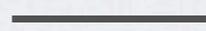


# LINUX EMBARQUÉ



# LES DISTRIBUTIONS

Carrefour de l'électronique — 24 octobre 2012



## Creative Commons

### Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0)

**Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

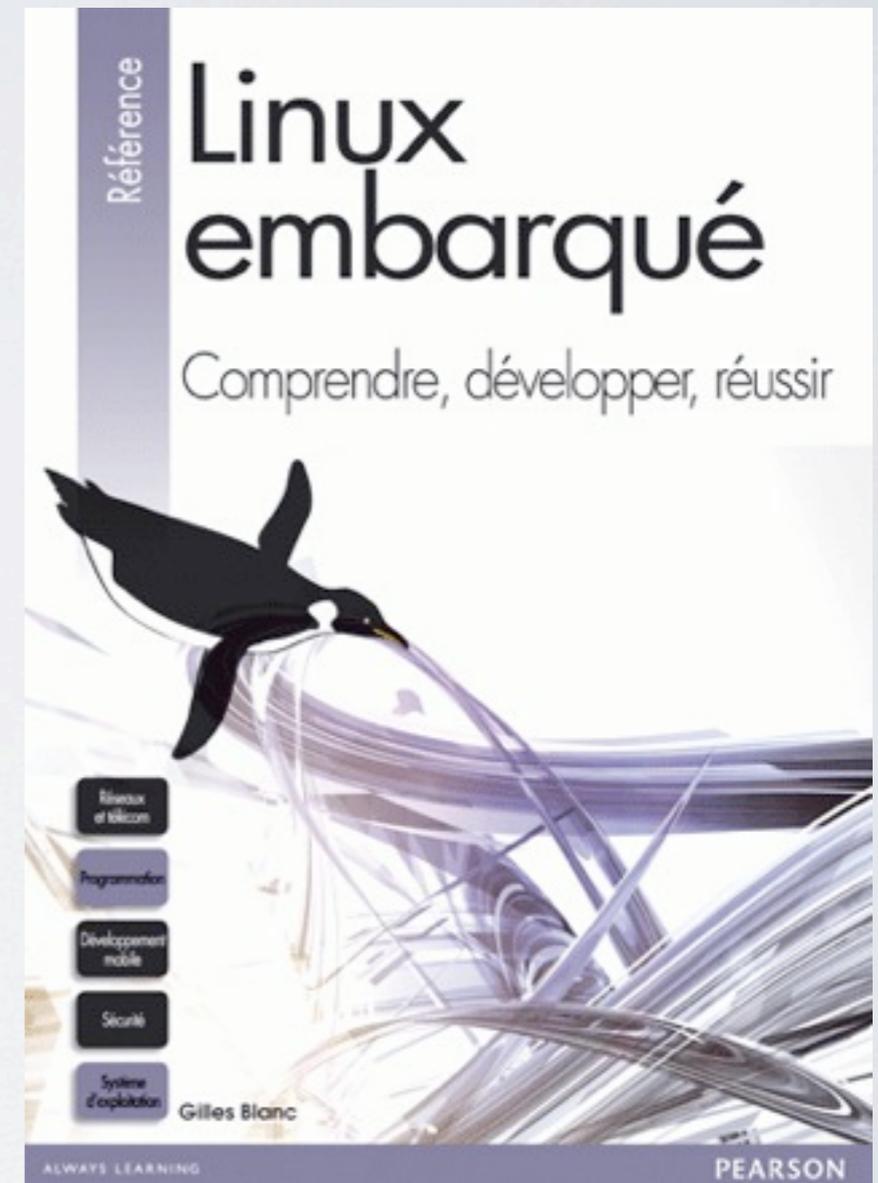
**Noncommercial** — You may not use this work for commercial purposes.

**No Derivative Works** — You may not alter, transform, or build upon this work.

Le texte intégral de la licence applicable à ce document se trouve à l'adresse :  
<http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>

# LE LIVRE

- *Linux embarqué*  
*comprendre, développer, réussir*  
(éd. Pearson, déc 2011, 460 pp.)
- management (économie, juridique, stratégie, etc.)
- technique
- au-delà : plaidoyer



# LINACS

## EN QUELQUES MOTS

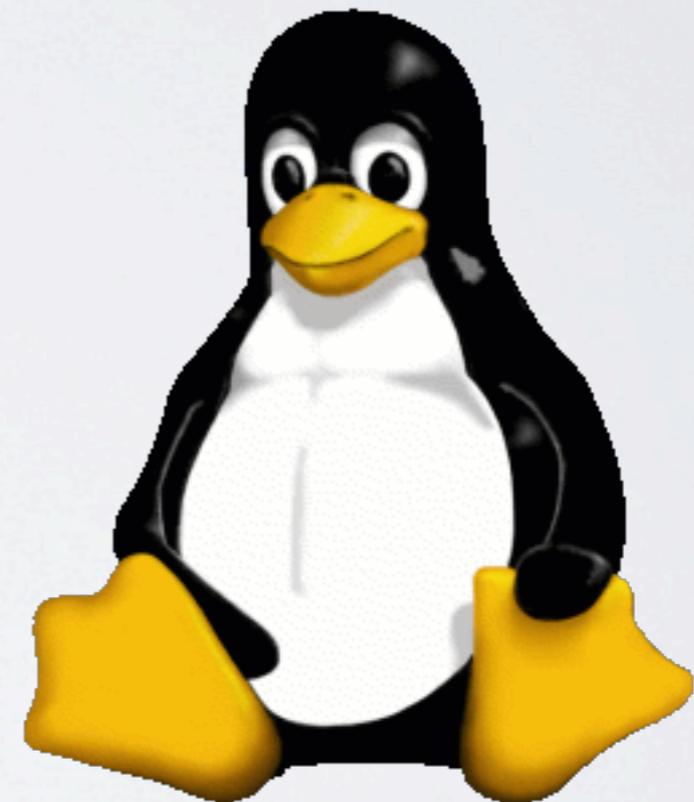
- Linacs = *Linux project accelerator*
- Un constat : les projets de logiciel embarqué à base de Linux sont devenus très complexes et échouent trop souvent
- Une idée : aider les entreprises à appréhender leurs projets, plutôt que de les réaliser pour elles
- Travail en open innovation, sur-mesure, en étroite collaboration
  - Préparation de projet
  - Assistance à Maîtrise d'Ouvrage
  - Suivi de projet
  - Audit
  - Formation
  - Recrutement



