

# **Linux sur processeur Blackfin**

Jean-Christian de Rivaz

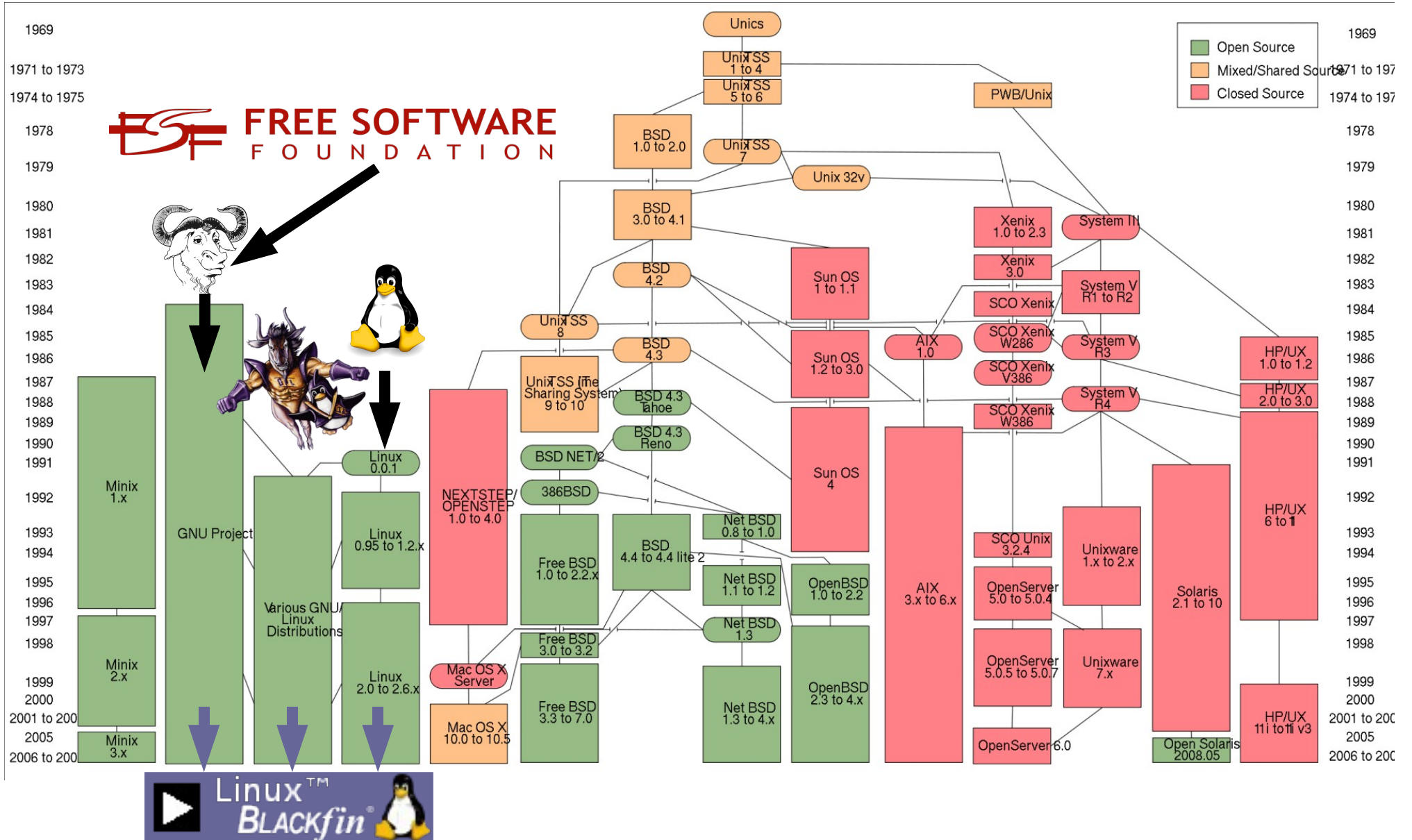
Advanced DSP Users Group

08 octobre 2008

# Déroulement

- Introduction
- Définition des termes et projets impliqués.
- Présentation de la distribution de référence Debian GNU/Linux.
- Comparaison avec la distribution uClinux pour Blackfin.
- Vue des services que propose le site [blackfin.uclinux.org](http://blackfin.uclinux.org).
- Quelques particularités qui peuvent faire perdre du temps.
- Le processus d'installation de la chaîne de construction.
- Description des principales images possibles et leur utilisation.
- Comment organise la mémoire flash et la programmer.
- Quelques approches possibles pour le système de fichier.
- Linux en multiprocesseur sur le Blackfin BF-561 double coeur.
- L'architecture logicielle mise en place au laboratoire MIS-SE.
- Conclusion

