Systèmes d'exploitation pour l'embarqué

UV 5.2 - Exécution et Concurrence

Paul Blottière

ENSTA Bretagne

2018 / 2019

https://github.com/pblottiere

Amélioration continue

Contributions

- Dépôt du cours : `https://github.com/pblottiere/embsys`
- Souhaits d'amélioration, erreurs, idées de TP, ... : ouverture d'Issues
- Apports de corrections : Pull Request

Organisation

Volume horaire : 35 heures

Organisation

Volume horaire : 35 heures

9 cours :

- Introduction, Généralités
- Programmation Système
- Linux embarqué

Organisation

Volume horaire : 35 heures

9 cours :

- Introduction, Généralités
- Programmation Système
- Linux embarqué

Le reste : des travaux pratiques

- Programmation sous Linux
- Utilisation d'outils pour l'embarqué : raspberry-pi

Introduction

Plan

1. Un peu d’histoire
2. Les normes
3. Logiciel Libre et Logiciel Open-Source
4. Licences de distribution
5. Définitions et propriétés
6. Les OS embarqués existants
7. Comment choisir?
8. Et si on choisit Linux...
9. Aspects matériels
10. Conclusion
11. Références

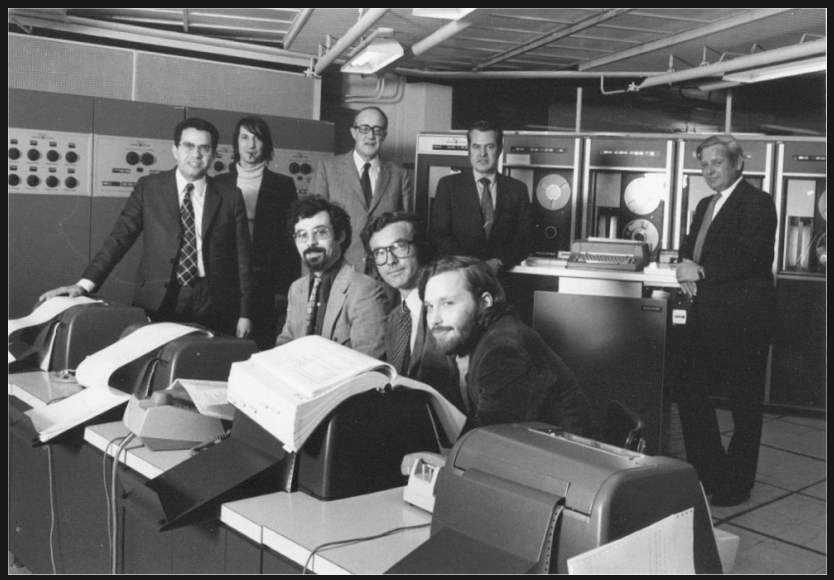
Un peu d’histoire (1)

Le premier jeu vidéo

Un peu d’histoire (1)

Le premier jeu vidéo

1964 : Multics

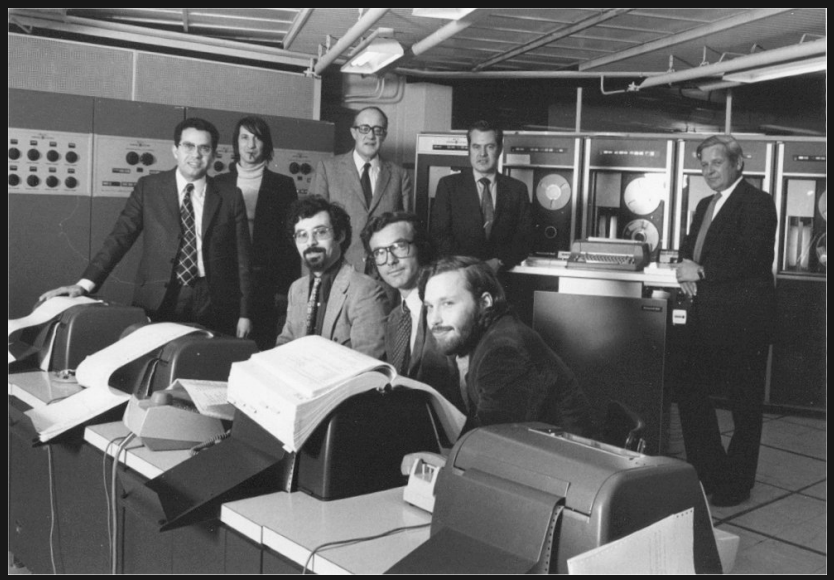


[GE-645 - 1972]

