

Systemes d'exploitation pour l'embarqué

Didier DONSEZ

Université Joseph Fourier

IMA –IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`,

`Didier.Donsez@ieee.org`

Plusieurs définitions car plusieurs points de vue

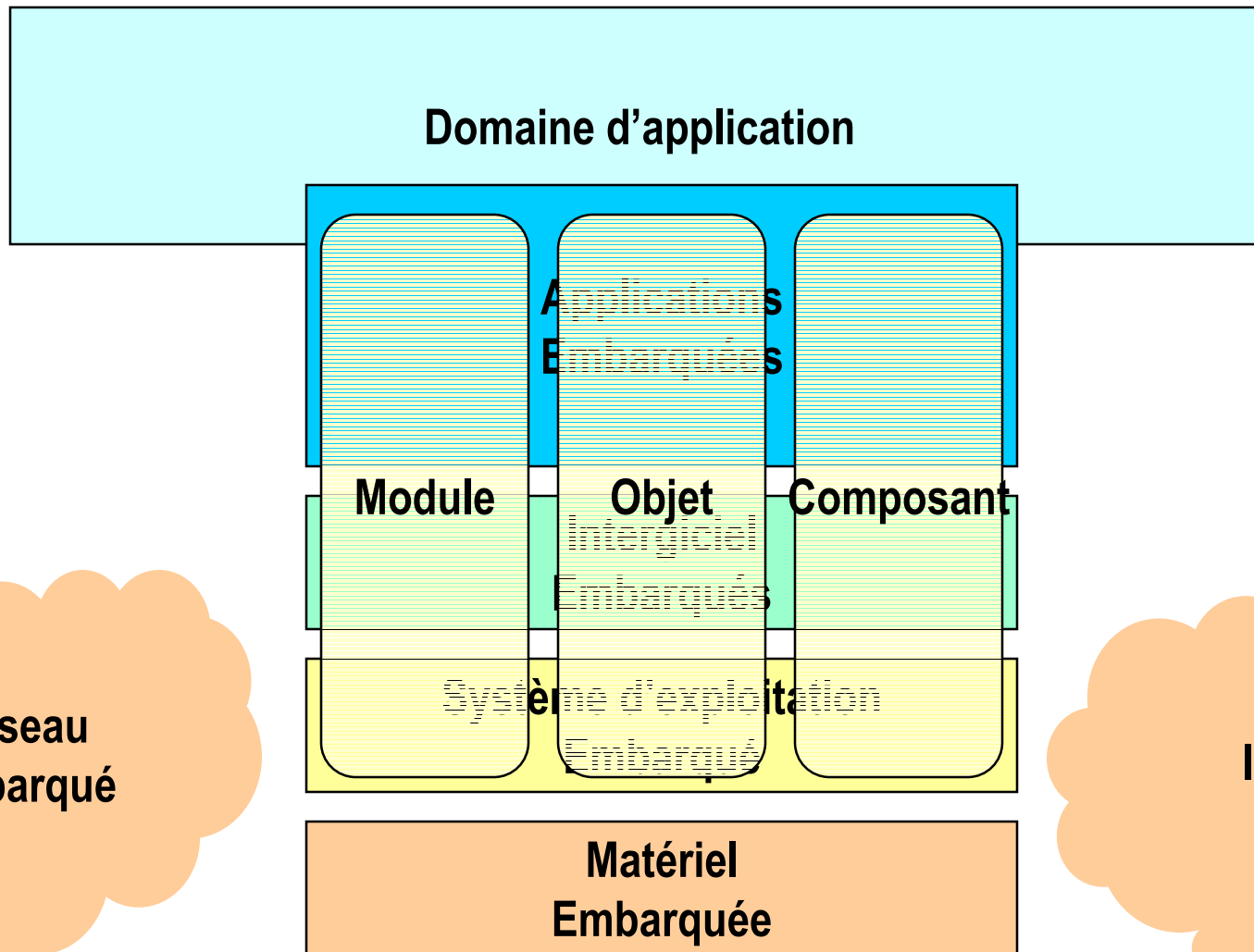
■ Matériel

- System on Chip (SOC)
- Contraintes d'environnement physiques
 - Température, humidité ambiante, choc/vibration, bruit, rayonnement γ
 - ...
- ...

■ Logiciel

- Certification/ Validation formel
- Temps Critique / Temps Réel
- Contraintes d'environnement d'exécution
- ...

Plusieurs points de vue



Quels chiffres

- 7 milliards de processeurs en fonction sur la planète en 2002
 - Une grande part sont des 8-bits
- 98% des μP sont dans des systèmes embarqués
- 30-70 μP ou μC dans une voiture moderne

- Les équipementiers qui fabriquent des appareils communicants
- Les opérateurs qui offrent un service

- Evolution vers de nouvelles fonctions
 - Nouveaux bus: CAN, USB, I2C,

« System On Chip »

■ "system on chip"

- pour une seule application pré-déterminée,
- plutôt sur une seule puce,
- en grande série,
- avec des interfaces spécifiques de l'application (par exemple radio dans le cas des objets portables communicants),
- à base d'un ou plusieurs processeurs spécialisés, ressemblant aux processeurs d'usage général, mais pouvant présenter des différences,
- comportant un logiciel sur mesure pour l'application visée, ce qui inclut, par exemple un système d'exploitation spécifique,

■ Exemple

- téléphones portables, consoles de jeux de poches, assistant personnels de poche, contrôleurs de TV satellite, ...

La gamme des systèmes embarqués

deeply
embedded

Besoin	Miniature	Petit	Moyen	Haut de gamme	PC embarqué	Embarqué haute disponibilité
Taille RAM	<0,1 Mo	0,1-4 Mo	2-8 Mo	8-32 Mo	16-64 Mo	> x Mo
Taille ROM/FLASH	0,1-0,5 Mo	0,5-2 Mo	2-4 Mo FLASH	4-16 Mo FLASH	Xx Mo	Go-To
Processeurs	DragonBall 68K Mcore ColdFire ARM	MIPS Hitachi SH x86 PowerPC				Pentium PowerPC
Caractéristiques matérielles	MMU optionnelle	Ardoise Internet Carte unité centrale System on Chip (SoC)				CompactPCI
Exemples d'applications	Caméra numérique PDA Téléphone	Routeur Décodeur Stockage en réseau Imprimante en réseau				Commutateur téléphonique Routeur haute performance Serveur central

Systeme embarqué faible coût

- Le prix est le facteur
- Systemes < 30 euros (hors écran)
 - SOC intégrant μ P μ C GPIO UART Flash
 - CAN, Ethernet
 - Pas de MMU
 - Mem < 1 MB
- L'équation à résoudre
 - Puissance calcul + Puissance dissipé + prix < Limite

Quelques définitions sur les logiciels embarqués (d'après Pierre Ficheux)

- Dédié ou intégré
- Ciblé
- Fiable et sécurisé
- Durée de vie longue
 - 20-30 ans pour le spatial ou militaire
- IHM spécifique (parfois réduite)
 - LCD, bouton poussoir, ...
- Optimisé
 - Petit taille ou contraint
 - Exemple: gagner 50Ko de RAM sur 1 million d'appareil
 - Hubble (386 durci)
 - Temps contraint
- Lien étroit entre développement logiciel
 - « SW becomes slower faster than HW becomes faster »

Logiciel embarqué vs système d'exploitation embarqué

- Système désigne maintenant SE (OS)
- SE parfois trop complexe pour certaines tâches très réduites
 - Ex: Firmware du lecteur MP3
- Dès que le nombre de taches/services devient important, le SE est rentable
 - Ajout de fonction communicantes (HTTP, DHCP, SNMP ...)
 - Ex: Stack IP dans modem ADSL
- Environnement de cross développement
 - quand la machine cible n'existe pas encore !
 - Ex: Développement des Jeux pour la Sony PS2 sur RedHat PC

Banalisation des performances

- Masque les imperfections
- Incite à une consommation effréné de HW
 - ce qui compense la baisse des coûts de celui ci !
- Donne des mauvaises habitudes aux programmeurs
- Masque le fonctionnement réel du système
 - on ne sait plus trop ce qui est vraiment utile
- Exemple: MS Windows + Office, Red Hat, ...

