


Matériel de réseau



heig-vd  
Haute Ecole d'Ingénierie et de Gestion  
du Canton de Vaud

---

Chapitre 1

# Introduction

---

Jürgen Ehrensberger IICT/HEIG 1-Introduction.ppt

## Contenu du chapitre


---

Introduction à la [partie I](#) :

### Linux embarqué

1. Systèmes embarqués
2. Systèmes d'exploitation embarqués
3. Linux comme OS embarqué
4. Licences Logiciels Libres
  - GPL, LGPL, BSD, Copyright

---



Jürgen Ehrensberger IICT/HEIG Cours MAR 1-Introduction 2

## Exemples de systèmes embarqués

- Photocopieuses
- Contrôle électronique dans les voitures
- Contrôleurs de machines industrielles
- Systèmes de contrôle des avions
- Appareils de divertissement (lecteurs MP3, lecteurs DVD, TV modernes)
- Téléphones mobiles
- PDA
- **Dispositifs de réseau** (routeurs, switches, AP WiFi, firewall, ...)



## Systemes embarqués

- Possèdent, comme les PC habituels,
  - un microprocesseurs
  - un OS
  - des applications spécifiques
- Mais intégrés dans le dispositif qu'ils contrôlent
  - Limitation de **taille**
  - Limitation des ressources disponibles
  - Interaction limitée avec l'utilisateur
  - Fonctionnalités spécialisées



## Taille

- Concerne les ressources disponibles (processeur, RAM, mémoire de stockage)

### Taille petite

- Clavier sans fil, souris gaming
- 2-4 MB RAM, microcontrôleur peu puissant (8-16 bit)

### Taille moyenne

- Lecteurs MP3, Natels, dispositifs de réseau, ...
- 64 MB RAM, 32 MB ROM, processeur relativement puissant (32 bit)

### Taille grande

- Exemples: Équipements des opérateurs, simulateurs d'avion
- Souvent multi-processeur, beaucoup de mémoire



## Contraintes de temps réel

- Permet de définir des limites au délai de réponse d'un système

### Temps réel strict (*hard real-time*)

- Le dépassement d'un délai défini est considéré comme une erreur du système
  - Contrôle des freins dans une voiture, contrôle de machines

### Temps réel souple (*soft real-time*)

- Des dépassements du délai sont tolérés quand ils sont rares
  - Lecteur MP3

### Sans contraintes temps réel (*non real-time*)

- Doit correctement réagir aux événements, mais sans délai imposé
  - PC, PDA, dispositifs de réseau (sauf très haut de gamme)

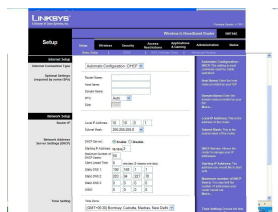


## Interaction avec l'extérieur

- Interface utilisateur limitée
  - Ecran tactile, touches, LEDs, etc



- Connexion au réseau
  - Configuration via interface Web / ligne de commande
  - Transfert de fichiers
  - Traitement de trafic réseau



## Systèmes d'exploitation embarqués

- PDA / Tablets
  - [Palm webOS](#)
  - [Windows CE / Mobile / 8](#)
  - [Linux - Android](#)
- Smartphones / Tablets
  - [Palm webOS](#)
  - [Apple iPhone OS](#)
  - [Windows Mobile / Phone / 8](#)
  - [Symbian OS](#)
  - [Linux - Android](#)
- Routeurs
  - [Cisco IOS](#) : propriétaire, avec ou sans temps réel
  - [JunOS](#) de Juniper: temps réel, basé sur FreeBSD
  - [OpenWrt](#) : OS Linux
- Microcontrôleurs
  - [eCos](#): OS non Linux de Red Hat, Open Source
  - [FreeRTOS](#) : OS temps réel, GPL
  - [LynxOS](#): OS temps réel commercial, compatible Linux
  - [QNX](#): OS temps réel, type Unix (utilisé par Cisco)
  - [BSD](#): NetBSD, FreeBSD, OpenBSD
  - [Linux](#): OS commercial ou libre
  - [RTLinux](#): Linux temps réel, commercial ou libre
  - [uCLinux](#): OS Linux réduit, pour systèmes à forte contrainte, libre
  - [VxWorks](#): OS temps réel commercial de WindRiver
  - [Windows CE](#)



## OS des dispositifs de réseau

### Cisco IOS et Juniper JunOS

- Propriétaires, ne sont pas commercialisés séparément

### Linux embarqué

- ~48% du marché des systèmes embarqué (2007)
- Commercial ou libre

### Windows CE

- ~13% du marché, notamment PDA (2007)