Un peu d'histoire (1)

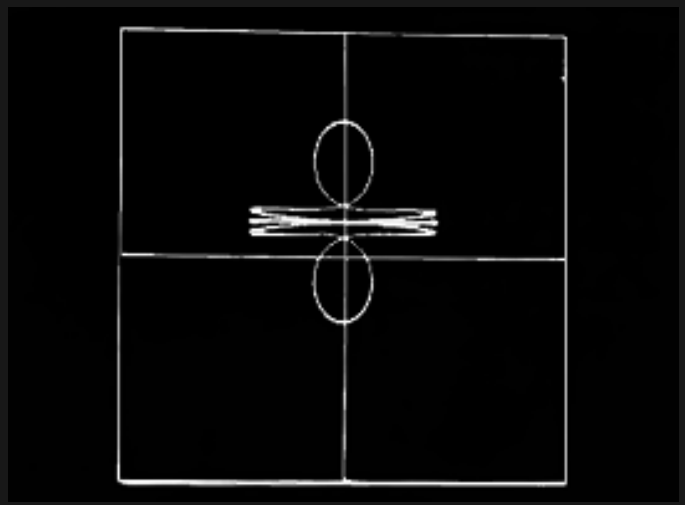
Le premier jeu vidéo

1964 : Multics



[GE-645 - 1972]

1969 : Space Travel/Unics par Ken Thompson



[Gameplay image of Space Travel]

Un peu d’histoire (2)

La naissance du langage C

Un peu d'histoire (2)

La naissance du langage C

1971 : Le C par Dennis Ritchie

[Dennis Ritchie et Kenneth Thompson]

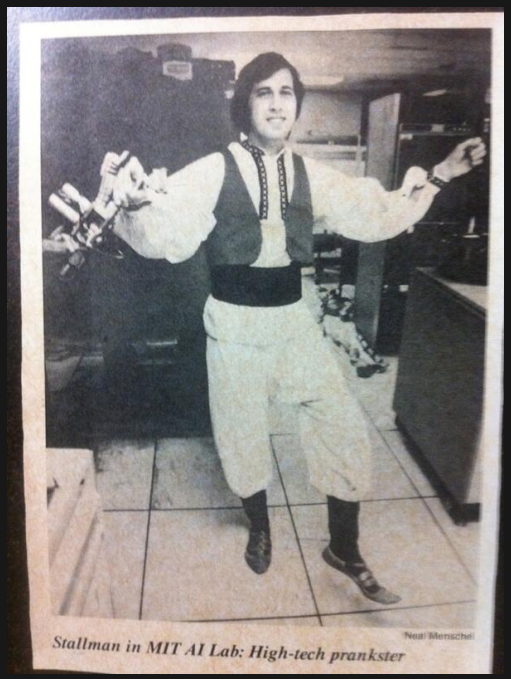
Un peu d'histoire (2)

La naissance du langage C

1971 : Le C par Dennis Ritchie

[Dennis Ritchie et Kenneth Thompson]

1983 : GNU par Stallman



[RMS in MIT - pre 1985]

Un peu d’histoire (3)

Un hobby devenu célèbre

Un peu d’histoire (3)

Un hobby devenu célèbre

1987 : Linux par Linus Torvalds

[Linus Tovalds]