Domaines

- Ctrl processus industriel, machine outils
- Auto, transport, spatial
- Réseau télécoms: routeurs, PABX
- Périphériques informatiques: imprimante, ..
- Magnétoscope numérique, TV numérique
 - FreeBox (sous Linux)
- Equipement grand public jusque-là isolés
 - HiFi/TV, automobile, electro-ménager, ...
 - ScreenFridge ElectroLux
 - <http://www.electrolux.co.uk/screenfridge>
- Appliances

Systemes embarqués et Systemes Nomades

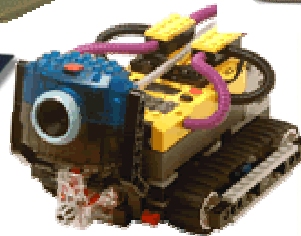
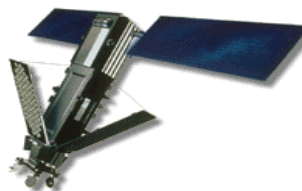
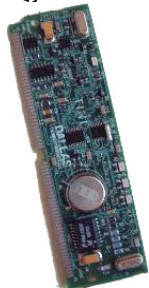
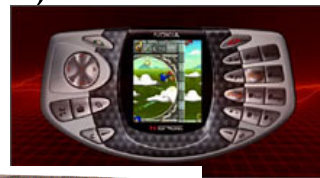
■ Embarqué

- Low Cost / product
- Remote Maintenance
- No UI
- Power consumption
- Physical conditions
 - Temperature, Humidity, ...
- Real Time (Hard & Soft)
- ...

■ Mobile

- Cost / product
- Power consumption
- Compact Size
- Consumer resistant
- Limited control & UI
- Memory capacity
- Limited communication bandwidth
- ...

pour l'embarqué



Typologie des Systèmes Embarqués

■ Individuels versus collectifs

- Téléphone, voiture
- Centrale nucléaire : Effet ressentis par une collectivité

■ Visibles versus Enfouis

- IHM du distributeur de boissons
- radar de parking d'une voiture

■ Non Temps Réel versus Temps Réel

- déterminisme
 - même cause -> même réponse dans des délais requis

■ Normal versus avec Sûreté de Fonctionnement

- SAV du vidéo-projecteur
- missile, centrale nucléaire
 - doit absolument fonctionner et le non fonctionnement serait préjudiciable

■ Facteur prix

- Importance du prix dans la décision d'achat

Tendance du marché dans les Systèmes embarqués

- Réduction du Time2Market
 - Idée-Conception de + en + court
- les équipes de R&D ne maîtrisent plus le spectre complet des technologies innovantes nécessaires
- le fabricant devient de + en + un intégrateur
 - (exemple de bien d'équipements: automobile, camera, ...)
- le métier et la technologie se sont plus intimement associé
 - (exemple: pour un fabricant de terminaux de paiement, SAV, ergonomie se sont ajoutés pour solution de paiement)
- Long cycle de vie avec gestion de l'obsolescence
 - compteur électrique EDF : 15 ans
 - porte avion : 30 ans
 - Disponibilité des composants électroniques au bout de 10 ans
- Tendance
 - Autonomie (intelligence locale pour le traitement et le filtrage)
 - Communicante (M2M: Machine to Machine)
 - Intègre dans un infrastructure (sécurité (virus, ...), ...)

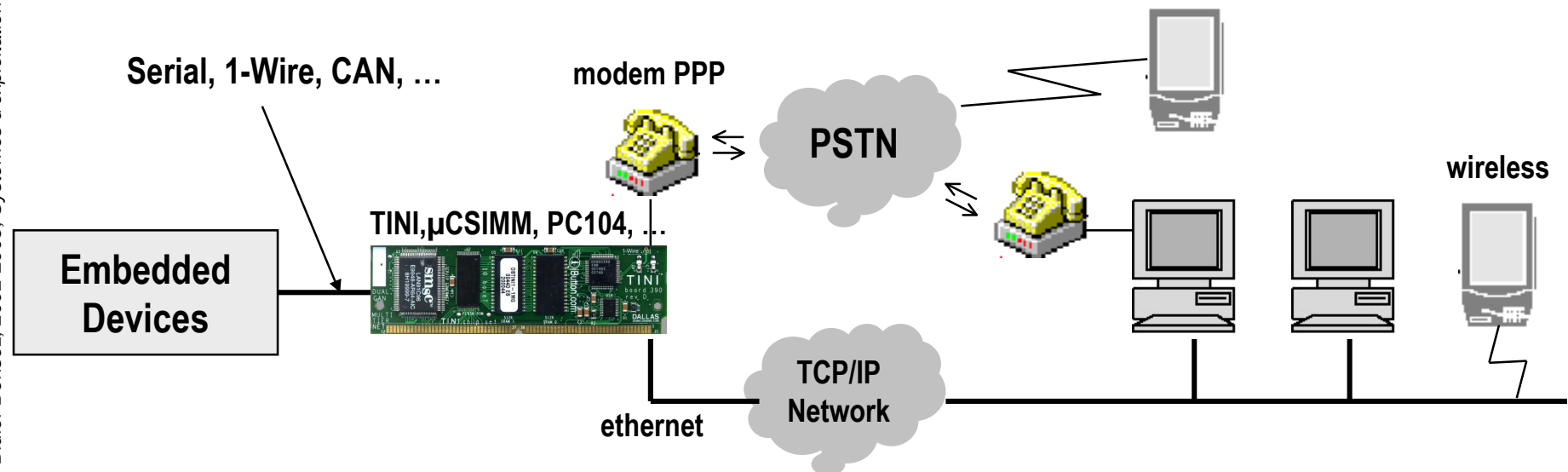
Un cas d'utilisation

■ Motivation : Coupleur Réseaux

- entre Ethernet et des réseaux IP
- et des périphériques sur des réseaux non IP
 - Capteur température, actionneur circuit électrique (CAN, ModBus, 1Wire, ...)
 - Enregistreur ECG, récepteur GPS, ... (RS232, USB, ...)