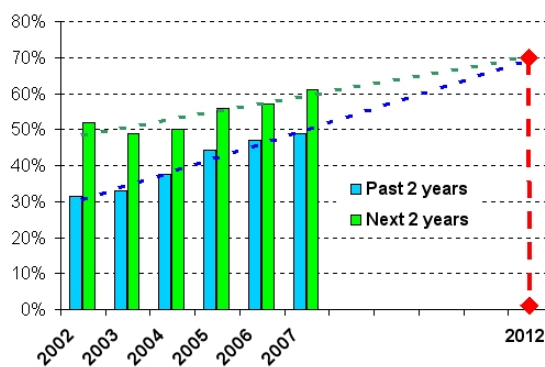
### VxWorks de WindRiver

- ~30% du marché (2009)
- Aviation, réseaux, contrôle industriel



## Linux comme OS embarqué

Past/future Linux use



Source : linuxfordevices.com



## Windows CE (Consumer Electronics)

- OS embarqué temps réel
  - Noyau différent de celui de Windows sur PC
  - OS modulaire,
    - Le développeur choisit les composants (GUI, réseau, pilotes, ...)
  - Code source [disponible](#) et [modifiable](#)
    - Sans obligation de partager les modifications avec Microsoft
- Coûts
  - Environnement « Platform Builder »: ~ \$ 1'000
  - Royalties: \$3 - \$16 par appareil vendu



## Windows Embedded

### Variantes

- Windows Embedded Compact
  - Pour des Set-Top Box, contrôle industriel, réseau
  - Sur x86, ARM, MIPS, SuperH
- Windows Phone (ancien Microsoft Windows Mobile)
  - Pour PDA, smartphones
  - Sur plateformes ARM
  - Avec versions réduites de Word, Excel, support pour SMS, GSM, ...
- Windows Embedded Standard
  - Basé sur Windows XP, non pas sur Windows CE
  - Modulaire
  - Sur plateformes x86
  - Pour systèmes médicaux, consoles de jeux, appareils multimédia

### Avantages

- Maturité
- Richesse des applications
- Coût total bas
- Délai de développement court

### Inconvénients

- Royalties :
  - Beaucoup de sociétés excluent le paiement de royalties
- Dépendance du fournisseur
  - Comme pour tous les produits propriétaires



## VxWorks de WindRiver

- OS temps réel commercial
- Répandu dans les systèmes exigeant une fiabilité élevée, des temps de réaction courts
  - Contrôle d'avions
  - Équipements télécoms des opérateurs
  - Mission Pathfinder sur Mars
- Coûts
  - Sans ou avec royalties
  - Coûts des licences de développement
    - En fonction des royalties










## Linux

- « Linux » peut désigner plusieurs choses
  - Le noyau Linux
  - Un système d'exploitation complet
    - Noyau Linux
    - Utilitaires GNU/Open Source (p.ex. commandes ls, cat, ...)
    - Bibliothèques, ...
  - Une distribution Linux
    - OS GNU/Linux préconfiguré
    - Outils d'administration, d'installation
    - Applications supplémentaires, libres ou commerciales



## Linux comme OS embarqué

- Ressources minimum
  - 2 MB de RAM
  - 2 MB de mémoire non volatile (ROM, Flash)
  - Processeur 32 bit (x86, PowerPC, ARM, MIPS, SuperH, ...)

Distributions embarquées (sondage 2005, Linuxdevices)

Distribution	Type	Part du marché
Debian (embarqué)	libre	13,8%
Distribution 'faite maison'	libre	13,4%
Red Hat	commerciale	10,0%
OpenEmbedded	libre	9,9%
MontaVista Linux	commerciale	7,2%
Fedora	libre	6,0%
Novell/SuSe	libre/commerciale	4,2%
Koan KLinux	commerciale	3,8%
RTAI	libre	3,8%



## Détails de quelques distributions

### Debian embarqué (emdebian)

- Distribution 100% libre
- Très stable, mais mise à jour lente
- Beaucoup d'applications
- Plateformes x86, ARM, PowerPC, MIPS

### Distribution faite maison

- OS composé à partir de composants Open Source
  - Noyau
  - Utilitaires GNU
  - Applications spécifiques
  - Configuration manuelle

### Snapgear Linux

- Distribution libre
- Spécialisée aux dispositifs de réseau

### MontaVista Linux

- Spécialisée aux dispositifs de réseau, avec temps réel strict
- Outils de développement
- Portée sur beaucoup de plateformes
- Coût: \$15'000 par développeur