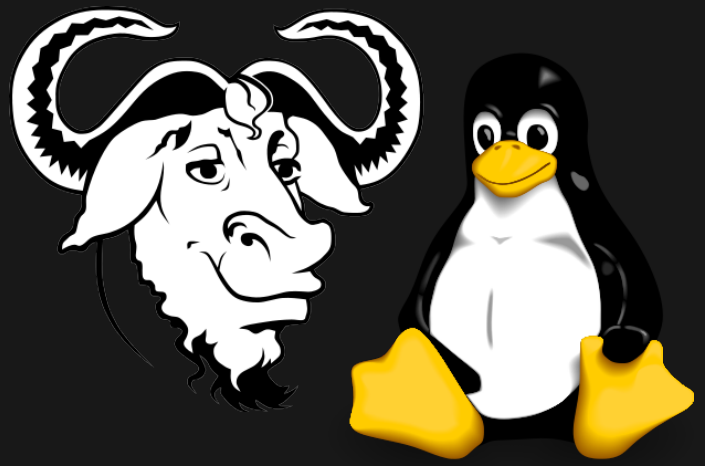
Un peu d’histoire (3)

Un hobby devenu célèbre

1987 : Linux par Linus Torvalds

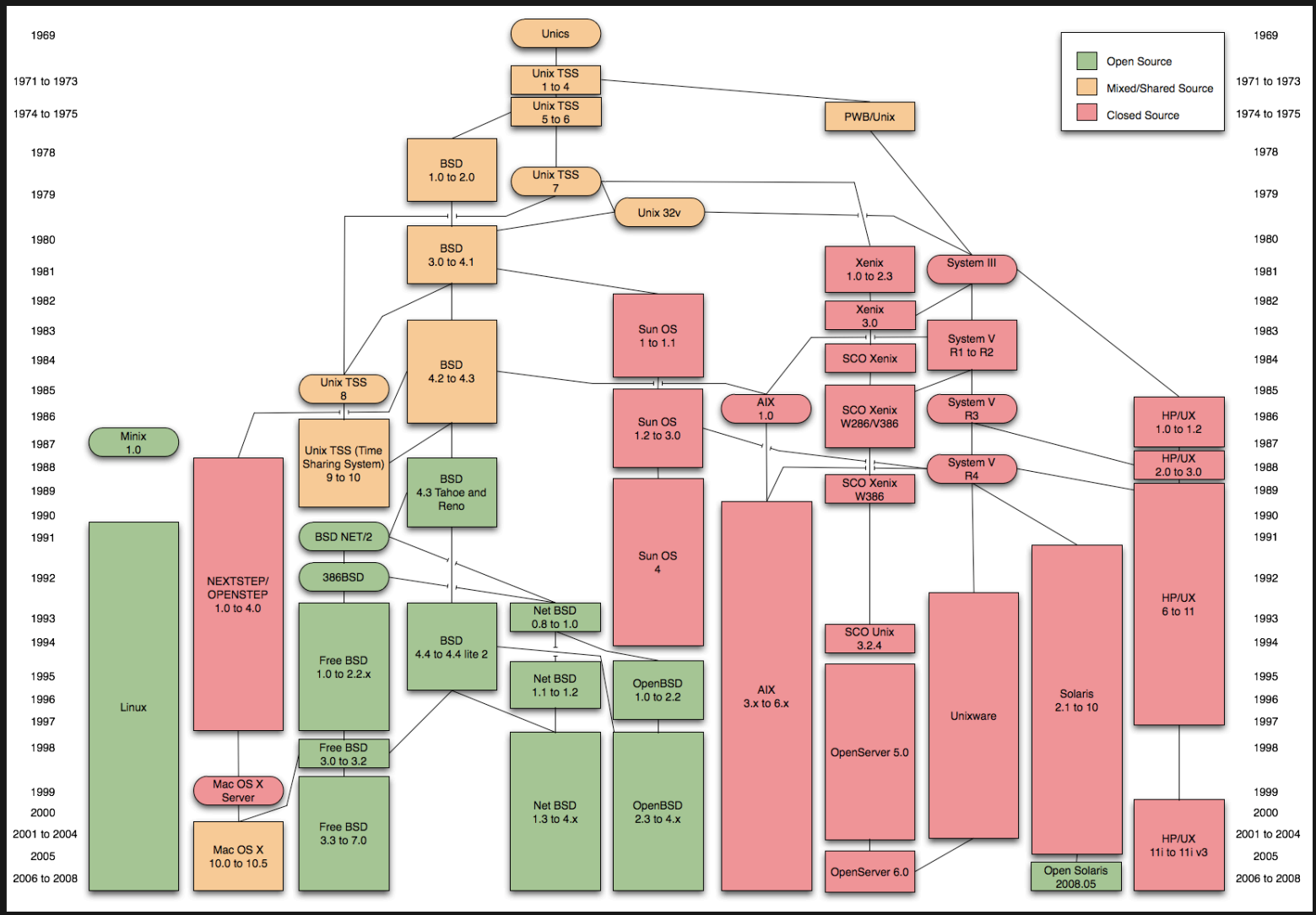
[Linus Tovalds]