■ Applications : *Web based management*

- Mesure et Contrôle à partir de l'Internet (fil/sans fil)



Systemes d'exploitation pour l'embarqué

- Grande hétérogénéité
 - Très différent
- Mémoire et μ Processeurs limitées
 - Taille des pile et données
 - CPU pour calcul fonctionnel
- Capacité communication
 - QoS variable
 - Mode online, offline, dégradé, ...
- Energie
 - Power Aware Computing
- Contraintes Temps Réel / Temps Critique
 - Non TR, « soft » RT , « hard » RT

Plusieurs approches pour un OS pour l'embarqué

- *Plus d'une centaine d'OS embarqués*
- Noyau élagué
 - Dur d'élaguer dans le sac de nœud
 - Linux, RTLinux, μ CLinux ...
- OS Commercial
 - Mais cible un marché large donc fonctionnalités en trop
 - VxWork, pSOS, QNX ...
 - Jaluna 2.0 (μ kernel Chorus+ noyau Linux)
- OS Domaine
 - PalmOS, Symbian
- OS Custom
 - Coûteux en temps. Difficilement portable et maintenable
- OS modulaire et flexible
 - Think, eCos, ...
- *Middleware/Intergiciel*
 - Comme intermédiaire entre application et système
 - Trop gros mais nécessaire dans un système distribué

OS pour l'embarqué

■ Académique

- ExoKernel, SPIN, Think
- Choices, OSKit, Coyote, PURE, 2K

■ Commercial

- VxWorks, QNX, pSOS, Windows CE
- JavaOS, Jbed, MMLite, icWORKSHOP, Pebble

■ Open Source de qualité industrielle

- Linux, μ Clinux, eCos

■ Lire

- L.F Friedrich, J. Stankovic, M. Humphrey, M. Marley, J. Haskins, “A Survey of Configurable, Component-Based Operating Systems for Embedded Applications”, IEEE Micro, May-June 2001, pp54-68.

VxWorks AE



Wind River Systems (<http://www.windriver.com/>)

- Le leader du marché (pour l'instant)
- RTOS
 - Temps réel
 - Embarqué
- Cibles
 - Intel, PowerPC, ARM, MIPS

pSOS



- racheté par WindRiver, disparu depuis

QNX RTOS

<http://www.qnx.com>

Under Construction
En Construction

■ μ Kernel

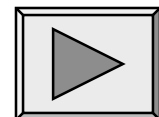
- multiprocessing, transparent distributed processing, fault-tolerant networking
 - real time scheduling
 - virtual memory protection (process)
 - Footprint : 12 KB
-
- free for non-commercial use (download @ <http://get.qnx.com>)

Windows embarqué

■ 2 gammes

- Windows NT embedded & Windows XP embedded
 - x86, >16MB RAM, Alimentation secteur
 - Cibles/Domaines: Guichets/Bornes automatiques (Billetique), Contrôle Process, ...
 - 1er exemple XPe: l'OS de la X-Box
- Windows CE.Net embedded
 - Intel X86, Intel ARM/Xscale, Hitachi SH3-4, Nec VR
 - >2MB RAM si XIP (execution in place), >4MB RAM sinon
 - Secteur/Batterie
 - Cibles/Domaines: PDA, SmartPhone, Domotique, Contrôle industriel (carte PC104), ...

■ Voir la présentation détaillée Windows CE



Linux Embarqué

■ Noyau Linux (standard ou des distributions)

- Configuré pour limiter les modules aux périphériques présents
- *From scratch* de la distribution standard (Linus Torvald)
 - Téléchargez <http://www.kernel.org> puis `make xconfig`
- Distribution commercial
 - Outil de création/ gestion de projet graphique, Vente du support technique (bug, drivers, ...)
 - Embedix, BlueCat, MontaVista, Red Hat
- Distribution libre
 - PeeWee Linux, Embedded Debian Project, ETLinux
- Distribution du Noyau *patché* temps réel
RT/Linux, RT/AI

■ Néanmoins adapté à des machines puissantes

- Trop consommateur en ressources
 - RAM >1MB, MMU, cache
 - PPC, MIPS, IA32, IA64
- Souvent trop de fonctionnalités (processus, MV, Fichiers ...)



■ Portage du noyau Linux

- Cibles: μP et μC 16-bits et 32-bits sans MMU
 - Motorola 683xx, Motorola ColdFire, Intel i960 d'Intel, ARM7TDMI, Altera NIOS, ...
- Empreinte mémoire
 - Noyau inférieure à 512 Ko.
 - Noyau + commandes Linux inférieures à 900 Ko.

■ Limitations d'usage

- Pas de mémoire virtuelle
 - Pas de espace protégé (overflow possible)
 - Pile fixé pour chaque processus (pas de sbk())
- Pas de fork() mais vfork()
 - L'appelant est bloqué du exec() jusqu'au exit()
- Image binaire d'un processus < 256Ko

■ Outils de développement

- GNU gcc, gdb, ...
- Portage de shells, serveur Web, ...
- Pile TCP/IP complète

eCos

Under Construction
En Construction

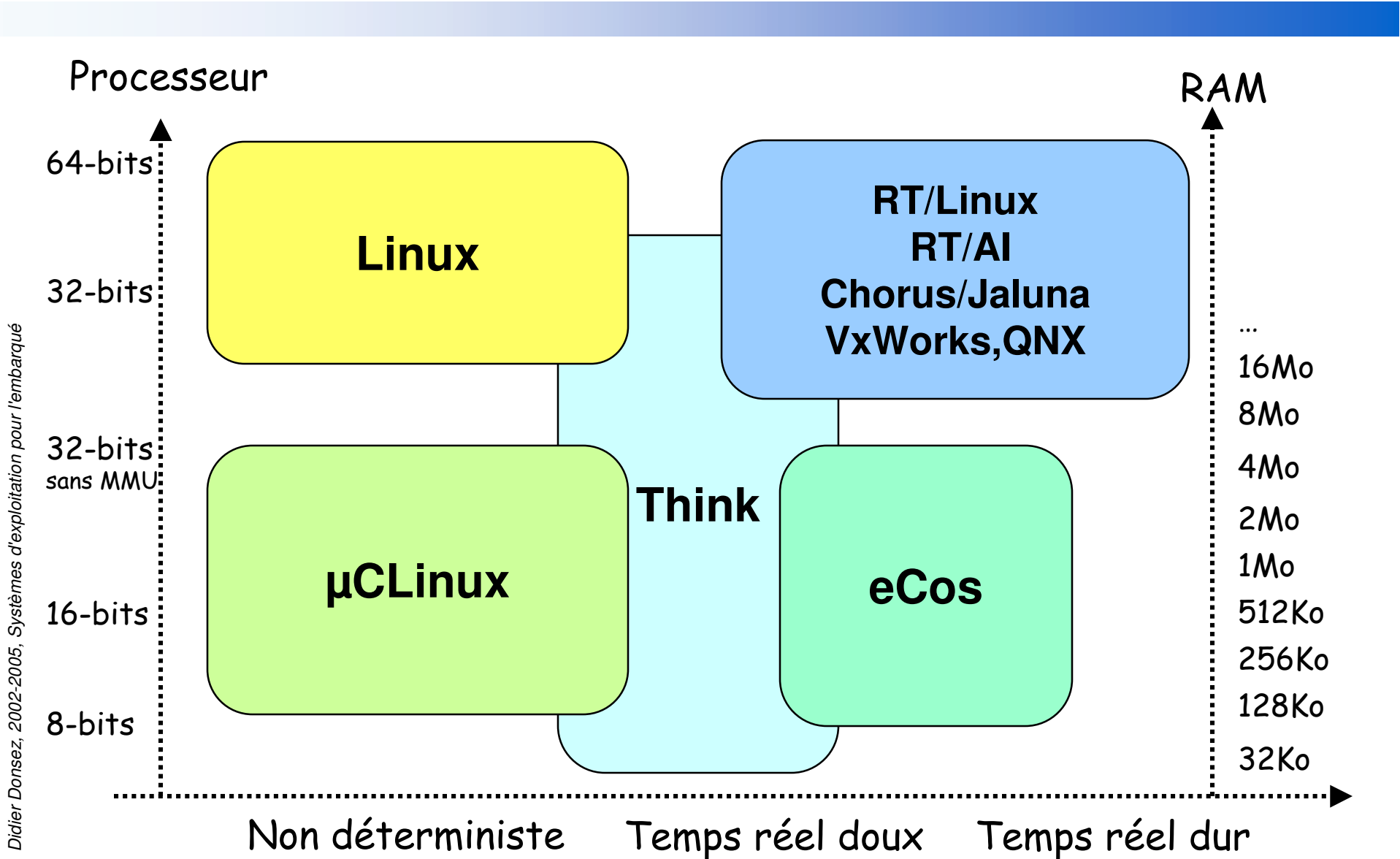
- Voir présentation eCos

Think



- Voir présentation Think

Comparaison OS / HW



Didier Dorsey, 2002-2005, Systèmes d'exploitation pour l'embarqué

Avantages et inconvénients des SE open source (d'après Pierre Ficheux)