# LINUX EMBARQUÉ : QUOI ET POURQUOI

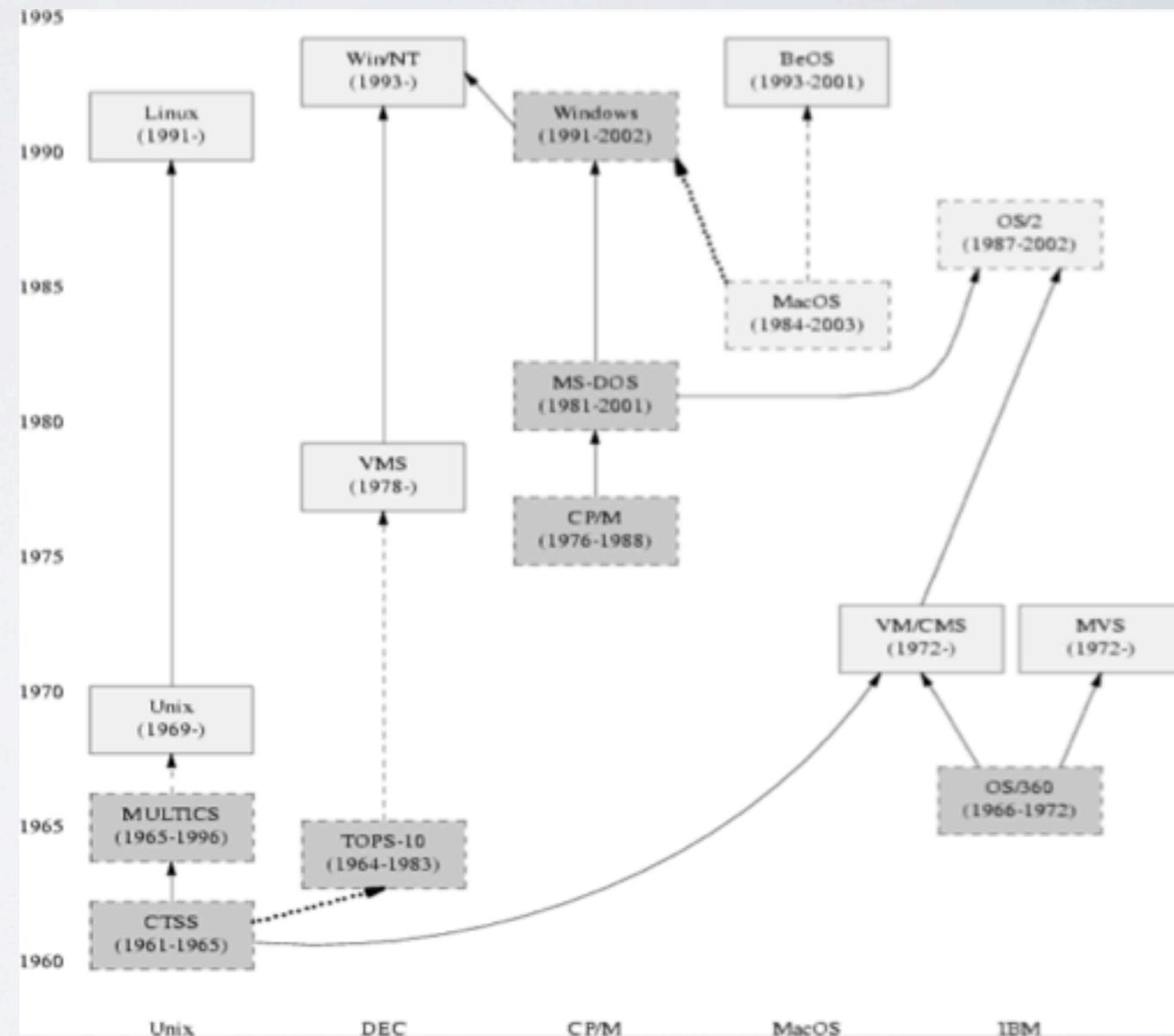
# QU'EST-CE QUE LINUX ?

- un **noyau**, un **système d'exploitation**
- à destination du PC, puis d'autres architectures
- un projet communautaire
- un système **libre**
- un héritage **\*NIX**