# Historique de UNIX



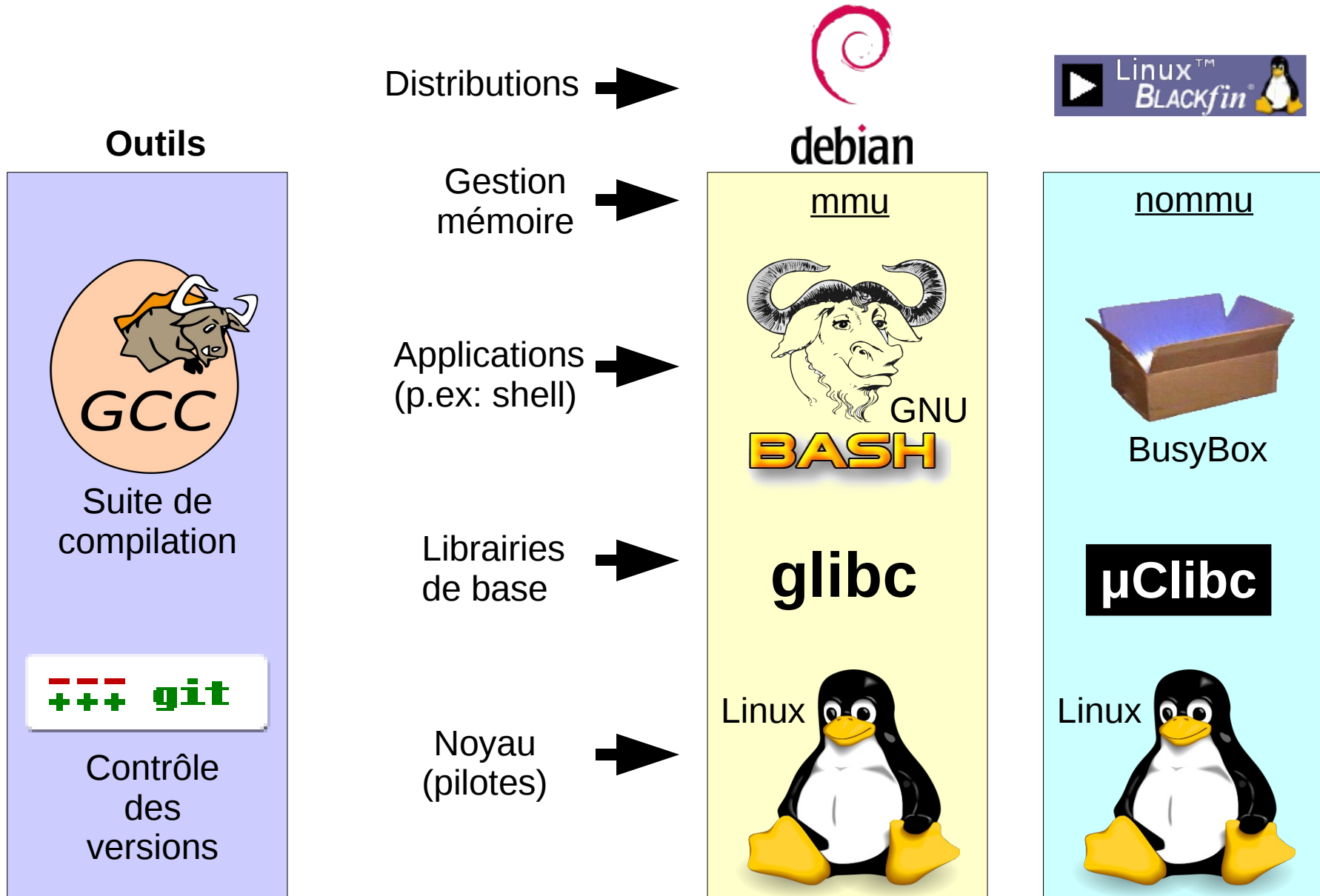
**FSF** FREE SOFTWARE FOUNDATION



## Introduction

- Un système complet émerge de la communication entre des projets qui résolvent chacun un problème spécifique. Chaque projet a une communauté qui lui est propre. Leur organisation sont différentes tout comme leurs conventions.
- Linux sur Blackfin émerge d'un historique de bientôt 40 ans de projets qui n'ont pas été créés dans ce but précis. Il y a donc forcément une certaine lourdeur, mais il y a surtout beaucoup d'expérience et de qualité.
- C'est un monde vaste. Préparez-vous à apprendre tout les jours quelques choses en plus. Même après 10 ans de pratique !
- Abonnez-vous aux mailing-listes des projets que vous utilisez.
- Définition rapide de UNIX, Linux, GPL, FSF, GNU, gcc, glibc, git et leur rôle dans le fonctionnement d'un système Linux sur le processeur Blackfin.
  - Utilisez Wikipedia et les sites de ces entités pour plus d'information.