Avantages

- Disponibilité du source
 - maintenabilité, augmentation de la qualité, pérennité de sources plutôt que de la société qui vend le binaire et détient le source
- Redistribution sans royalties
 - économie
- Développement dérivé de ce code source
 - économie et délai de développement
 - Ex: Décodeur JPEG, RSA, Pile IP, daemon HTTPD
- *Programmer familiarity*
- Grands Acteurs
 - des grands acteurs s'y mettent (IBM, Sony ...)

Inconvénients

- Méfiance des décideurs
 - Pas garantie contractuelle
 - Modèle décentralisé
 - Certification, ...
- Complexité et contraintes des licences
 - + de 100 licences
 - GPL, LGPL, BSD, Mozilla ...)
- Support technique
 - En France : OpenWind, MontaVista
- Jeunesse des sociétés open-source
 - stabilité financière, pb de la pérennité du service

Bibliographie

■ Programmation

- Michael Barr , « Programming Embedded Systems in C and C++ », O'Reilly, First Edition January 1999, ISBN: 1-56592-354-5, 191 pages

■ Linux

- Pierre Ficheux, Linux embarqué, Eyrolles, 2002, ISBN : 2-212-11024-3
 - Présente la problématique et propose de la pratique
 - Le rare livre en français
- Dr. Craig Hollabaugh, "Embedded Linux: Hardware, Software and Interfacing", Ed Addison Wesley Professional, ISBN: 0672322269
- John Lombardo, "Embedded Linux", Ed New Riders Publishing 07/2001, 0-7357-0998-X, <http://safari.informit.com/?XmlId=0-7357-0998-X>
- Abbott, "Linux for Embedded and Real-Time Applications", Ed Butterworth-Heinemann, (September 2002), ISBN: 0750675462

■ Windows CE

- Programming Microsoft Windows CE, Douglas Boling, Microsoft Press
- Inside Microsoft Windows CE, John Murray, Microsoft Press

Articles

- L.F Friedrich, J. Stankovic, M. Humphrey, M. Marley, J. Haskins, “A Survey of Configurable, Component-Based Operating Systems for Embedded Applications”, IEEE Micro, May-June 2001, pp54-68.

Webographie

■ Évaluation de RTOS

- <http://www.dedicated-systems.com/encyc/buyersguide/rtos/rtosmenu.htm>

■ Cours « Systèmes Embarqué » de Patrice KADIONIK

- <http://www.enseirb.fr/~kadionik>

Quelques eOS en détail

Windows CE

Jaluna

eCos

Think

<http://www-adele.imag.fr/~donsez/cours>

Windows CE

Didier DONSEZ

Université Joseph Fourier

IMA –IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`,

`Didier.Donsez@ieee.org`

Windows CE: Caractéristiques

■ Système d'exploitation

- Conçu pour les systèmes embarqués
 - PDAs, Téléphones, Consoles de Jeux fixe, Web Tablets, Téléviseur Interactif ...
- Architecture 32 bits, multitâche, multithreadé
- Portabilité WinCE-Win32
- Conception par composants

■ Architecture en couche

- Le noyau (kernel)
- La couche OAL (OEM adaptation layer)
- Le boot loader

Windows CE: Architecture système (1/3)

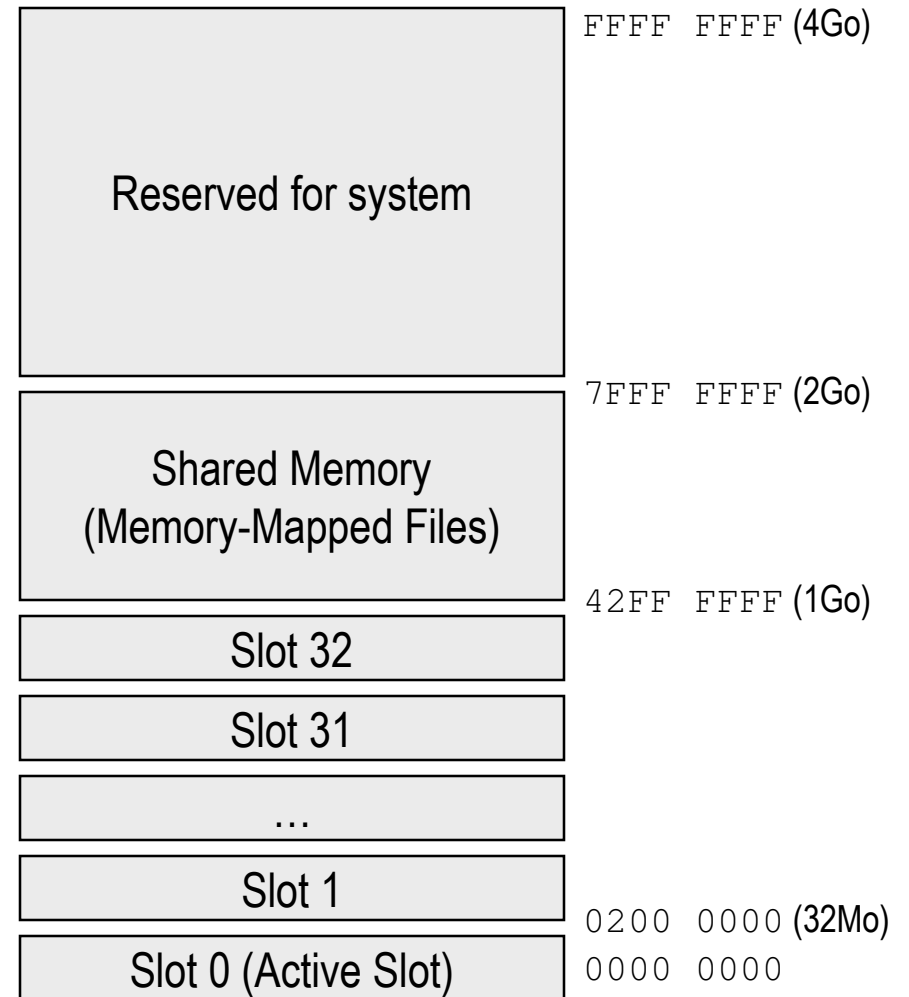
■ Gestion des processus

- Processus: définition
- Thread
 - Définition
 - Modes d'exécution: noyaux et utilisateurs
- Ordonnancement des threads, 256 niveaux, Round Robin
- Objets de synchronisation entre threads
 - Mutexe
 - Sémaphore
 - Évènement
 - Section critique

Windows CE: Architecture système (2/4)

■ Gestion de la mémoire

- Types
 - RAM
 - ROM
 - FLASH
- Mémoire Virtuelle
- Architecture



Windows CE: Architecture système (3/4)

- Gestion du système de fichier
 - Object Store en RAM stockage données (BD, registre, SGF)
 - Partition FAT
 - FSD Manager: support partitions autre que FAT
- Gestion des entrées sorties
 - Device Driver Model
 - Device Driver Interfaces