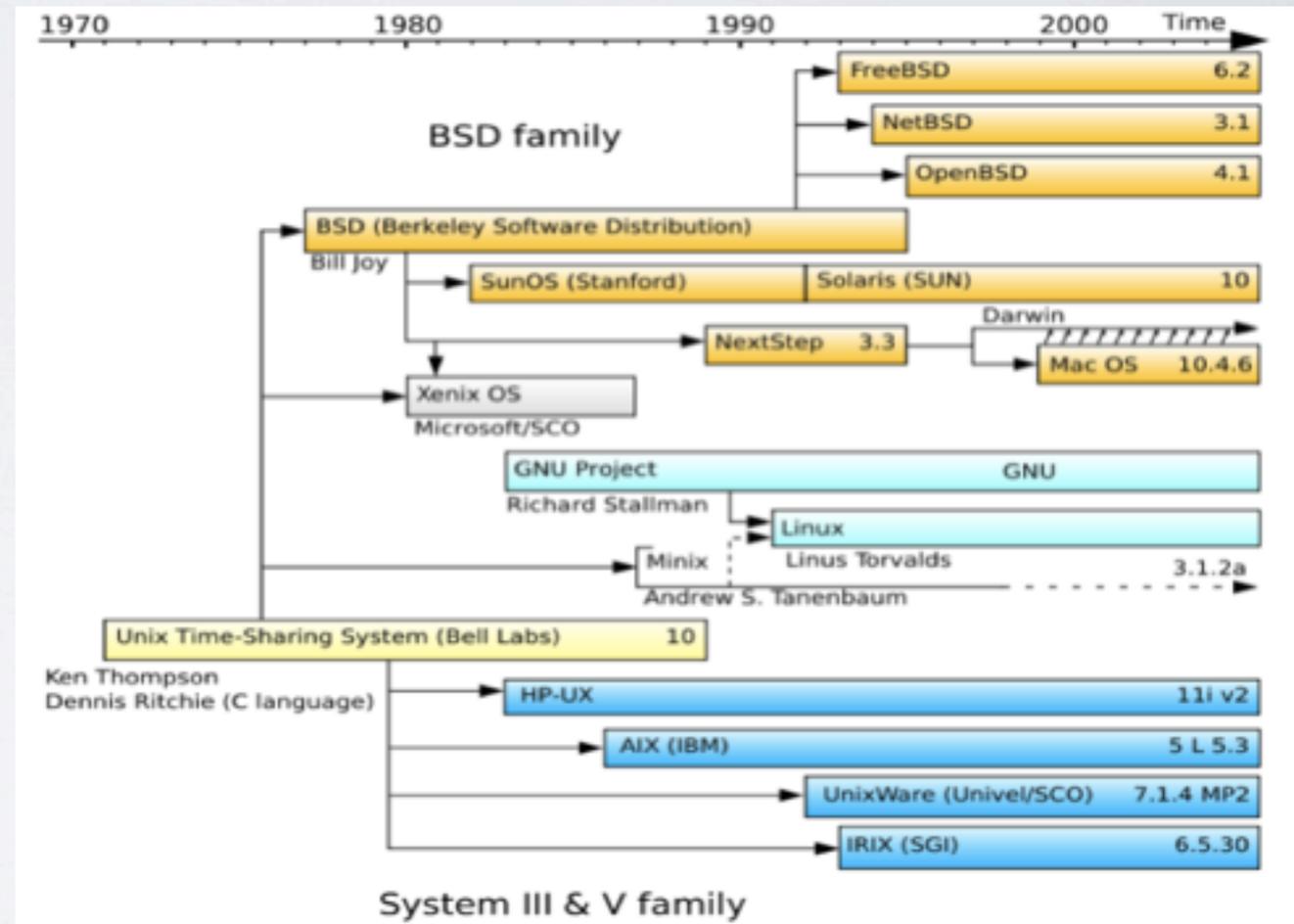
# HISTORIQUE

- « 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](http://en.wikipedia.org/wiki/History_of_Linux)



# SYSTÈME D'EXPLOITATION

- héritage des UNIX
  - tout est fichier
  - arborescence : /, /dev, /etc, ...
  - noyau séparé du système
- distributions
- GNU/Linux : noyau Linux + système GNU



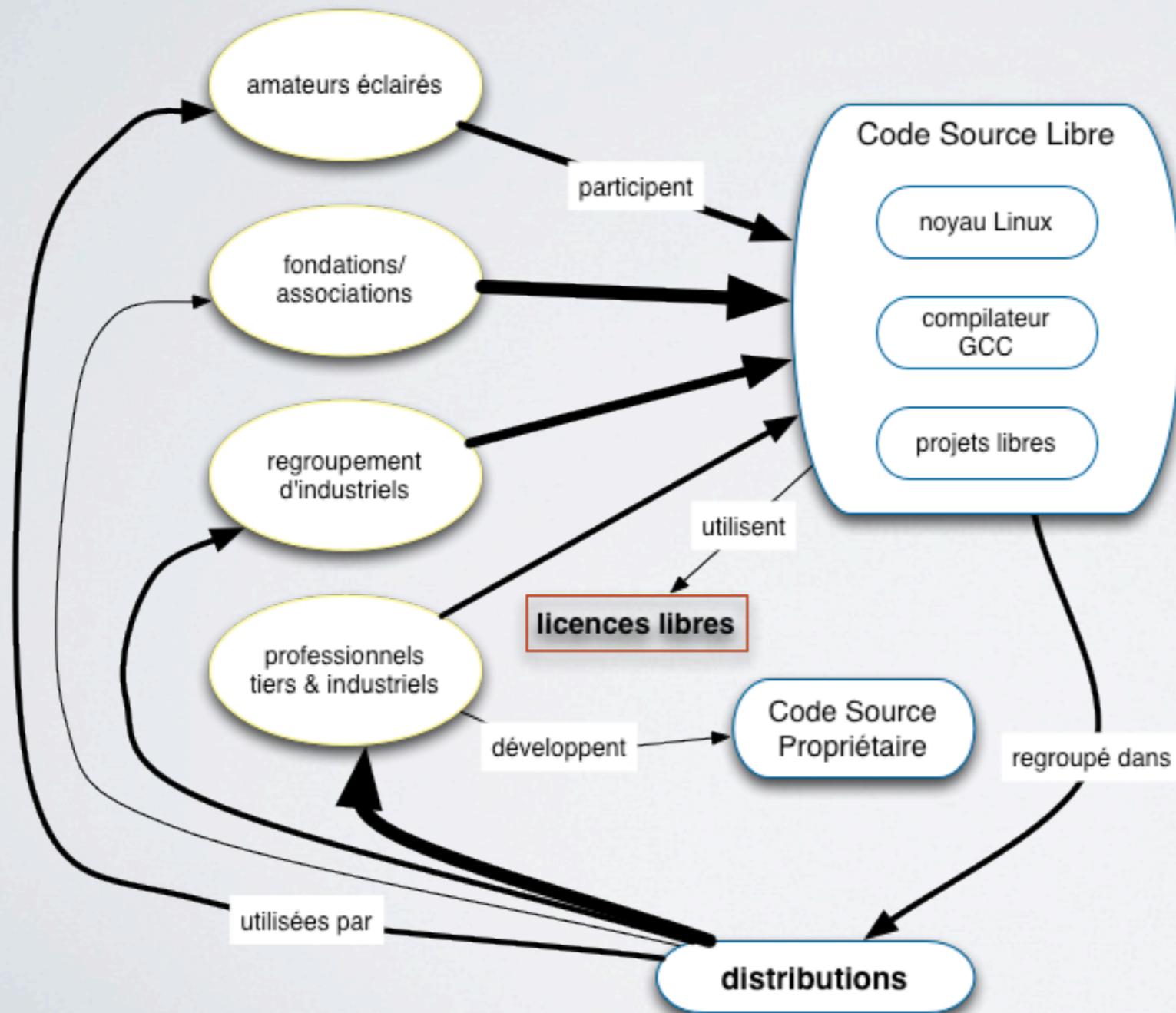
# LE LOGICIEL LIBRE

- une philosophie
- origine : le projet GNU (<http://gnu.org>)
  - 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 : licence GPLv2
- 4 libertés :
  0. la liberté d'exécuter le programme, pour tous les usages,
  1. la liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins,
  2. la liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies),
  3. la liberté d'améliorer le programme et de distribuer ces améliorations au public, pour en faire profiter toute la communauté.





