties*
 - Développement dérivé de code source existant
- Inconvénients
 - Méfiance des décideurs, modèle *décentralisé* apparaissant comme flou
 - Contraintes des licences (GPL, LGPL)
 - Support technique parfois non officiel pour certains matériels (mais la situation évolue !)

Embedded OS sourcing trends





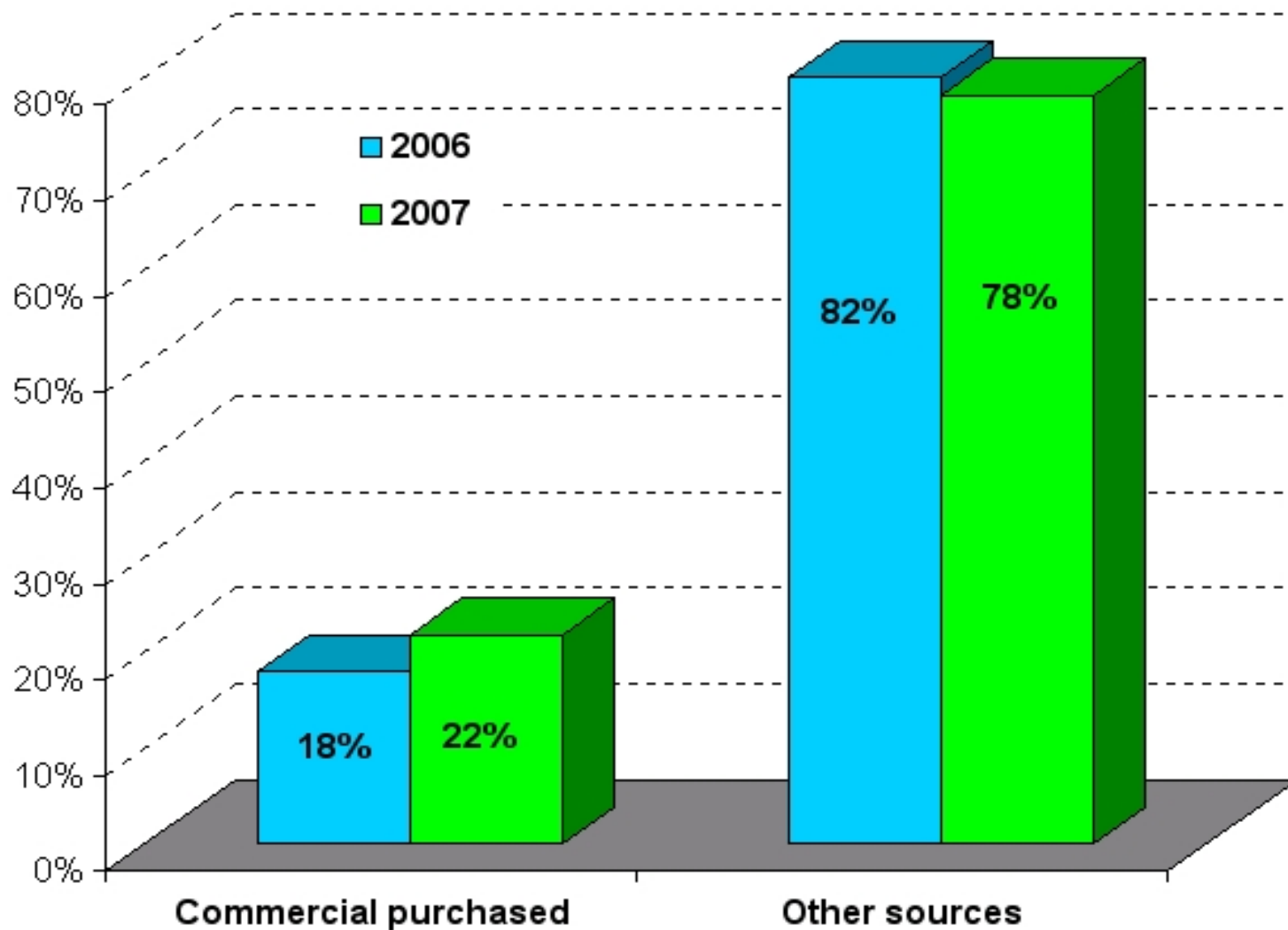
- Fiabilité et performances de Linux => validation dans le monde du serveur
- Disponibilité du code source
- Support matériel généralisé ou presque
- Pas de royalties
- Réduction des coûts (prix des licences développeur)
- Orientation *stratégique* des leaders vers Linux: Wind River, Mentor Graphics (Android)
- Effet de masse, ou de mode !