Windows CE: Architecture système (4/4)

■ Interface utilisateur: sous système GWES

- GDI structurée en Device Context
- Windows and Dialog Management
- Windows Messaging
- Entrée utilisateur:
 - IMM, Input Method Manager
 - IME, Input Method Editor

Webographie

- Collection d'Open Sources pour Windows CE
 - <http://openwince.sourceforge.net/>

<http://www-adele.imag.fr/~donsez/cours>

Jaluna

Didier DONSEZ

Université Joseph Fourier

IMA –IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`,

`Didier.Donsez@ieee.org`

CV Société

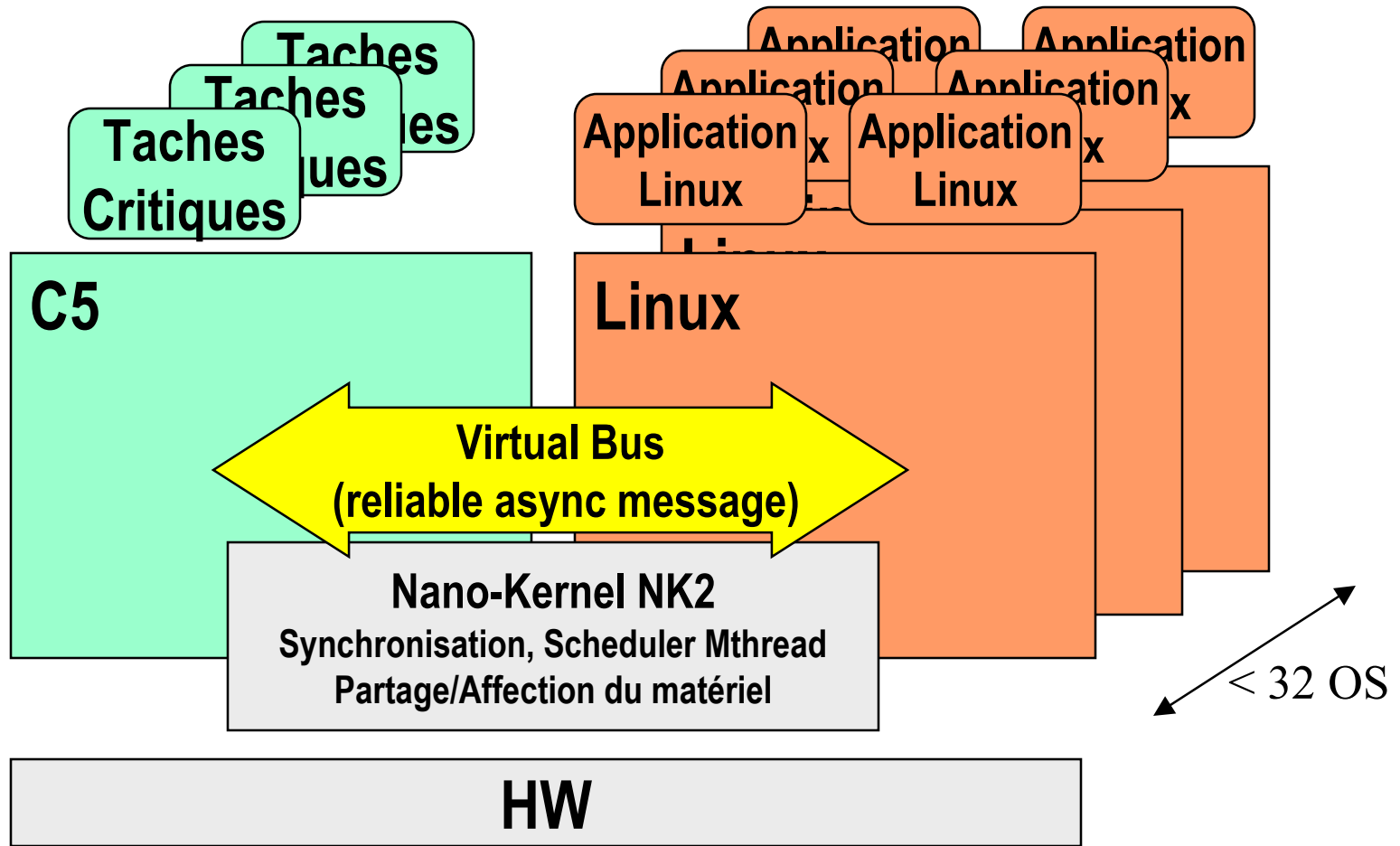
■ Jaluna

- Chorus racheté par Sun (1997) puis spinoff Jaluna
- 30 personnes

■ Clients

- Telecom, Fabricant de mobile (OneTouch Alcatel)
- Imprimantes et Routeurs

Architecture Jaluna



C5 (ex Chorus v5)

■ Features

- Hard RT
- Config
- IPv6
- Chorus IPC
- Isolation Mémoire
- Mode superviseur ou utilisateur des tâches
- Support J2ME/CLDC
- ...

■ Haute qualité

- Qualification OS SUN !