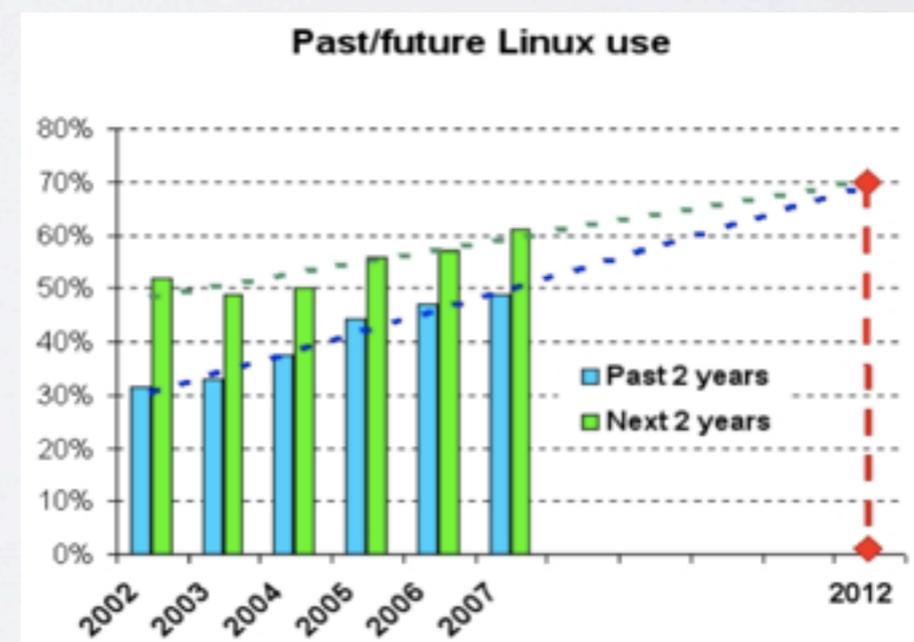
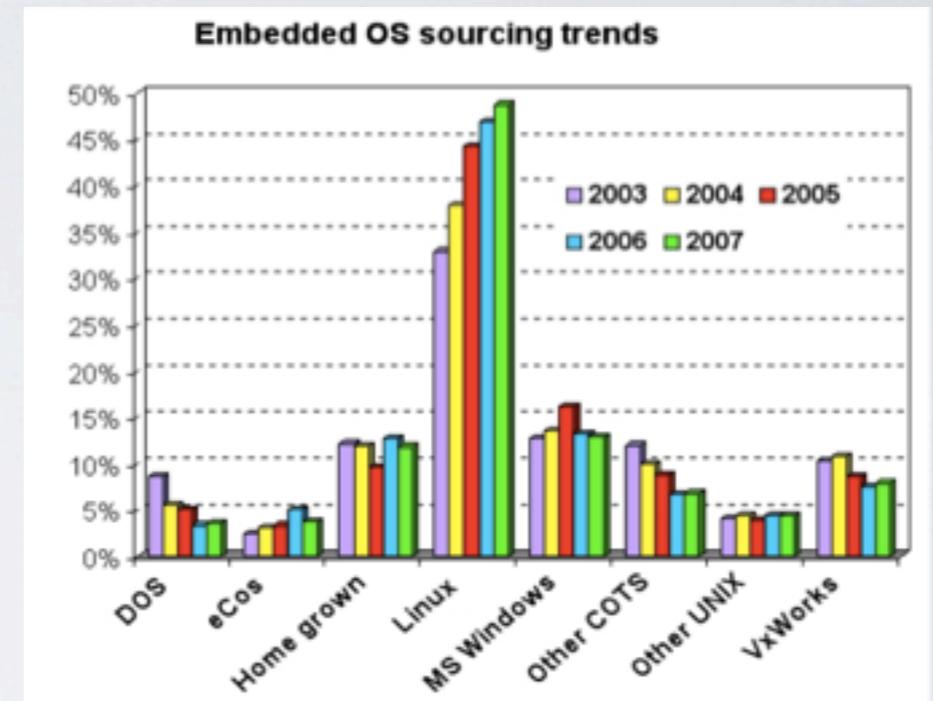
# ÉCOSYSTÈME LIBRE EMBARQUÉ



- écosystème complexe
- mêle du libre et du propriétaire
- ceux qui participent le plus ne sont pas ceux qui utilisent le plus