### OpenWrt Linux

- Distribution libre
- Originellement orientée vers les WRT54G
- Disponible aujourd'hui sur beaucoup de plateformes
- Beaucoup d'applications
- Stable
- Mises à jour constantes





## Avantages de Linux embarqué

- Portabilité
  - Support pour toutes les plateformes courantes à 32 ou 64 bit
- Disponibilité du code source
  - Inspecter et modifier le code, apprendre des autres
  - Indépendance des fournisseurs
- Modularité et flexibilité
  - Le système (noyau et autres composants) peut être adapté aux besoins
- Disponibilité des pilotes et des protocoles de réseau
  - Bon support pour les périphériques et protocoles courants
- Coût bas
  - Beaucoup de logiciels sont disponibles gratuitement



## Inconvénients de Linux embarqué

- Licence GPL
  - Peut empêcher une société de commercialiser son produit
- Coût total du développement
  - Le temps des développeurs est souvent plus cher que le coût des outils de développement
  - Le développement sur la base de logiciels libres prend souvent plus de temps
    - Manque de support
    - Outils de développement faits maison
- Peu de support pour le temps réel
  - Linux n'est pas conçu pour le temps réel
- Difficulté de choisir une distribution
  - Travail de sélection dans un environnement peu stable



## Les licences « Logiciels Libres »

- Les différentes idéologies de « Libre »
  - « **Logiciels Libres** » (*Free Software*)
    - Terme créé par la « Free Software Foundation » (Richard Stallman)
    - Souligne l'aspect éthique de la distribution de logiciels
    - Souvent perçu comme anti-commercial par l'industrie
  - « **Open Source** »
    - Terme créé par la « Open Source Initiative »
    - Promotion des logiciels libres dans l'industrie et leur utilisation commerciale
    - Souligne la qualité des logiciels et les avantages d'utiliser le code source
- La plupart des licences satisfont aux deux philosophies
  - Termes « **FOSS** » (*Free and Open Source Software*)  
ou « **FLOSS** » (*Free/Libre/Open Source software*)  
pour éviter une connotation idéologique
- Licences les plus importantes: **GPL, LGPL, BSD**



## La licence GPL *GNU General Public Licence*

- Licence utilisée par la plupart des logiciels des distributions Linux
- Objectifs
  - Donner le droit d'utiliser, d'étudier, de modifier et de distribuer un logiciel
  - Garantir que ces droits ne peuvent pas être abrogés
  - Exclure toute responsabilité du développeur



## Stipulations de la GPL

- Droit de **copier un logiciel** GPL reçu, sans restrictions, pourvu que la licence et le copyright soient laissés intacts
- Un logiciel GPL est **fourni sans aucune garantie**, sauf si explicitement mentionné par le fournisseur du logiciel
- **Obligation de fournir le code source complet** à tout tiers **si le logiciel est distribué** sous forme de code exécutable
- **Exclusion de restrictions supplémentaires des droits** transmis au destinataire d'un logiciel
- Un logiciel qui modifie ou inclut du code GPL ne peut être distribué que **sous la même licence GPL**
- **Mais aucune obligation de distribuer un logiciel GPL**



## Interprétations

La GPL couvre uniquement la **modification** et la **distribution** d'un logiciel, non pas son **utilisation**

### Utilisation du logiciel

- N'est pas sujette à des restrictions
- Un programme compilé par un compilateur GPL ne devient pas GPL
- Un OS qui utilise le noyau Linux n'est pas (forcement) GPL

### Modification d'un logiciel GPL

- Le logiciel résultant est également couvert par la GPL
- S'il est distribué, par exemple sur un **dispositif embarqué, le client a le droit de demander, modifier et distribuer le code source**

### Distribution d'un logiciel

- Vous avez le *droit* mais pas l'*obligation*
- Une entreprise peut développer un logiciel propriétaire sur la base de code GPL et peut l'utiliser en interne, sans autres obligations
- Une société d'audit peut développer des logiciels sur la base de code GPL et l'utiliser chez un client, mais sur ses propres machines



## Utilisation ou modification/inclusion ?

### Règle générale

La modification/inclusion utilise le [code source](#) d'un logiciel GPL

- Exemples de modification/inclusion
  - Programme qui inclut du code source GPL
  - Programme qui utilise une librairie GPL
    - Linkage nécessite le code source des fichiers headers
    - Les librairies sont rarement distribuées sous GPL
  - Extensions du noyau Linux
- Zone grise:
 

Modules binaires Linux, par exemple pilotes propriétaires

  - Utilisent l'API standard du noyau
  - Sont tolérés actuellement par les développeurs du noyau



## Licences compatibles avec la GPL

- Que se passe-t-il si un logiciel utilise du code source couvert par deux licences différentes ?
- [GPL + licence compatible](#)
  - Le logiciel résultant est couvert par la GPL
  - La plupart des licences FOSS sont compatibles avec GPL
  - Voir liste sur le site de GNU
- [GPL + licence incompatible](#)
  - Un tel logiciel ne peut pas être distribué légalement
  - Il est donc déconseillé d'utiliser une licence incompatible GPL pour son propre développement
  - Exemple de licences incompatibles avec GPL
    - Licence Apache
    - Licence Mozilla / Netscape



GNU GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

---

■ **Déve**

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.