Produits Jaluna

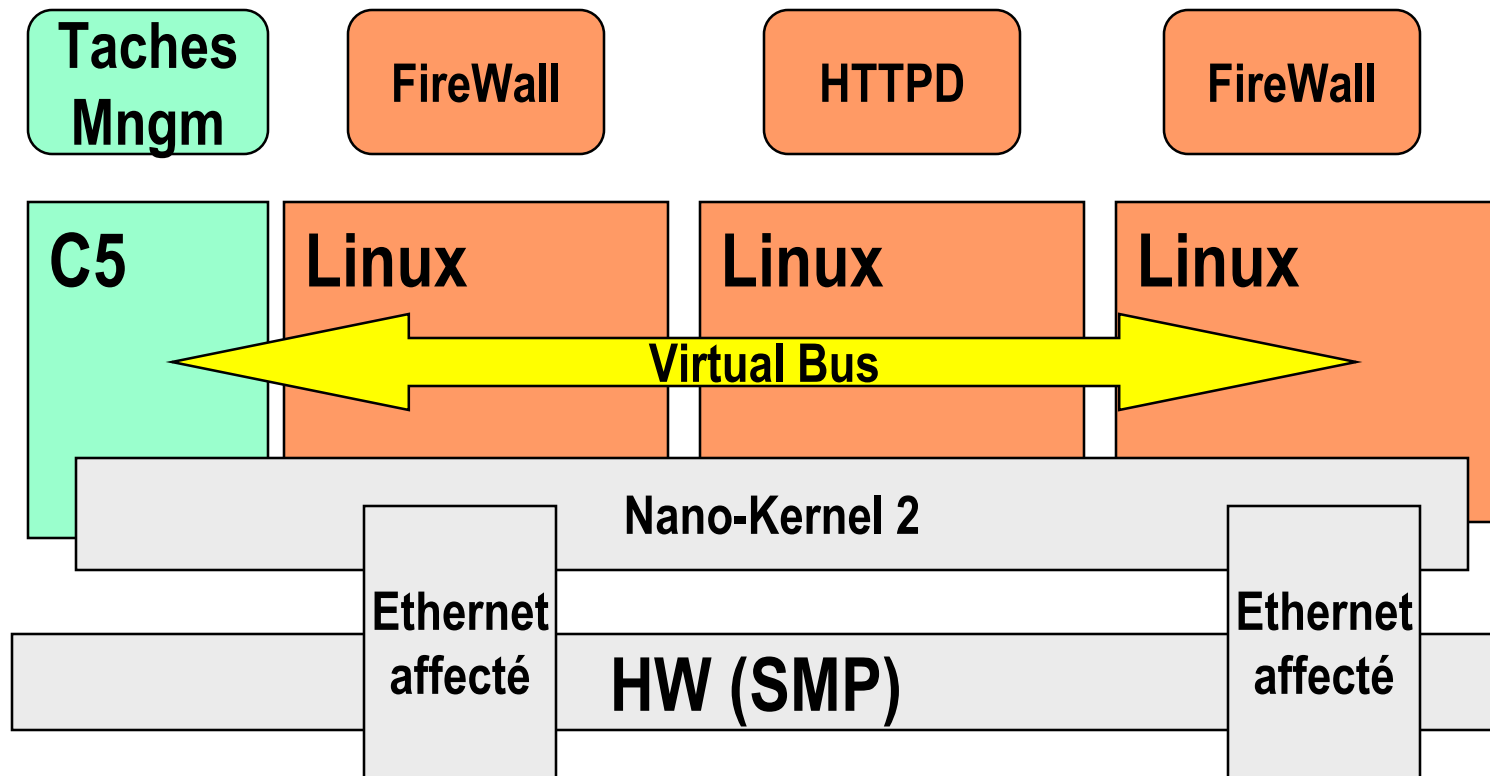
■ Cibles

- X86, PPC, SPARC, MIPS, ARM

■ Gamme

- Jaluna 1
 - C5 RT Posix
 - Empreinte: 2MB RAM et min 512 KB Flash pour C5 et plus pour applis
- Jaluna2/EL
 - Cible: STB, ...
 - C5
 - Standalone Embedded Linux
Soft Real Time (1 ms)
- Jaluna2/RT
 - RT C5
 - 1 Linux
- Jaluna2/VL
 - RT C5
 - Plusieurs OS
 - Cible: Super-Calculateur

Cas d'étude de DMZ



Virtual Bus

■ Messages

- Asynchrone, *reliable*, structurés ??
- Principe même des μ Kernels (MACH, Chorus, ...)

■ Point à Point (instance à instance)

■ 2 FIFO par pair d'instances

- jusqu'à $(32+1)^2/2$ FIFO ???

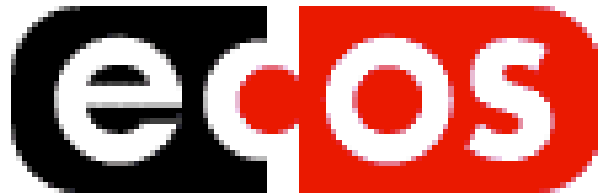
■ Taille de chaque FIFO est configurable

- dynamiquement ???

Démo Jaluna

- Partage de la mémoire sur un PIII 256 Mo
 - 2 instances Linux
 - C5
- Partage d'un port Ethernet
 - 3 adresses privées IP
 - Serveur NFS Linux
- WatchDog Linux
 - pour redémarrer une des 2 instances Linux en cas de crash
- Mémoire résistante au crash Linux
 - Protégée contre les crashes Linux

<http://www-adele.imag.fr/~donsez/cours>



Embedded Configurable Operating System

Didier DONSEZ

Université Joseph Fourier

IMA –IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`,

`Didier.Donsez@ieee.org`

eCos

Embedded Configurable Operating System

■ OS Open Source

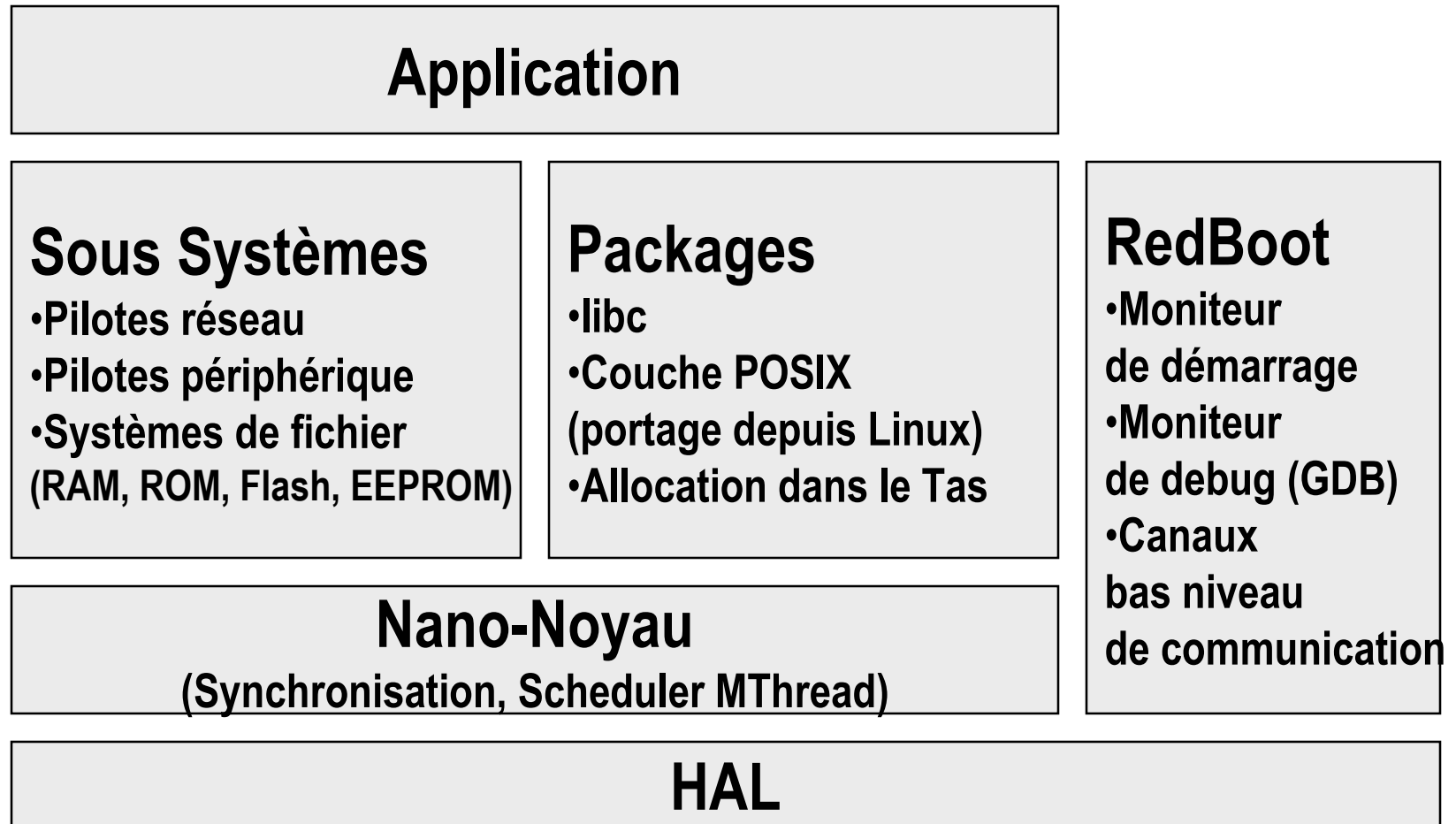
■ Ressources

- μ P 32 bits léger
 - ARM, M68K/ColdFire, Hitachi H8/300, Fujitsu FR-V, Intel x86, Matsushita AM3x, MIPS, NEC V8xx, PowerPC, SPARClite, SPARC, SuperH
- RAM <1 MB, sans MMU

■ Caractéristiques

- Nano-Noyau (Scheduler, Synchronisation)
- Temps Réel (Multithread préemptif)
- Extensible par « package »
- Empreinte minimal (outils d'élagage à l'édition de lien)
- Execution in place (Flash)

Architecture eCos



Implémentation de eCos

- C++ réduit (GNUTools)
 - Pas d'exception (g++ -fno-exceptions ...)
 - New retourne (void*) null
- Appel direct au nano-noyau
 - API kapi.h
- Assertion utilisées pour valider la consistance globale de l'OS
 - Pas de mode PANIC

Contextes d'exécution

- 4 contextes
 - Initialisation
 - ISR
 - DSR (Differed Service R???)
 - Thread
- Toutes les primitives ne sont pas disponibles dans tous les contextes
- Pas de privilèges pour les applications
 - Tout est lié ensemble
 - Tout l'espace mémoire est accessible
 - Pas de nettoyage automatique des ressources
- WatchDog (chien de garde) ???