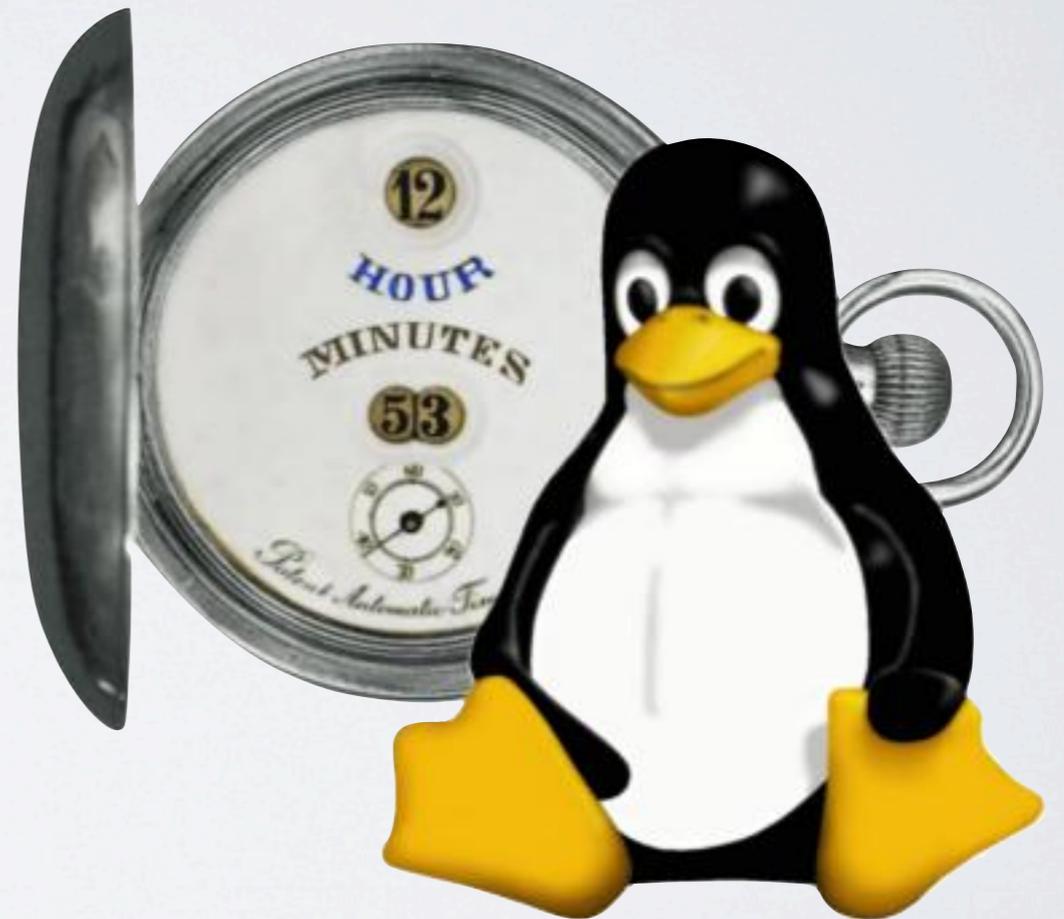
# LINUX EMBARQUÉ

- depuis ~2000
- éclosion réelle avec les version 2.4 (2004)
- explosion avec la version 2.6 (2004/2005)
- tournant en 2007
- acteur incontournable à présent (1 projet sur 2 !)



# EMBARQUÉ & TEMPS RÉEL

- Définition du TR : déterminisme temporel
- Des mondes qui se recoupent
- TR « mou » vs « dur » : Linux TR mou par défaut
- Solutions TR dur :
  - Linux, par modification du code : **RTLinux, MontaVista, RTAI, Concurrent Computer**
  - Par paravirtualisation : **Xenomai/ADEOS**, Wind River Hypervisor, PikeOS (Sysgo), Integrity (GreenHills), ...



# LIBRE EMBARQUÉ

- au-delà de Linux :  
un mouvement
  - bootloader : u-boot, grub, ...
  - compilateur : GCC, LLVM
  - outils divers de développement/intégration : Eclipse, GPS, SVN, Git...
  - logiciels
  - polices de caractères
  - environnements complets : GPE, Android, MeeGo, Tizen, ...
  - OS embarqués : eCos, RTMS, FreeRTOS, Pok, ...
- au-delà du libre :  
les formats ouverts
  - communication : XMPP, UPnP
  - image : png/mng, svg
  - vidéo/son : ogg, WebM
  - compression : gzip, bzip2, LZMA, 7z
- matériel libre ! (OpenHardware)
- Certification !  
(OpenDO, Opees, Babylone)

# POURQUOI LE LIBRE DANS L'EMBARQUÉ ?

- Nombreux avantages pour l'industriel :

- technicité (à la Torvalds)
- pérennité/éthique (à la RMS)
- couts (question entrepreneuriale)

- amélioration continue
- GPL/LGPL : amélioration « forcée »
- qualité du code
- interactions/documentation

- image positive (développement durable)
- aucun contrôle absolu, garantie d'existence sur le long terme

- mais libre n'est pas gratuit !

- pas de royalties (problème majeur dans l'embarqué)
- sources gratuites

→ couts cachés : couts de réalisation, couts de formation, couts de migration, etc.

# INCONVÉNIENTS DU LIBRE DANS L'EMBARQUÉ

- manque d'interlocuteurs identifiés : problème technique (assistance) et contractuel (responsabilité)
- manque de contrôle : dépendance à la communauté
- contraintes juridiques (propriété intellectuelle) : comprendre les licences libres
- problème de compétences : expertise et expérience rares
- difficulté d'estimations en temps et en couts (cf. effet IKEA !)
- beaucoup d'aléas à estimer
- de nouveaux paradigmes à maîtriser pour une industrie à forte inertie

# POURQUOI LINUX EMBARQUÉ ?

- Libre : GPLv2 → forte évolutivité, modifiable/adaptable
  - cible/marché très étendu :  
du très embarqué au Consumer Electronic
  - temps réel : « mou » ou « dur »
- écosystème existant : GNU/Linux
- disponibilité de développeurs formés
- nombreux points forts techniques

# POINTS FORTS DE GNU/ LINUX POUR L'EMBARQUÉ

- mérites du système
  - faible empreinte mémoire par rapport aux capacités (notamment pour les drivers)
  - efficace, optimisé, fortes performances
  - très stable : uptime
  - espace utilisateur (userland) efficace

- réalisation
  - beaucoup d'outils disponibles, de bibliothèques (briques élémentaires très fonctionnelles)
  - support de nombreuses technologies, natives ou sous VM, notamment Java (et HTML)
  - développement userfriendly (hôte ~ cible)

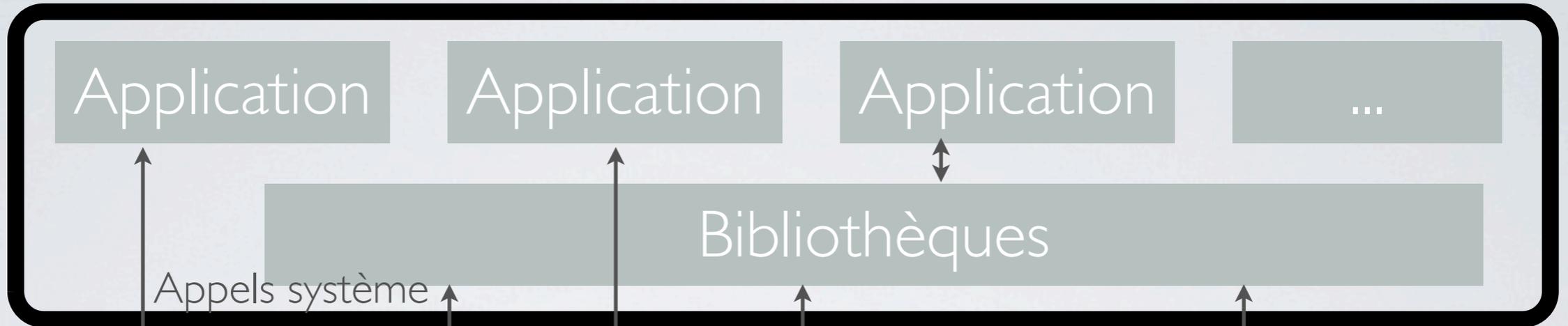
- architecture interne du noyau
  - modulaire, paramétrable très finement
  - énorme support matériel
  - portabilité : ~75 architectures, 32 bits, MMU ou non
  - TCP/IP (serveurs !), bluetooth, multimédia, ... : de base
  - ordonnanceur de complexité  $O(1)$  puis  $O(\log n)$ , préemptible (à partir de 2.6)