# L'émergence des plusieurs projets



## Définitions: UNIX et Linux

- Unix est système d'exploitation créée en 1969 par AT&T et Bell labs qui a eu un énorme succès dans la recherche, les centres de calculs et les milieux académiques. Avec le temps, de nombreux dérivés de UNIX sont apparus. Leurs incompatibilités a rendu tout l'écosystème UNIX peu attractif.
- Linux est une implémentation sous licence GPL d'un noyau compatible avec UNIX, créé au début des années 1990 par le finlandais Linus Torvalds.
- Particularité: l'interface entre les pilotes et le noyau changeront toujours constamment: `Documentation/stable_api_nonsense.txt`
- Actuellement environ un millier de personnes contribuent à chaque nouvelles versions publiées par Linus Torvalds sur le site kernel.org. Il a y actuellement environ une version par saison.
- Il existe un nombre incalculable d'autres publications de Linux, dont celle pour Blackfin sur le site blackfin.uclinux.org.

## Définition: git

- Le système de contrôle de version `git` à été créé par Linus Torvalds en 2005 pour la gestion du noyau Linux.
- Architecture complètement décentralisée, ne nécessitant pas de connexion pour effectuer la grande majorité des opérations.
- En interne `git` introduit un concept radicalement nouveau: l'indexation par hashage du contenu.
- En surface, `git` propose des couches compatibles avec les anciens ténors du contrôle de version que sont en particulier CVS et Subversion.
- Avec `git`, il n'a pas de marche à suivre imposée, ce qui lui permet de remplir de nombreux rôles. Chaque projet peut décider librement du workflow adapté à son audience.
- L'adoption de `git` est fulgurante Il n'y a pratiquement aucun désavantage autre que d'apprendre à utilise un nouvel outil.

## Définitions: GPL et FSF

- La licence General Public Licence (GPL), créée par la Free Software Foundation, garantit l'accès au code source et sa libre modification tant que la licence est respectée. A noter que le comportement récursif que cette licence est légalement l'aspect le plus important. Aucun avocat n'a à ce jour trouvé une façon de la mettre en défaut.
- La Free Software Foundation (FSF) est une organisation à but non lucratif créée en 1983 par Richard Stallman pour aider au financement du projet GNU et de la communauté du Logiciel libre.
- GNU is Not UNIX (GNU), est une implémentation sous licence (L)GPL des bibliothèques et des applications de base que l'on trouve habituellement sur les systèmes UNIX. C'est le projet principale de la FSF.

## Les fondations du système GNU

- Les applications les plus fondamentales sont celles de la suite GNU C Compiler (`gcc`). C'est le compilateur utilisé pour le noyau Linux. Il supporte un très grand nombre de processeurs, principalement 32 et 64 bits. GCC supporte le processeur Blackfin pour faire de la compilation croisés (cross-compiler).
- La librairie la plus importante est la GNU library C (`glibc`) qui implémente les fonctions normalisées principalement par les standards POSIX, plus un bon nombre d'extensions, dont certaines spécifiques à d'autres UNIX ou au projet GNU.
- La `glibc` ne supporte que des processeurs qui ont une Memory Management Unit (MMU), ce qui n'est pas le cas du processeur Blackfin. La librairie `uClibc` est utilisée à la place de la `glibc` sur le processeur Blackfin, avec toutefois quelques limitations importantes.