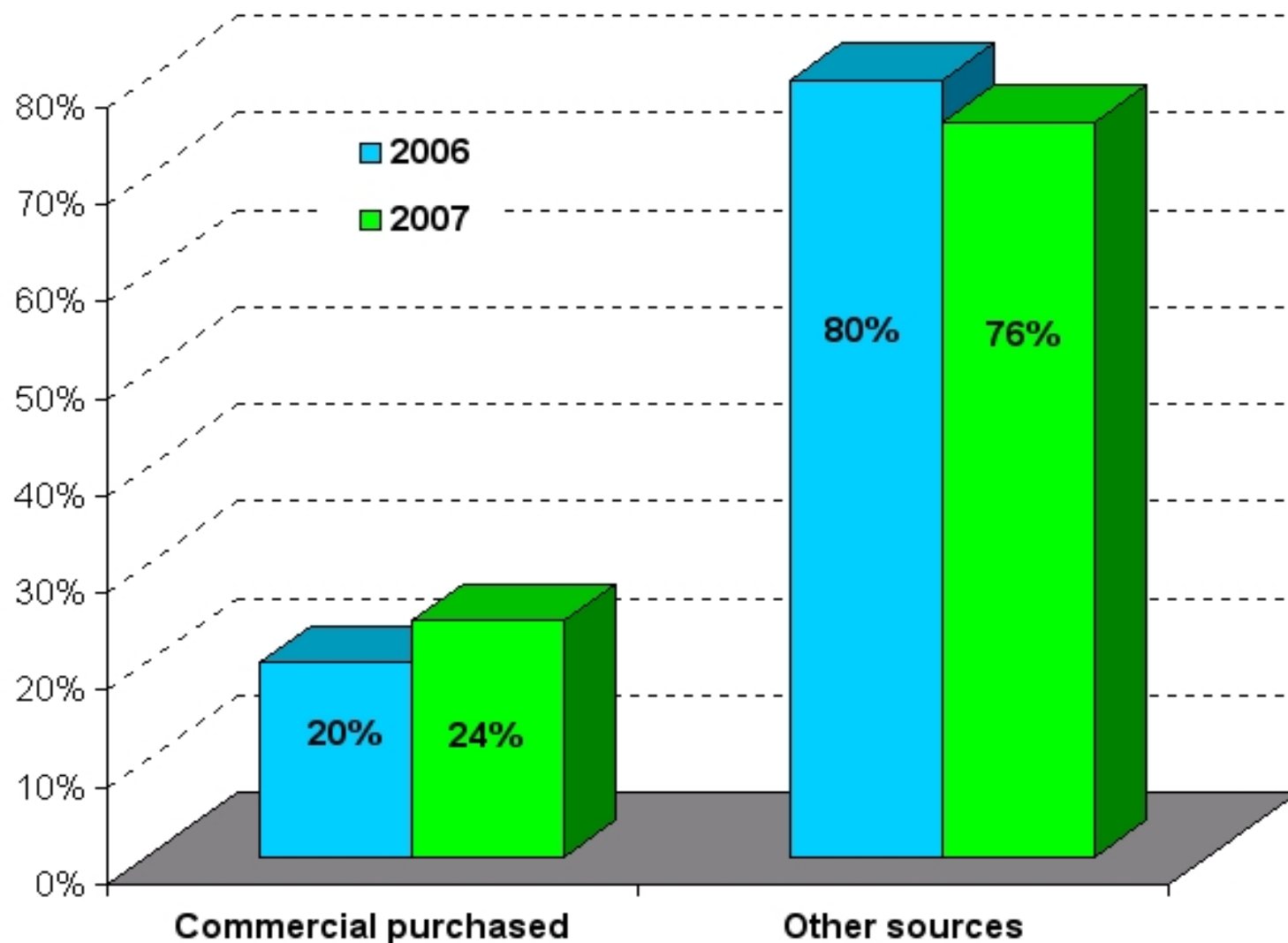
- Les promoteurs de Linux *ne se cachent plus*
- Linux est entré au cœur de la stratégie des grands groupes industriels
 - THALES
 - THOMSON/Technicolor (depuis 1997)
 - SAGEMCOM / SAGEM Défense
 - EADS
 - Opérateurs de Télécom
 - Orange (impose Linux à ses sous-traitants)
 - Free a toujours utilisé Linux
 - Les autres suivent (SFR, Bouygues)
 - Éditeurs de logiciels embarqués classiques (Wind River)