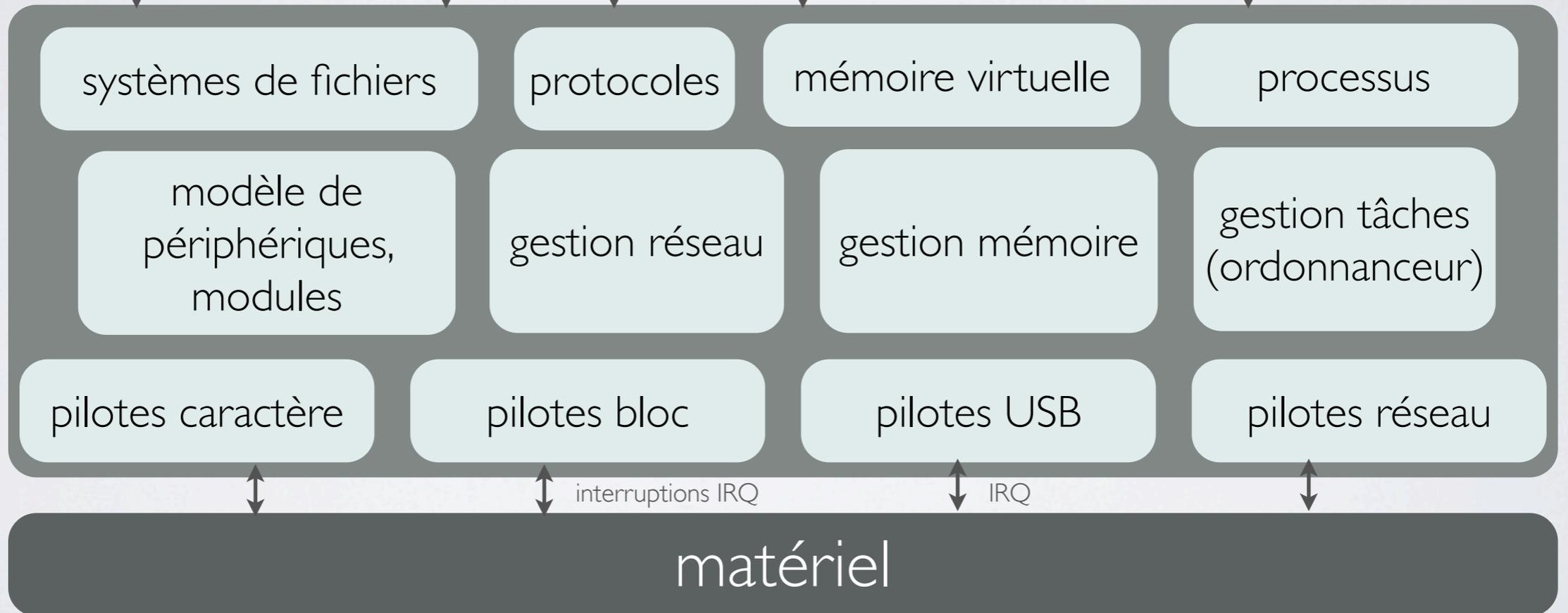
QU'EST-CE QUE LINUX EMBARQUÉ ?