# Commandes de base: GNU / BusyBox

## GNU: ls -l /bin/

```
-rwxr-xr-x 1 root root 4648 2008-01-02 12:35 arch
-rwxr-xr-x 1 root root 769368 2006-12-11 23:28 bash
-rwxr-xr-x 3 root root 28880 2006-08-26 22:54 bunzip2
-rwxr-xr-x 1 root root 519432 2007-01-21 16:57 busybox
-rwxr-xr-x 3 root root 28880 2006-08-26 22:54 bzipcat
-rwxr-xr-x 2 root root 2105 2006-08-26 22:54 bzipcmp
-rwxr-xr-x 2 root root 2105 2006-08-26 22:54 bzdiff
-rwxr-xr-x 3 root root 3642 2006-08-26 22:54 bzegrep
-rwxr-xr-x 1 root root 4878 2006-08-26 22:54 bzexe
-rwxr-xr-x 3 root root 3642 2006-08-26 22:54 bzfgrep
-rwxr-xr-x 3 root root 3642 2006-08-26 22:54 bzgrep
-rwxr-xr-x 3 root root 28880 2006-08-26 22:54 bzip2
-rwxr-xr-x 1 root root 10728 2006-08-26 22:54 bzip2recover
-rwxr-xr-x 2 root root 1297 2006-08-26 22:54 bzless
-rwxr-xr-x 2 root root 1297 2006-08-26 22:54 bzipmore
-rwxr-xr-x 1 root root 21448 2007-01-30 21:38 cat
-rwxr-xr-x 1 root root 39736 2007-01-30 21:38 chgrp
-rwxr-xr-x 1 root root 36072 2007-01-30 21:38 chmod
-rwxr-xr-x 1 root root 41592 2007-01-30 21:38 chown
-rwxr-xr-x 1 root root 64608 2007-01-30 21:38 cp
-rwxr-xr-x 1 root root 101520 2008-05-01 08:07 cpio
lrwxrwxrwx 1 root root 21 2007-05-31 13:50 csh -> /etc/alternatives/csh
-rwxr-xr-x 1 root root 52032 2007-01-30 21:38 date
-rwxr-xr-x 1 root root 40384 2007-01-30 21:38 dd
-rwxr-xr-x 1 root root 37616 2007-01-30 21:38 df
-rwxr-xr-x 1 root root 85536 2007-01-30 21:38 dir
-rwxr-xr-x 1 root root 7056 2008-01-02 12:35 dmesg
-rwxr-xr-x 1 root root 11560 2006-11-23 21:35 dnsdomainname
-rwxr-xr-x 1 root root 19304 2007-01-30 21:38 echo
-rwxr-xr-x 1 root root 53456 2006-03-16 13:44 ed
-rwxr-xr-x 1 root root 33 2006-10-10 12:46 egrep
-rwxr-xr-x 1 root root 15560 2007-01-30 21:38 false
-rwxr-xr-x 1 root root 7232 2006-09-05 18:37 fgconsole
-rwxr-xr-x 1 root root 33 2006-10-10 12:46 fgrep
-rwxr-xr-x 1 root root 23760 2006-08-09 14:04 fuser
-rwxr-xr-x 1 root root 101896 2006-10-10 12:47 grep
-rwxr-xr-x 4 root root 60216 2006-09-19 23:41 gunzip
-rwxr-xr-x 1 root root 4895 2006-09-19 23:41 gzexe
-rwxr-xr-x 4 root root 60216 2006-09-19 23:41 gzip
-rwxr-xr-x 1 root root 11560 2006-11-23 21:35 hostname
-rwxr-xr-x 1 root root 154848 2006-12-14 20:21 ip
-rwxr-xr-x 1 root root 14952 2006-09-13 04:23 kill
-rwxr-xr-x 1 root root 27632 2007-01-30 21:38 ln
-rwxr-xr-x 1 root root 40584 2006-09-05 18:37 loadkeys
-rwxr-xr-x 1 root root 37712 2007-02-27 20:13 login
-rwxr-xr-x 1 root root 85536 2007-01-30 21:38 ls
```

## BusyBox: ls -l /bin/

```
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 [ -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 [[ -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 awk -> busybox
-rwxr-xr-x 1 root root 3856 2008-06-26 15:01 basename
-rwxr-xr-x 1 root root 4928 2008-06-26 15:01 bcdb
-rwxr-xr-x 1 root root 75684 2008-06-26 13:47 boa
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 bunzip2 -> busybox
-rwxr-xr-x 1 root root 293628 2008-06-26 15:01 busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 bzipcat -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 cat -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 chgrp -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 chmod -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 chown -> busybox
lrwxrwxrwx 1 root root 14 2008-06-26 10:56 chroot -> ../bin/busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 clear -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 cmp -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 cp -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 cpio -> busybox
lrwxrwxrwx 1 root root 14 2008-06-26 10:56 crond -> ../bin/busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 crontab -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 cut -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 date -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 dd -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 df -> busybox
-rwxr-xr-x 1 root root 40528 2008-06-26 15:01 dhcpcd
-rwxr-xr-x 1 root root 14600 2008-06-26 15:01 dhrystone
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 diff -> busybox
-rwxr-xr-x 1 root root 3752 2008-06-26 15:01 dirname
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 dmesg -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 du -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 echo -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 env -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 expr -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 false -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 find -> busybox
-rwxr-xr-x 1 root root 10256 2008-06-26 15:01 flash_eraseall
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 free -> busybox
-rwxr-xr-x 1 root root 69504 2008-06-26 15:01 ftp
-rwxr-xr-x 1 root root 49192 2008-06-26 15:01 ftpd
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 ftpget -> busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 ftpput -> busybox
-rwxr-xr-x 1 root root 50380 2008-06-26 15:01 gdbserver
-rwxr-xr-x 1 root root 5092 2008-06-26 13:47 gen-keys
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 getopt -> busybox
lrwxrwxrwx 1 root root 14 2008-06-26 10:56 getty -> ../bin/busybox
lrwxrwxrwx 1 root root 7 2008-06-26 10:56 grep -> busybox
```