- Évolution du marché des télécoms
- Convergence téléphonie/Internet
- Linux arrive au niveau des terminaux (Android) et concurrence les OS spécialisés comme Symbian.
- Linux est omniprésent au niveau des routeurs grand public, de gré ou de force !
- Les projets libres communautaires (ex: OpenWrt) deviennent viables pour les industriels.
- Des solutions techniques «temps réel» sont disponibles sur Linux (RTLinux, RTAI, Xenomai)
- Meilleure confiance des décideurs

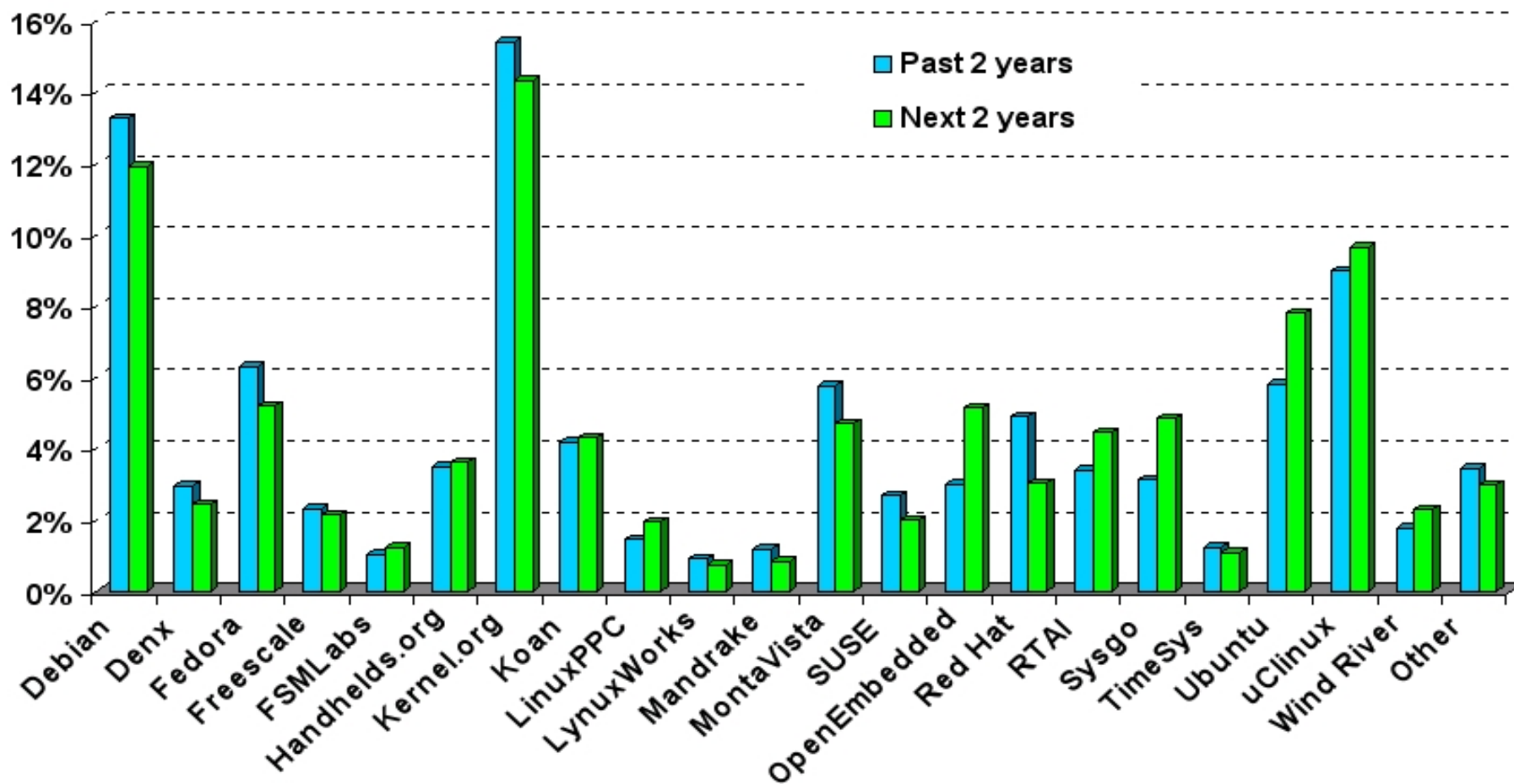
Embedded Linux OS Sourcing Trend - Current Use



Embedded Linux OS Sourcing Trend - Future Plans



Embedded Linux OS sourcing trends



KISS 1600 (Kiss Technology)