# NOYAU

espace  
utilisateur

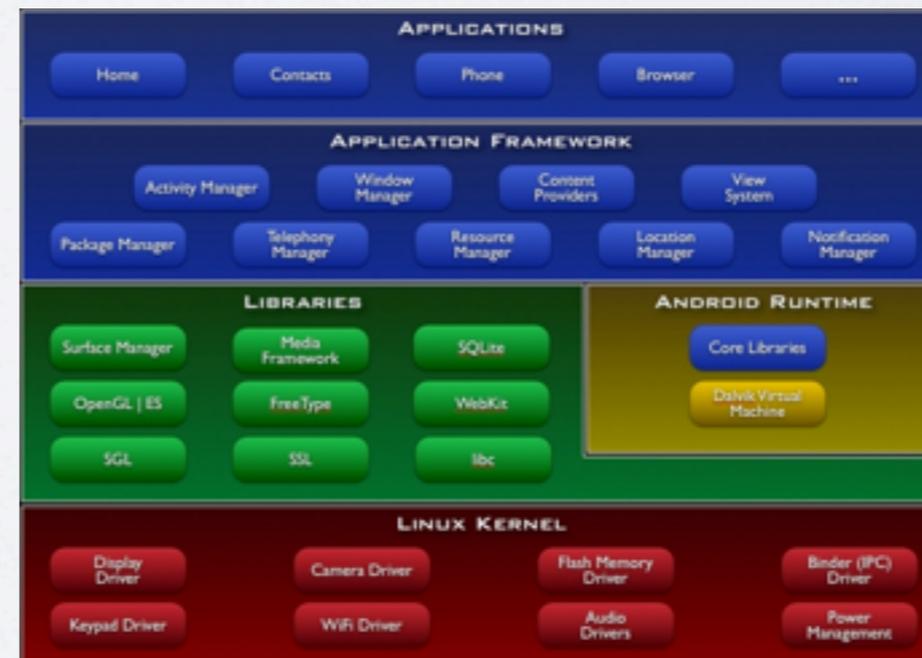
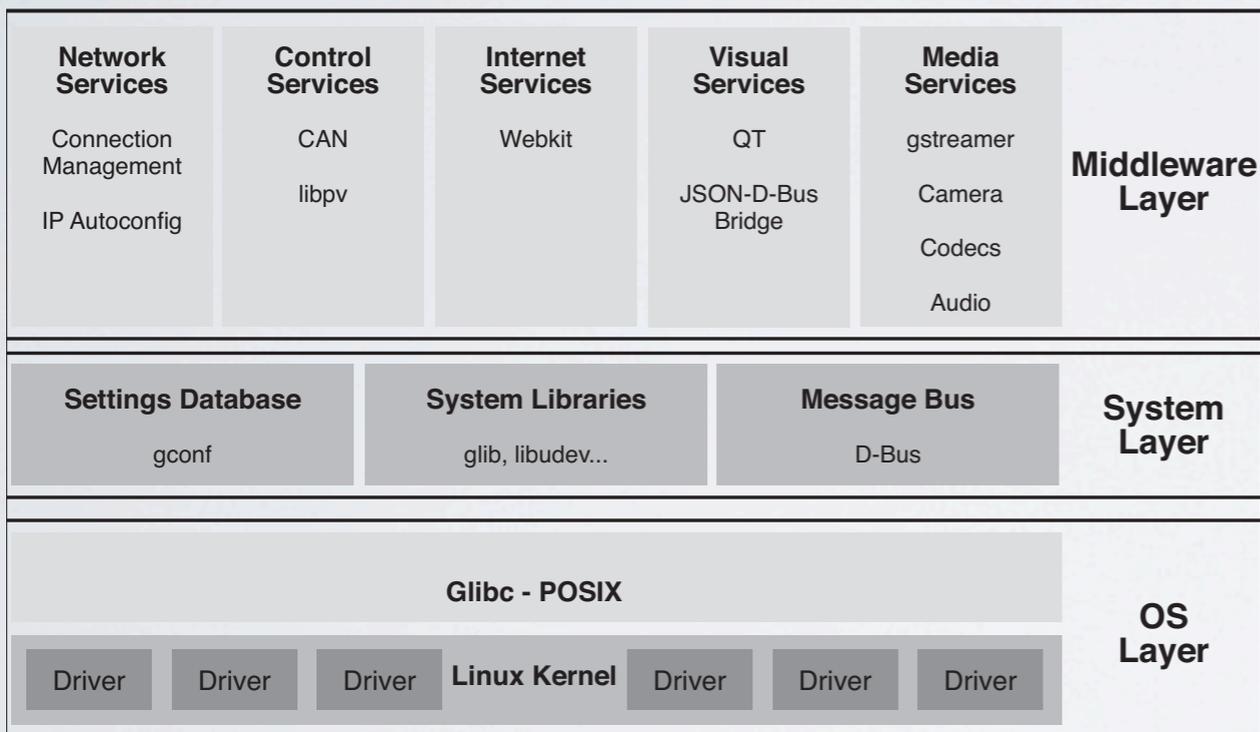
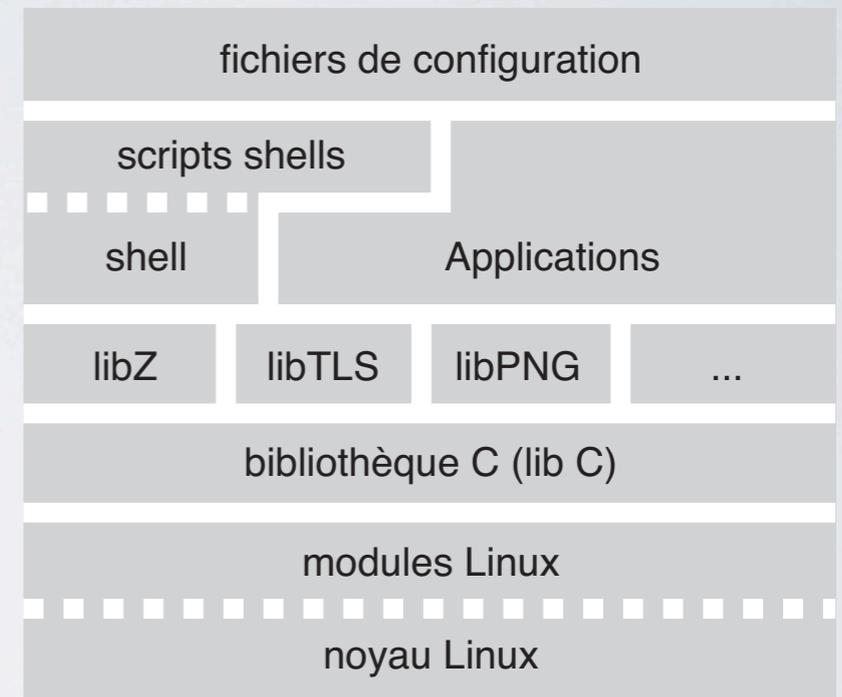


noyau Linux



# ORGANISATION LOGIQUE DU SYSTÈME GNU/LINUX

Système Linux  
embarqué « standard »