1992 : GNU / Linux



Un peu d'histoire (4)

De nos jours



[UNIX timeline]

Les normes (1)

SUSv4

Des normes sont nécessaires pour assurer une compatibilité des logiciels entre systèmes d'exploitation:

- 1988 : Portable Operating System Interface (POSIX). Plusieurs versions : 1, 1.b, 1.c
- 1997 : UNIX98 ou Single UNIX Specification V2
- Standard actuel : SUSv4 (fusion entre POSIX et UNIX98)

`http://www.unix.org/version4/`

Les normes (2)

POSIXLY_CORRECT SIR!

POSIX:

```
tool [-a][-b][-c option_argument] \
     [-d|-e][-f[option_argument]][operand...]
```

Sous GNU/Linux, les règles sont différentes!

Les normes (2)

POSIXLY_CORRECT SIR!

POSIX:

```
tool [-a][-b][-c option_argument] \
     [-d|-e][-f[option_argument]][operand...]
```

Sous GNU/Linux, les règles sont différentes!

Conséquences sous une distribution Linux:

```
    > ls -a .
    .vimrc .vim devel doc
    > ls . -a
    .vimrc .vim devel doc
    > POSIXLY_CORRECT=1 ls . -a
    ls: impossible d'accéder à -a: Aucun fichier ou
        dossier de ce type
    .vimrc .vim devel doc
```

Logiciel Libre et Logiciel Open Source (1)

Question de philosophie...

- Logiciel Libre : code source ouvert et pouvant être modifié. Fond philosophique => liberté des utilisateurs (Free as Freedom)!

Logiciel Libre et Logiciel Open Source (1)

Question de philosophie...

- Logiciel Libre : code source ouvert et pouvant être modifié. Fond philosophique => liberté des utilisateurs (Free as Freedom)!
- Logiciel Open Source : code source ouvert et pouvant être modifié... Fond pragmatique => efficacité, praticité!

Logiciel Libre et Logiciel Open Source (1)

Question de philosophie...

- Logiciel Libre : code source ouvert et pouvant être modifié. Fond philosophique => liberté des utilisateurs (Free as Freedom)!

- Logiciel Open Source : code source ouvert et pouvant être modifié... Fond pragmatique => efficacité, praticité!

`http://www.gnu.org/philosophy/free-software-for-freedom.fr.html`

Logiciel Libre et Logiciel Open Source (2)

Origine des licences GPL et LGPL

1985

Logiciel Libre et Logiciel Open Source (2)

Origine des licences GPL et LGPL

1985

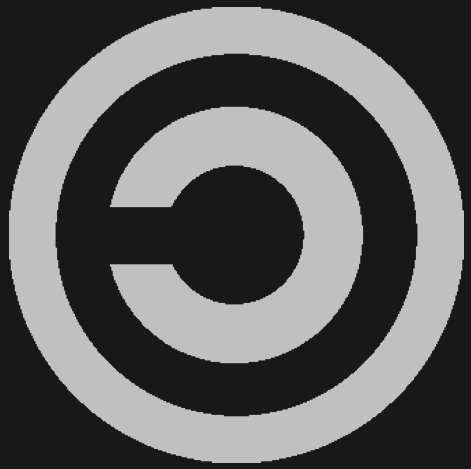
Son but:

- protéger les utilisateurs contre les logiciels ”privateurs”
- élaborer des licences de distribution

Licences de distribution (1)

Copyleft

L’utilisateur refuse qu’une évolution quelconque de son travail soit accompagnée d’une restriction!