Outillages eCos (i)

- sous Linux et Windows/Cygwin

■ Configurateur

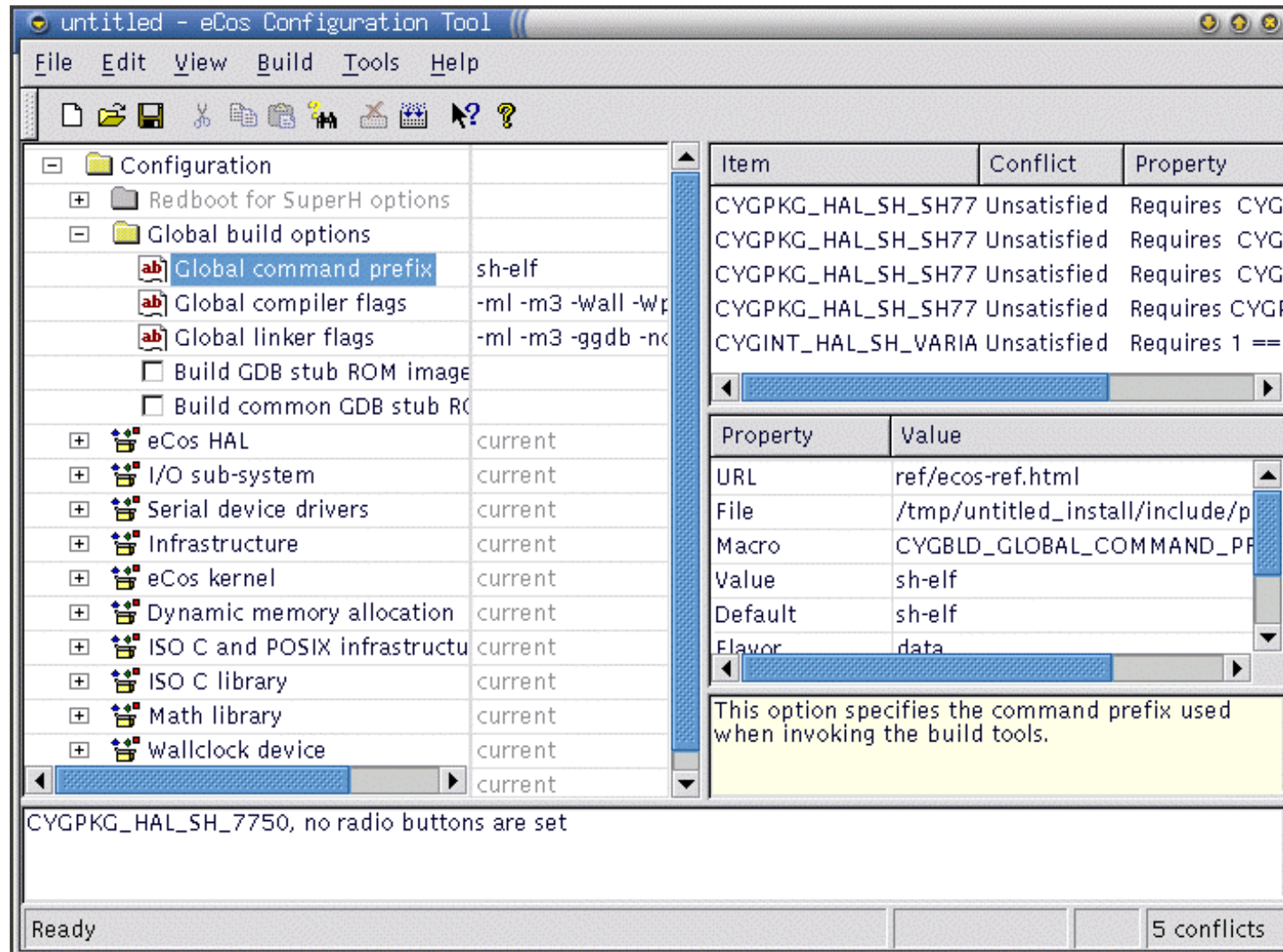
- Produit un .ecc pour un configuration
- Notion de template (configuration partielle)

■ Outils de développement

- GNUTools pour la Cross-compilation
 - gcc, binutils, gdb

Outillages eCos (ii)

Le configurateur



Outillage eCos (iii)

- Émulateur eCos sur Linux: eCos synthétique
- RedBoot *Red Hat's Embedded Debug and Bootstrap Program*
 - bootstrap/loader
- Chargement
 - Série, Ethernet, Flash, ROM

RedBoot

Red Hat's Embedded Debug and Bootstrap Program

■ bootstrap/loader

- Chargement d'un noyau à partir
- de la flash
- du réseau (BOOTP, TFTPBOOT)
- du port série

Package eCos

- Pilotes périphérique
- Flash File System
- Couches réseaux
 - Plusieurs piles TCP/IP (BSD, AWIP, ...)
 - Light weight IP
 - SLIP, PPP
- Services Internet
 - HTTPD et FTPD embarqués
- Java
 - μ VM en 32 KB RAM
- Sécurité
 - PKI <http://www.europepki.org>
 - Lib Crypto
 - OpenSSH / OpenSSL
- Contributions tierces
 - <http://sources.redhat.com/ecos/contrib.html>

Licence GPL+exemption (pour 2.0)

- Intégrable dans un produit commercial

- Communauté open source eCos
 - Initialement (97) Cygnus puis Red Hat par rachat de Cygnus
 - puis eCosentrix + CVS Red Hat
 - Inconvénients
 - Pas d'évolution si personne pour la coder
 - Pas de support garantie
 - Avantages
 - Pas de licence
 - Support/Ingénierie par des sociétés spécialisés (Silicomp, ...)

Bibliographie

- <http://sources.redhat.com/ecos/>
- Anthony Massa, Embedded Software Development with eCos, Ed Prentice Hall, ISBN: 0-13-035473-2
 - www.phptr.com/massa
- Doctor Dobb's Journal (Nov 2002)
 - <http://www.ddj.com/articles/2002/0211/>
- Software Development Magazine (Oct 2002)
 - <http://www.sdmagazine.com/documents/s=7578/sdm0210f/0210f.htm>
- EDN (Sep 5, 2002)
 - <http://www.e-insite.net/ednmag/index.asp?layout=article&articleid=CA240905&update=9/5/2002>
- Embedded Systems Programming (Aug 2002)
 - <http://www.embedded.com/story/OEG20020729S0043>
- Embedded Systems Programming (Jan 2002)
 - <http://www.embedded.com/story/OEG20011220S0059>

<http://www-adele.imag.fr/~donsez/cours>

Think

Didier DONSEZ

Université Joseph Fourier

IMA –IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`,

`Didier.Donsez@ieee.org`

Think

■ Framework de construction de kernel

- Principe: Limiter au maximum concepts et abstraction
- Préoccupation :
 - Extensibilité / Flexibilité / Adaptabilité / Modularité / Efficacité / Performance

■ Architecture orienté composants

- Interfaces relié par des composants « liaison »
- Composants composites
- Composants reconfigurables dynamiques
- Kortex
 - Bibliothèque de composants « systèmes » portables : Threads, MV, Pilotes, ...