■ **Modif**

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

25




## La licence GNU LGPL

### *Lesser General Public Licence*

---

- Similaire à la GPL
  - Toute modification d'un logiciel LGPL est aussi couvert par la LGPL avec des obligations similaires à la GPL
- Mais **permet le linkage** avec une librairie LGPL sans imposer une licence au programme résultant
- La plupart des **librairies** Open Source sont LGPL
  - GLIBC: Librairie standard C de GCC
  - Librairies graphiques comme Gnome GTK, ...
- Seule obligation lors de la distribution
  - Il **faut permettre de remplacer la librairie** par une autre
  - Si linkage statique, fournir les fichiers code objet (\*.o)
  - Remplacement toujours possible avec un linkage dynamique

---


 Jürgen Ehrensberger IICT/HEIG Cours MAR 1-Introduction 26

## La licence BSD

### *Berkeley Software Distribution*

#### Licence très simple et permissive

##### Objectifs

- Exclure toute responsabilité du développeur
- Promouvoir l'utilisation (**commerciale** ou **libre**) du logiciel et du code source

##### Obligations

- Lors de modifications du programme, garder le texte de la licence dans les fichiers
- Si le logiciel est distribué sous forme binaire, inclure la licence dans la documentation
- Ne pas utiliser le nom du développeur ou de son organisation à des buts de promotion du logiciel
- **Le logiciel modifié doit indiquer qu'il contient du code BSD**

##### La licence BSD et ses variantes sont utilisées par beaucoup de logiciels

- Les OS BSD (FreeBSD, NetBSD, OpenBSD, ...)
- Base de Apple OS X, JunOS; code réseau utilisé p.ex. dans les premières versions de Windows
- Le système graphique X (licence MIT, similaire à BSD)

##### Licence compatible avec GPL

- Un programme qui combine du code GPL et BSD devient GPL



## La double licence – GPL et commerciale

- Il est difficile de commercialiser un logiciel GPL
  - Un client a droit au code source qu'il peut ensuite redistribuer sans frais
- La double licence permet de **combiner les avantages** de l'Open Source avec les intérêts commerciaux
  - Le logiciel est **distribué gratuitement sous la licence GPL**
  - Une deuxième licence commerciale permet aux clients d'acquérir le logiciel et de **l'utiliser dans un produit commercial** sans être contrainte par la GPL
    - Exemple: librairie graphique Qt
      - Gratuit pour l'utilisation non commerciale
      - Licence payante pour le développement de logiciels commerciaux
- Uniquement le propriétaire du code source complet (copyright !) peut faire cela
  - **Il est impossible de prendre du code GPL développé par d'autres personnes et de le distribuer sous une autre licence**



## Le copyright

- Loi qui donne à l'auteur d'une oeuvre le **droit exclusif de contrôler la reproduction** pendant un laps de temps
  - Dans un programme, indiquer « Copyright *Année Nom* »
- Le détenteur du copyright peut **distribuer son oeuvre sous une licence** qui définit les droits des utilisateurs
  - Le copyright est donc la base juridique de la licence GPL
- Les lois de copyright des différents pays sont similaires mais non identiques
  - Les lois européennes *pourraient* être incompatibles avec la GPL
    - Elles permettent de révoquer une licence, ce qui interdit de continuer à utiliser un logiciel
    - Elles ne permettent pas à un fournisseur d'un logiciel de déclinier toute responsabilité
  - La version 3 de la GPL essaie d'être plus internationale



## Les brevets

- Donne le droit exclusif d'exploiter une invention pendant un laps de temps (20 ans) sur un territoire déterminé (pays)
- En Europe, les logiciels ne peuvent pas (encore?) être brevetés
  - Le code source est protégé par le copyright
  - Les algorithmes peuvent être utilisés par d'autres personnes
- Aux États-unis, les logiciels et algorithmes peuvent être brevetés
  - Pour éviter de devoir payer des royalties, on devrait vérifier qu'un logiciel développé ne contient pas d'algorithme breveté !
  - Mais une grande partie des brevets ne passent pas un examen juridique