Licences de distribution (2)

GPL et LGPL

Licence libre copyleft :

- GPL : GNU General Public Licence. Édition de liens possible qu'avec du code GPL!

  `http://www.gnu.org/licenses/gpl-3.0.fr.html`

- LGPL : Lesser GPL. Édition de liens moins restrictive!

  `http://www.gnu.org/licenses/lgpl-3.0.fr.html`

Licences de distribution (2)

GPL et LGPL

Licence libre copyleft :

- GPL : GNU General Public Licence. Édition de liens possible qu'avec du code GPL!

  `http://www.gnu.org/licenses/gpl-3.0.fr.html`
- LGPL : Lesser GPL. Édition de liens moins restrictive!

  `http://www.gnu.org/licenses/lgpl-3.0.fr.html`

Licence libre non copyleft :

- BSD : les versions modifiées ne sont elles mêmes pas nécessairement libres!
- ...

Licences de distribution (3)

Conséquences dans la vie courante...

Exemples de licences:

- nmap : GPL `/usr/share/doc/nmap/copyright`
- GNU C library : LGPL
  `/usr/share/doc/libc6/copyright`

Licences de distribution (3)

Conséquences dans la vie courante...

Exemples de licences:

- nmap : GPL `/usr/share/doc/nmap/copyright`
- GNU C library : LGPL `/usr/share/doc/libc6/copyright`

Debian et le DFSG (Debian Free Software Guideline) :

- `main` : paquets conformes au DFSG
- `contrib` : paquets conformes au DFSG mais avec des dépendances en dehors du `main`
- `non-free` : paquets non conformes au DFSG

Licences de distribution (4)

Conséquences dans la vie courante...

VRMS (Virtual RMS) :

```
> vrms
Contrib packages installed on multi
flashplugin-nonfree                Adobe Flash Player
1 contrib packages, 0.0% of 2877 installed packages
```

[RMS : Saint Ignucius]

Définitions et propriétés (1)

Kernel et système d'exploitation

Kernel :

- noyau d'un système d'exploitation
- gère les ressources matérielles
- permet la communication entre composants logiciels/matériels
- existe plusieurs architectures

Définitions et propriétés (1)

Kernel et système d'exploitation

Kernel :

- noyau d'un système d'exploitation
- gère les ressources matérielles
- permet la communication entre composants logiciels/matériels
- existe plusieurs architectures

Système d'exploitation:

- kernel + logiciels comme compilateur, shell, debuger, ...
- couche d'abstraction par rapport au matériel
- interface générique de programmation

Définitions et propriétés (2)

Dans le monde des systèmes embarqués...

Système embarqué:

- composition d'une partie électronique et logicielle
- souvent très limitée d'un point de vue ressources matérielles (CPU, mémoire, ...)
- autonome, durée de vie très longue (plus de 20 ans pour les systèmes militaires)
- doit respecter des contraintes d'environnement (vibration, chaleur, ...)

Définitions et propriétés (3)

Dans le monde des systèmes embarqués

Système d'exploitation embarqué:

- OS sur lequel un logiciel embarqué va être exécuté.
- Contrainte forte par rapport à la consommation matérielle / énergétique.
- OS classique souvent inenvisageable

Définitions et propriétés (3)

Dans le monde des systèmes embarqués

Système d'exploitation embarqué:

- OS sur lequel un logiciel embarqué va être exécuté.
- Contrainte forte par rapport à la consommation matérielle / énergétique.
- OS classique souvent inenvisageable

Système d'exploitation temps réel (vs temps partagé):

- garantit les temps de réponse (temps réel dur/mou, préemptivité du Kernel, ...)
- Voir le cours associé!

Définitions et propriétés (4)

Dans le monde des systèmes embarqués

Logiciel embarqué:

- logiciel intégré pour une application dédiée
- un bon logiciel embarqué est un logiciel dont on oublie l'existence!

Définitions et propriétés (4)

Dans le monde des systèmes embarqués

Logiciel embarqué:

- logiciel intégré pour une application dédiée
- un bon logiciel embarqué est un logiciel dont on oublie l'existence!