■ Cibles

- PPC, IA32, H8/3298 , TINI DS80C390 (8051)

■ Projet open source

- INRIA/SARDES & FT RD puis ObjectWeb (<http://think.objectweb.org>)

Concepts

■ Composant

- Possède une ou plusieurs Interfaces de service (décrite en Java) associées à un nom
- Implémentés en C
- Utilisent les services d'un canevas pour communiquer entre eux (liaison)

■ Ressource matériel

- Réifié par un composant

■ Liaisons Flexibles

- Canal de communication inter-composant
- Locales / distantes
- Assemblage de composants (1 ou plusieurs)

Concepts

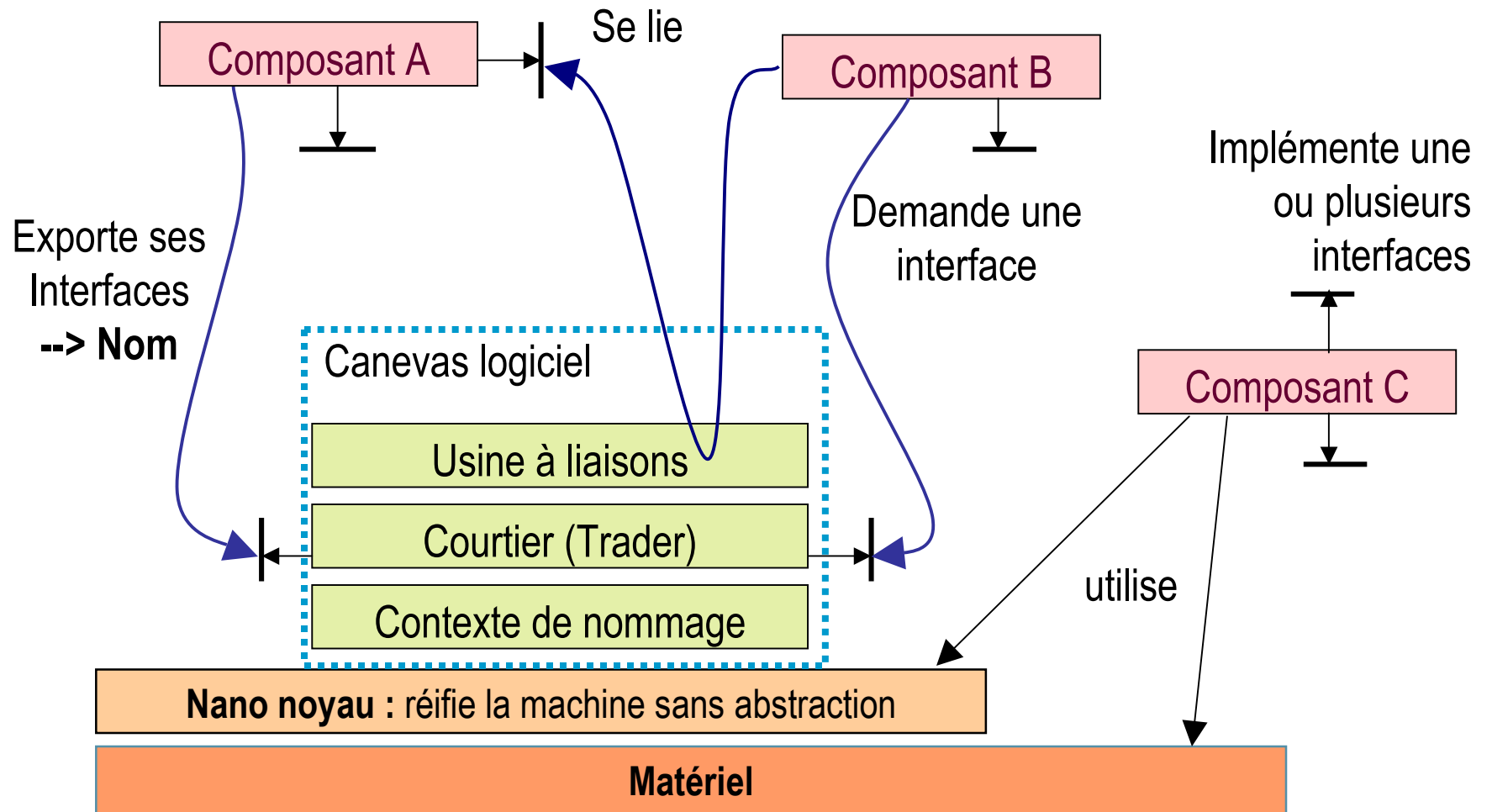
■ Canevas Logiciel

- Naming Context
- Trader (Courtier)
- Binding factory (Usine de liaisons)

■ NanoNoyau

- L'essentiel pour booter !

Architecture



Cas d'étude

ThinkRCX pour le Kit Lego MindStorm

■ Matériel Lego/RCX

- μ C 8bits H8/3298 16 MHz
- 6 Ko ROM
 - Pilote bas niveau des périphériques
 - Chargeur du « firmware »
- 32Ko RAM
 - Firmware : OS + Application
- 3 ports entrées
 - Température, vitesse rotation moteur, détection de choc, ...
- 3 ports sortie
 - Moteurs
- IrDA, Clavier 4 touches, LCD 5 caractères



Cas d'étude

ThinkRCX pour le Kit Lego MindStorm

■ Noyau Think

- même source C que celle du PowerPC
- Mini ORB (NamingFactory, BindingFactory), Mini Courtier
- Chargeur de programme (démarré l'exécution)

■ Application de test

- PathFinder : Recherche de chemins avec détection d'obstacles

■ Empreinte mémoire

- Think+PathFinder = 6Ko
- LegOS+PathFinder = 28Ko
- LejOS+PathFinder = 25Ko
 - LegOS et LejOS sont 2 OS dédiés au RCX

■ Durée de développement

- 1 jour pour le portage noyau
- 1 semaine pour le comportement PathFinder

Bibliographique

- Jean-Philippe Fassino, Jean-Bernard Stefani, Julia Lawall, Gilles Muller, « THINK: A Software Framework for Component-based Operating System Kernels », In Proceedings of Usenix Annual Technical Conference, Monterey (USA), June 10th-15th, 2002.
- Jean-Philippe Fassino, Jean-Bernard Stefani, « Think : un noyau d'infrastructure répartie adaptable », Deuxième Conférence française sur les Systèmes d'Exploitations (CFSE-2), Paris (France), Avril 2001.
 - <http://arcad.essi.fr/publications/fassino.pdf>
- Jean-Philippe Fassino, « THINK : vers une architecture de systèmes flexibles », Thèse de Doctorat de l'École Nationale Supérieure des Télécommunications, Décembre 2001.
 - <http://sardes.inrialpes.fr/papers/files/01-Fassino-PhD.pdf>

Lectures

■ Présentations

- <http://www.guilde.asso.fr/rencontres/4JLG/conferences.php>

Autres systèmes

■ RTEMS

- <http://www.rtems.com/>

■ OpenPlug

- <http://www.open-plug.com/>