## L'interpréteur de commande Busybox

- Un système UNIX est composé de nombreuses petites applications simples invoquées depuis un interpréteur de commande. Le projet GNU implémente un très large choix de ces applications et propose l'interpréteur de commande Bourne Again SHell (`bash`). Ces applications n'ont pas été optimisées pour des processeurs sans MMU.
- Le projet BusyBox est utilisé à la place de nombreuse application GNU de base sur le processeur Blackfin. BusyBox propose une implémentation minimaliste pour processeur sans MMU des applications de base essentielles et d'un interpréteur de commande rudimentaire (`ash`).
- Un aspect intéressant de BusyBox est qu'il permet de choisir pour chaque commande si l'on veut l'implémentation de BusyBox ou l'implémentation d'une application externe.
- Remarque: Le projet Berkeley System Distribution (BSD) propose aussi bon nombre d'applications de base qui sont parfois préférées à celles du projet GNU.

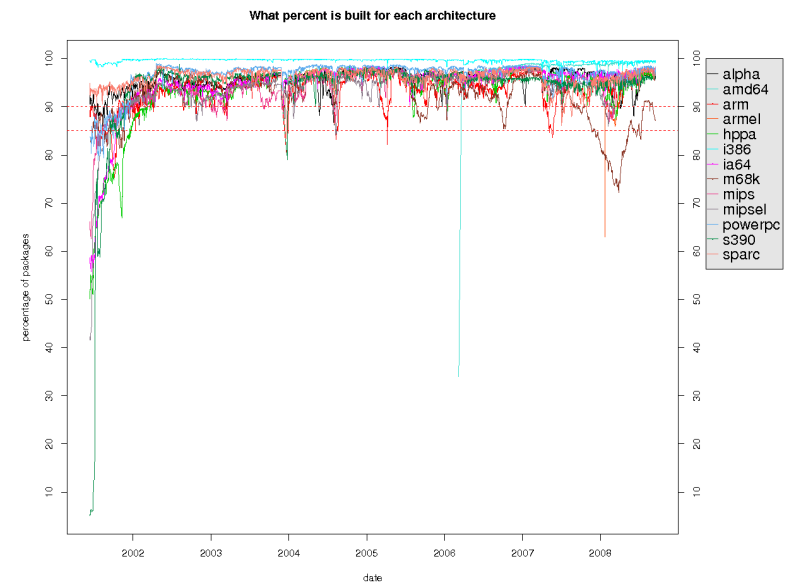
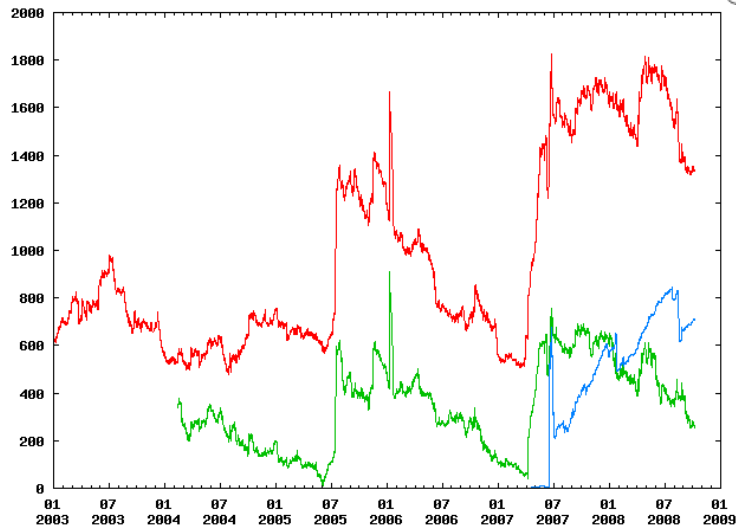
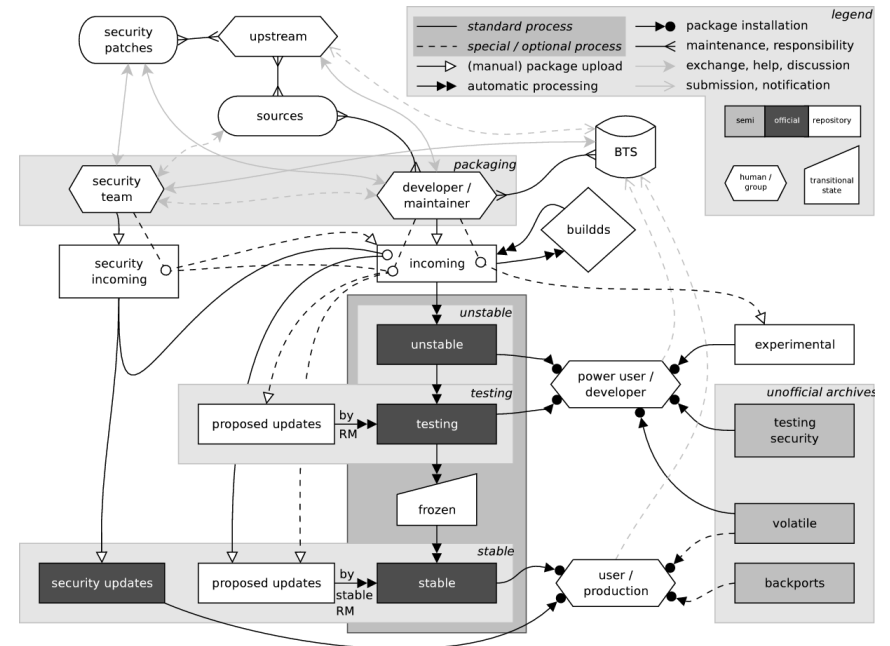
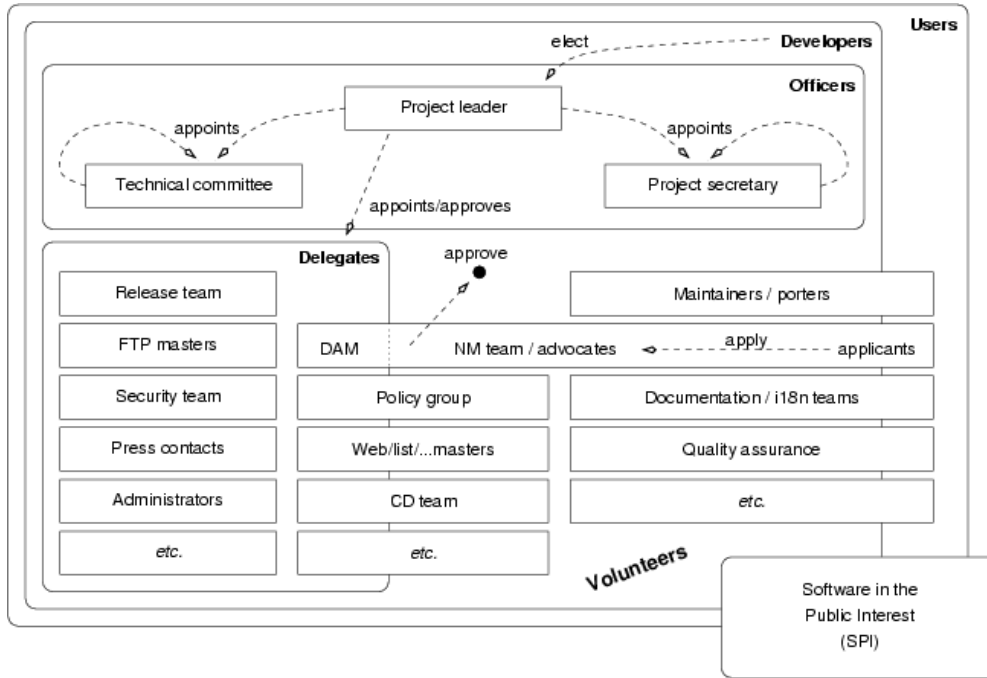
## Définitions: une distribution