Linux embarqué

- Kernel Linux + composants open-source
- construit sur mesure par rapport aux besoins

Les OS embarqués existants (1)

Sans base Linux

- VxWorks : noyau temps réel le plus utilisé dans l'industrie. Licence très coûteuse!
- QNX : noyau temps réel. Gratuit pour les applications non commerciales.
- micro-C OS : temps réel pour micro contrôlleur. Gratuit pour l'enseignement.
- Windows Phone : pour mobile. Se veut concurrent d'Android.
- Plein d'autres : LunxOS, Nucleus, eCos, ...

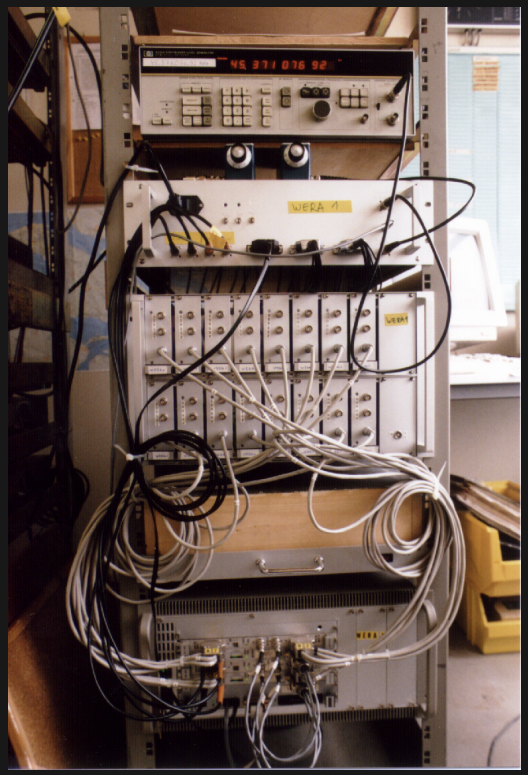
Les OS embarqués existants (2)

À base de Linux

- Wind River Linux : avec extension temps réel RTLinux
- ELDK (Embedded Linux Development Kit) : Fournit une distribution complète pour les architectures PowerPC, ARM et MIPS. Sous licence GPL.
- Google Android : application en Java. Existe un SDK C/C++.
- Tizen : dernier né (2012), Open Source. OS de la Samsung Gear S2!
- Plein d'autres : MontaVista Linux, BlueCat Linux, ...

Les OS embarqués existants (3)

Exemples



[Radars HF Wera déployés sur les côtes Bretonnes : VxWorks]

[OmniBusBox Ballard technology : ELDK]

Comment choisir?! (1)

Systèmes propriétaires ?

Avantages et inconvénients:

- Pas d'effet de masse donc cher (et cher donc pas d'effet de masse...)
- Personnes maîtrisant les outils associés rares sur le marché de l'emploi : rare donc cher!
- La durée de vie d'un système embarqué est très élevée. Donc que se passe-t-il si l'entreprise propriétaire disparaît? Risqué...
- Mais en théorie, très bon support, très bonne réactivité!
- Garantie en cas de problème, responsabilité de l'entreprise!

Comment choisir?! (2)

Systèmes Open Source?

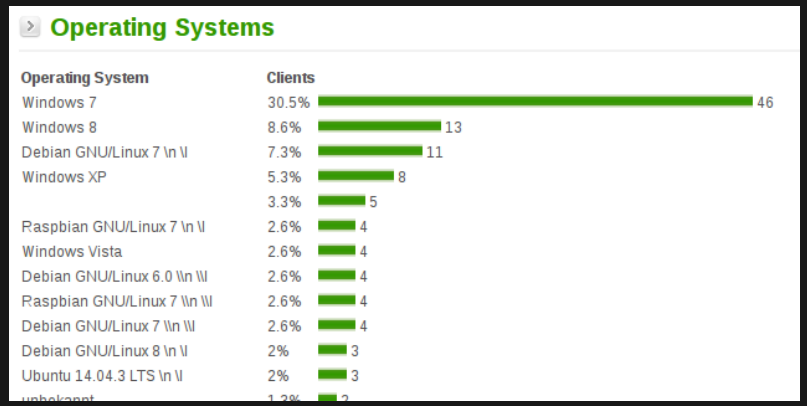
Avantages et inconvénients:

- Redistribution sans royalties
- Code source ouvert et donc modifiable à volonté! Mais les licences trop ouvertes (comme la GPL) peuvent poser problèmes pour les entreprises ne souhaitant pas reverser leurs travaux!
- Logiciel fourni ”As is”. Problème de responsabilité?
- Argument non nécessairement objectif : code souvent de bien meilleur qualité!