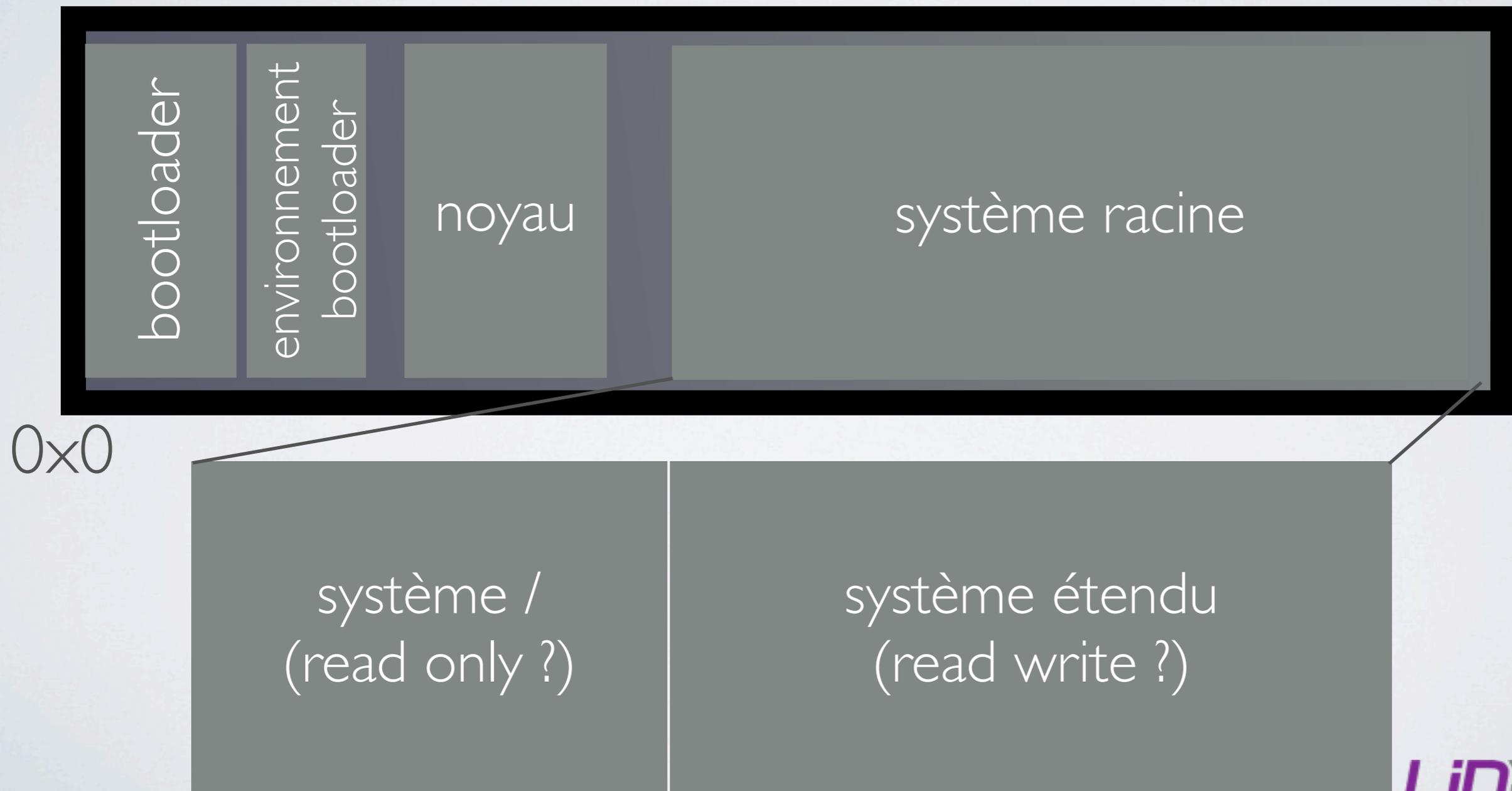
Android

Vue d'un système Linux embarqué industriel par Pengutronix  
([http://www.pengutronix.de/development/web/index\\_en.html](http://www.pengutronix.de/development/web/index_en.html)).

# LES DISTRIBUTIONS

- Qu'est-ce ?
  - « Collection » de logiciels et de configuration selon une certaine organisation
  - constitue le système (« racine », « rootfs »)
  - base de la création du firmware
- Pourquoi ?
  - dépendances applications/bibliothèques, bibliothèques/bibliothèques, parfois applications/applications
  - notion de « paquets » : application et/ou bibliothèques et (potentiellement) fichiers de configuration
  - mise à jour = évolution du système
  - mises à jour de sécurité !
- Construction du système (système de compilation / construction de paquets)
- Patches... Rapidement ingérable « à la main »

# ORGANISATION PHYSIQUE DU SYSTÈME SUR FLASH



# DEUX MODÈLES

- *communautaire*
  - projets libres, animés par une communauté (amateurs éclairés et professionnels, parfois fondations ad-hoc)
  - **OpenEmbedded, Ångström, Buildroot, OpenWRT**, etc.
  - **Linaro** : à travers une fondation chapeauté par ARM
  - **Android** (cas particulier : Google) ; **Tizen** (Linux Foundation), **FirefoxOS** (Mozilla Foundation)
  - autres distrib' Linux « classiques » réadaptées : **emdebian, Ubuntu Core**, etc.      *Attention : Distributions binaires*
- *commercial*
  - sur étagère (COTS), adaptable (service vendu)
  - **WindRiver, Sysgo, MontaVista**, etc.
- articulations
  - commercial s'appuie quasiment toujours sur du libre communautaire
  - relations commercial/communautaire (participation, intégration, etc.)
  - groupements d'industriels : OHA, Genivi, Babylone

# DISTRIBUTIONS SOURCES vs BINAIRES

- Binaires
  - rapide, tout est déjà compilé (croisé)
  - idéal pour les POC
  - mais manque de maîtrise
    - top-down
    - ajout de paquets seulement « à chaud » (lorsque le système fonctionne sur la cible)
  - concerne : les distributions « classiques » réadaptées (Ubuntu, etc.)
  - > 80 Mo, souvent ~250 Mo
- Sources
  - lent à mettre en place, tout doit être compilé, après une phase de configuration
  - pour la mise en production : maîtrise totale de la génération
    - bottom-up
    - ajout des paquets « à froid » sur l'image du firmware à intégrer (toujours possibilité de le faire « à chaud »)
  - Buildroot, OE-Core, OpenWRT, T2, OpenBricks
  - à partir de 2 Mo
- Hybrides
  - très bon compromis : distributions sources pré-compilées
  - permet la rapidité du binaire et la flexibilité des sources
  - point de départ du développement de sa propre configuration de distribution sources
  - par exemple : Angström, OpenWRT
  - de 2,5 Mo à 250 Mo

# ENVIRONNEMENTS PRÊTS À L'EMPLOI

- ▶ idée : simplifier la création d'un système Linux embarqué
- ▶ télécharge les sources à jour, les compile (croisé)
- ▶ gère les problématiques de compatibilité, de licences, de création de firmware (limité), etc.
- ▶ le projet **Linaro** pour ARM : compilateur, debugger, compatibilité noyau (gestion de puissance), etc.

## • **Buildroot**

- fichiers Makefile en arborescence
- rapide et efficace mais limité par la simplification
- peut coincer (implique alors de toute refaire sous OE)

## • **OpenEmbedded / OE-Core**

- arborescence plus complexe
- assez long à mettre en place mais très puissant
- pérenne sur le long terme

## • **OpenWRT**

- spécialisé pour les routeurs et appareils réseau
- utilise Buildroot-NG
- très grande quantité de paquets

# ENVIRONNEMENTS PRÉINTÉGRÉS/GRAPHIQUES

- But : base déjà existante, rajouter seulement sa valeur ajoutée
- **Ångström** : distribution complète, modulable par génération en ligne (Narcissus), très utilisée pour les POC
- **Android** : le grand gagnant, mais environnement complet tout Java (sur noyau Linux) incompatible avec une distribution Linux « standard », liant pour l'avenir (apparition cependant de systèmes hybrides)
- **MeeGo** : base Linux standard, écosystème cohérent, porté par de grands industriels avant d'être abandonné brutalement au profit de Tizen, rebooté dans un nouveau projet (**Mer**)
- **Tizen** : peu de recul encore, v2.0 alpha vient de sortir ; utilise HTML5 et EFL
- Gnome Mobile (**Maemo, OpenMoko, Sugar**) : solutions plus ou moins mourantes
- **GPE/OPIE** (handelds) : anciennes solutions abandonnées mais toujours utilisées sur de petits projets
- **Yocto (/Poky)** : géré par la Linux Foundation, OE-Core
- **Qt for embedded & QML** : bibliothèque graphique, très dynamique, mais pas vraiment un environnement

# QUESTIONS ?

Gilles BLANC — LINACS

<http://linacs-consulting.com>

[gilles.blanc@linacs-consulting.com](mailto:gilles.blanc@linacs-consulting.com)

+33 (0)6 70 33 00 03