- Une distribution permet d'installer simplement sur un ordinateur un noyau Linux, les bibliothèques et les applications de façon à être rapidement utilisable. Une distribution permet d'installer facilement des paquetages pour ajouter des applications et les maintenir à jour.
- Il existe un nombre incalculable de distributions basées sur Linux, et il y en a chaque jour plusieurs en plus ! Voici les distributions communément considérées comme majeures:
  - Ubuntu: basée sur Debian GNU/Linux.
  - Debian GNU/Linux: la distribution la plus générique.
  - Fedora: sur lequel est basée RedHat.
  - OpenSUSE: sur lequel est basée SuSE de Novell.
  - Mandriva: populaire en France.



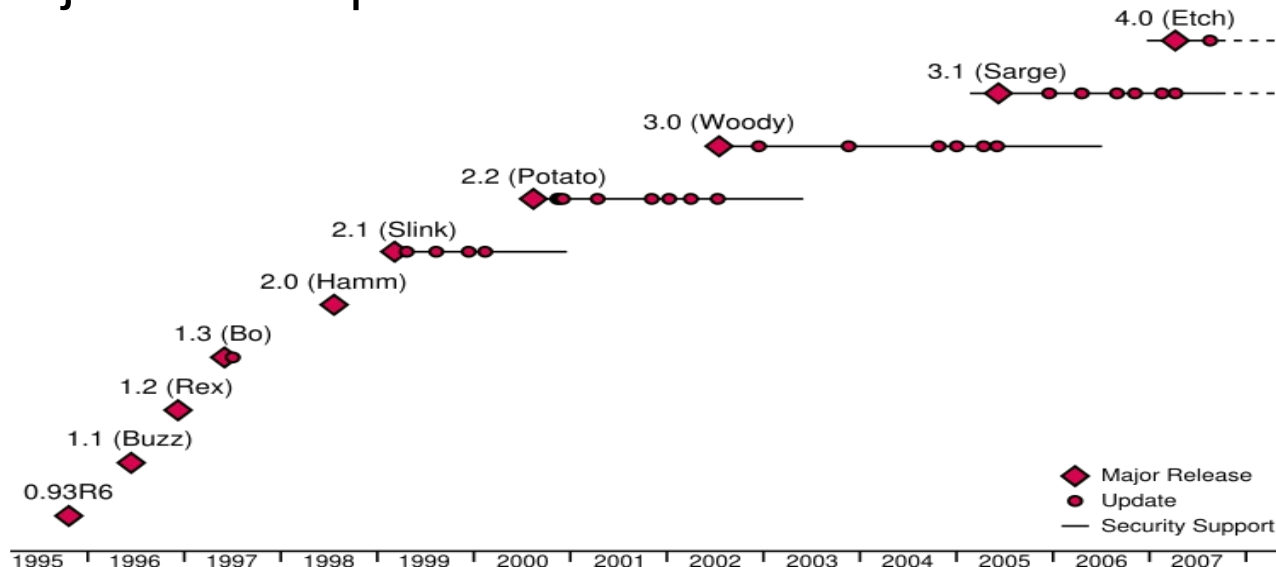
etc...

# La distribution Debian GNU/Linux



# La distribution de référence: Debian GNU/Linux

- Debian est la plus grande, la plus libre et la plus générique, des distributions Linux.
- Debian a été créé dans le milieu des années 1990 par un couple (Debra et Ian). Actuellement, Debian est un cas unique de communauté virtuelle planétaire, avec une constitution, des élections et des référendums, le tout gérés publiquement et démocratiquement.
- L'expertise légale de Debian pour défendre les licences Open Source Software (OSS) a permis la résolution de nombreux conflits importants entre des projets interdépendants.

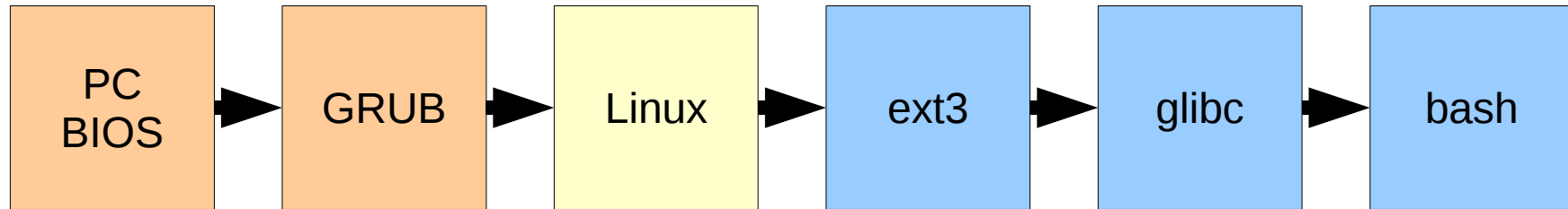


## Pourquoi choisir Debian GNU/Linux

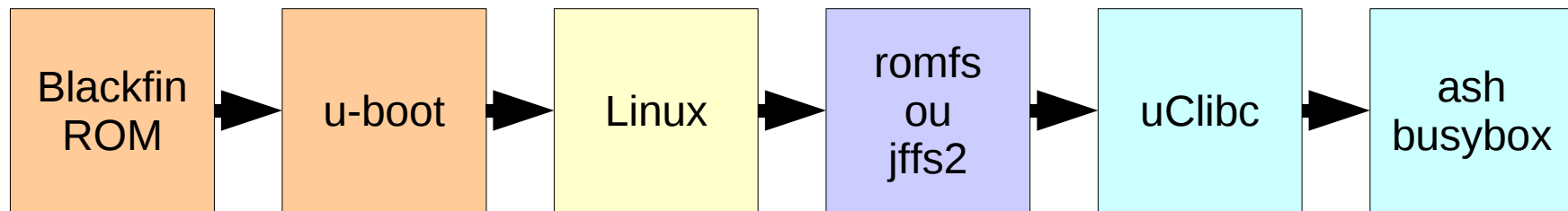
- La seule distribution majeure dirigée par une communauté libre.
  - Pas de dictat d'une entreprise commerciale comme pour Fedora (Redhat) ou OpenSuse (Novel).
- C'est la distribution avec le plus grand choix de paquetage.
- Le système de paquetage Debian a clairement démontré sa supériorité.
- Garantie que les sources distribués dans la section 'main' sont vraiment sous une licence libre approuvée comme tel par des experts.
- La distribution la plus générique.
  - La seule distribution majeure avec un dizaine d'architectures.
- Système de qualité ouvert permettant de facilement trouver des solutions.
  - Le passage d'une version à une autre est abondamment testé.