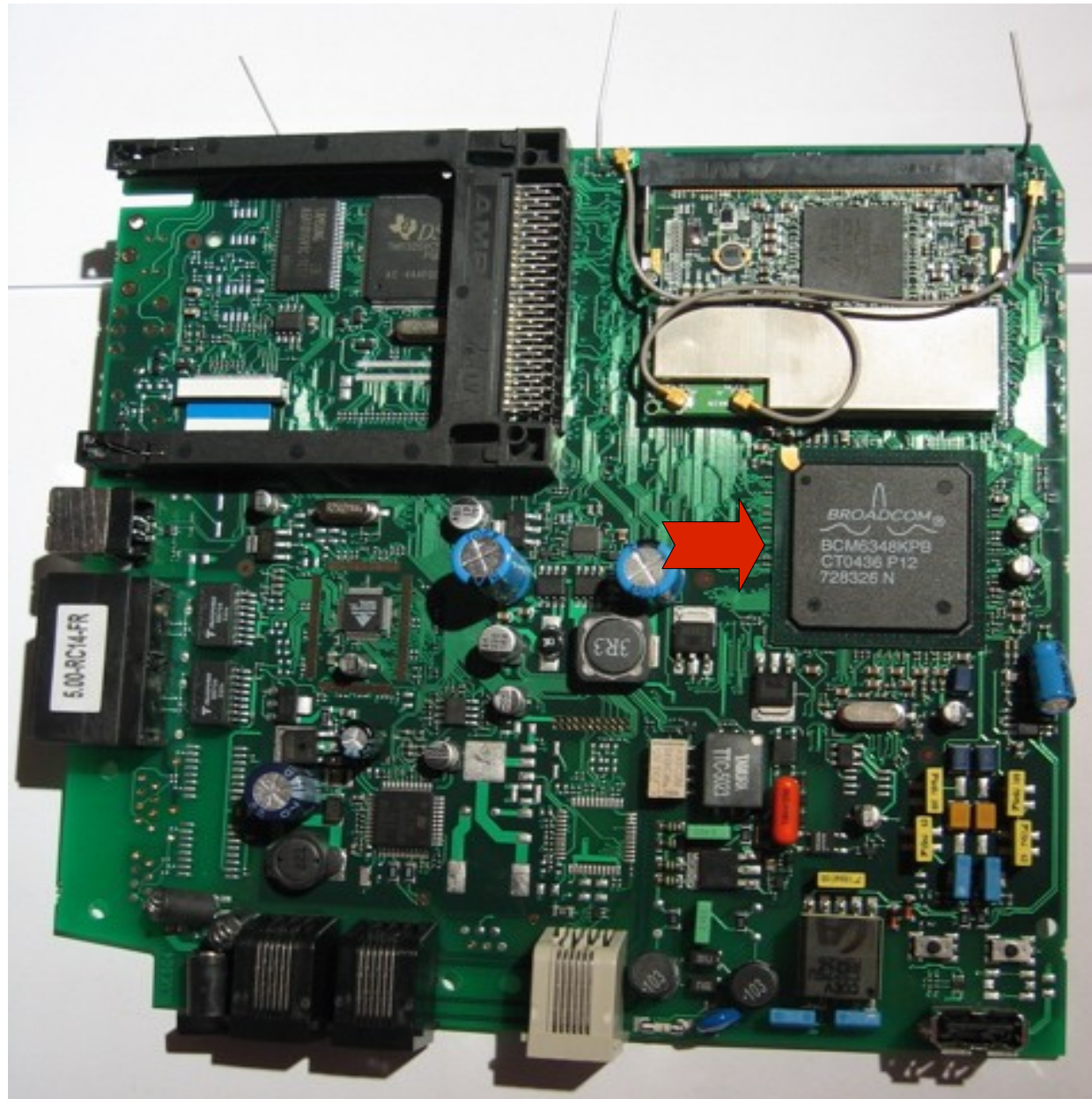














- Réservé aux systèmes « complexes »
 - Empreinte mémoire importante: 2 Mo pour μ CLinux, 4 Mo pour Linux
 - Consommation mémoire vive : 16 Mo minimum
 - Alternatives libres: eCOS, RTEMS beaucoup plus légers
 - Systèmes micro-contrôleur genre Arduino
- Migration temps réel des anciens RTOS un peu complexe => évolution avec les extensions PREEMPT-RT et Xenomai
- Certification difficile (voire impossible) de part le modèle décentralisé

- Les problèmes de licences GPL: non-respect, contraintes (ex: Free.fr, Broadcom, Dlink, etc.)
 - Pourtant les règles sont (assez) simples et équitables!
 - <http://gpl-violations.org>
- Les problème de brevets logiciels (patents)
- Manque d'outils
- Manque de documentation
- Encore trop peu d'ingénieurs spécialisés en *bas niveau*, le recrutement est très difficile
- Cependant, de plus en plus de formations spécialisées ou de modules dédiées