Et si on choisit Linux... (1)

Avantages

Reconnu pour sa très grande fiabilité!



[Uptime project - Operating System statistics - 2015]

Top 10 - overall

Rank	User	Current uptime	Availability	Record Uptime	Operatingsystem
1	tzeappa	2y 26d 23h 19m	99,9%	2y 26d 23h 19m	CentOS release 6.7 (Final)
2	srufer	351d 8h 21m	99,9%	351d 8h 21m	Debian GNU/Linux 8 (Cubietruck)
3	HenryN-vbox	295d 18h 3m	99,9%	1y 288d 9h 16m	Debian GNU/Linux 6.0 \n \l
4	nrsgzz	258d 3h 39m	99,9%	258d 3h 39m	Windows 7
5	jaraeez	135d 7h 34m	99,9%	154d 4h 39m	Raspbian GNU/Linux 7 \n \l
6	ktb	132d 10h 26m	99,9%	132d 10h 26m	Crazy Monkeys 1.0
7	HeiligerBimBamNAS	126d 3h 1m	99,9%	165d 3h 54m	
8	Furandyserver	88d 6h 7m	99,9%	179d 16h 27m	Ubuntu 10.10 \n \l
9	lambda-smon	82d 18h 30m	98,3%	82d 18h 30m	Debian GNU/Linux 8 \n \l
10	CaMPaRi	77d 11h 17m	99,9%	1y 98d 15h 36m	Debian GNU/Linux 7 \n \l

[Uptime project - top 10 - 2015]

Et si on choisit Linux... (2)

Avantages

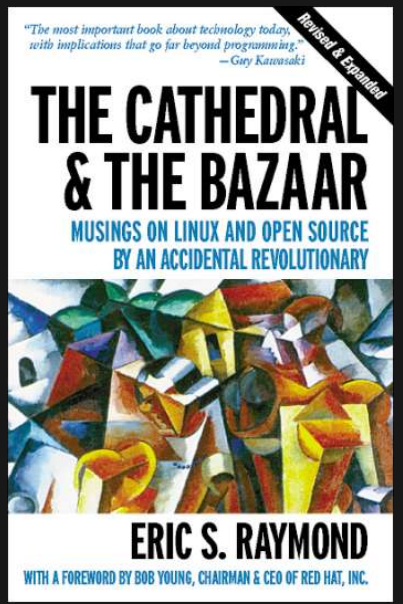
Comme vu dans les parties précédentes:

- Peu cher : pas de royalties, outils de développement libres, ...
- Portabilité : x86, arm, ppc, amd, sparc, ...
- Open Source

Et si on choisit Linux... (3)

Inconvénients

- Beaucoup de licences. Plus compliqué qu'une seule licence propriétaire.
- Beaucoup de solutions pour faire la même chose contrairement à une solution propriétaire sur étagère.



[The Cathedral and the bazaar]

Aspects matériels (1)

Processeurs

Le kernel Linux tourne sur de très nombreuses architectures de processeurs 32 bits et 64 bits.

[Samsung Gear S2 - Tizen]

[ARDRONE 2.0 - Linux 2.6.32 - Parrot]

Aspects matériels (2)

MMU

Memory Management Unit (unité de gestion de mémoire) permet de :

- protéger l'espace mémoire des processus (segmentation fault)
- traduction entre adresses physiques et adresses virtuelles

Kernel version 2.5.46 : processeurs sans MMU supportés via la µClibc

Conclusion

[Distributions]

Références

- Linux Embarqué - Pierre Ficheux
- Développement système sous Linux - Christophe Blaess
- Modern Operating Systems - Andrew Tanenbaum