## Comparaison système Debian / Blackfin

- Démarrage d'un système Debian GNU/Linux:



- Démarrage d'un système Blackfin Linux:



- On observe que le noyau Linux est l'unique code en commun!
- Ces systèmes sont toutefois largement compatibles du point de vue de la programmation des applications utilisateurs.

## Le site blackfin.uclinux.org

- Le site blackfin.uclinux.org réunit les services permettant le développement et le support du port de Linux sur processeur Blackfin. On y trouve principalement:
- Une documentation Wiki couvrant de très nombreux aspects.
- Les archives des codes source:
  - `toolchain`
  - `u-boot`
  - `linux-kernel`
  - `uclinux-dist`
- Les rapports de problèmes et les modifications proposées pour chaque archives.
- Des forums et des listes de diffusions pour chaque archives.
- D'autres projets matériel ou logiciel, relatifs au Blackfin.

# blackfin.uclinux.org

The screenshot shows the homepage of blackfin.uclinux.org. At the top, there is a navigation bar with links for Home, My Stuff, Users, Search, Projects, and Snippets. Below the navigation bar, the main content area features a heading "Welcome to the Blackfin Open Source Koop" followed by a paragraph describing the site's focus on open source and free software for Analog Devices Blackfin processors. A section titled "What is a Koop?" explains the cooperative nature of the project. Another section, "Site Feedback and Participation", encourages user registration. A list of main projects is provided, with the first three items (GNU Toolchain, Linux Kernel, and uClinux Distribution) highlighted by a red box. A red arrow points from the text "Les archives essentielles" to this red box. Below the project list, a pink box highlights the "Documentation Wiki" link, with a pink arrow pointing from the text "A consulter absolument" to it. The page also includes sections for "Documentation" and "Support".

Linux™  
BLACKfin™

Home My Stuff Users Search Projects Snippets

Home »

## Welcome to the Blackfin Open Source Koop

This site is the central location for open source and free software and hardware projects targeted for use with certain members of the Analog Devices Blackfin processor family. In addition to a wide range of applications, this Koop also focuses on supporting Open Source hardware and software tools, including the GNU GCC toolchain and the uClinux distribution.

### What is a Koop?

The word "Koop" is derived from the English word cooperative, which typically describes a jointly owned enterprise that produces and distributes goods and services and is run for the benefit of its owners. In this situation, it takes nothing to be an "owner", except the understanding/acknowledgement that you get out, what you put in — that is, sharing and cooperation. While the Koop does include a small dedicated team of contributors (some volunteer, some Analog Devices employees), everything is done for the betterment of the entire Koop - collaborating with others (including you) for a common purpose and benefit. What this means is that you (and we) must be willing to adjust to differences in opinion and implementation in order to obtain agreement for the Koop's greater good.

The blackfin.uclinux.org website is a free (zero cost) service to developers and users of Open Source Software, offering easy access to the best in source code, mailing lists, bug tracking, message boards/forums, software roadmaps and releases of Open Source software, as it relates to the Blackfin processor.

### Site Feedback and Participation

In order to get the most out of this site, you'll need to register as a new user. This will allow you to participate fully in all we have to offer. You may of course browse the site without registering, but will not have access to participate fully.

Here is a list of the main projects hosted on this wonderful site.

- [GNU Toolchain](#) ( [Releases](#), [SVN](#), [GIT](#), [Snapshots](#), [Help](#), [Lists](#), [Bugs](#), [ToDo](#) )
- [Linux Kernel](#) ( [Releases](#), [SVN](#), [GIT](#), [Snapshots](#), [Help](#), [Lists](#), [Bugs](#), [ToDo](#) )
- [uClinux Distribution](#) ( [Releases](#), [SVN](#), [Snapshots](#), [Presentations](#), [Help](#), [Lists](#), [Bugs](#), [ToDo](#) )
- [Das U-boot \(Bootloader\)](#) ( [Releases](#), [SVN](#), [GIT](#), [Help](#), [Lists](#), [Bugs](#) )
- [uClinux \(Run Linux on windows\)](#) ( [Releases](#), [SVN](#), [GIT](#), [Help](#) )
- [Hardware Projects and Schematics](#) ( [Releases](#), [Purchase](#), [Help](#) )
- [Documentation](#) ( [Releases](#) )

For the full list of projects hosted here, please use the [Projects](#) link at the top of the page.

### Documentation

Pretty much all of our documentation lives in our [Documentation Wiki](#). You can find [snapshots](#) of the whole site which is useful for offline browsing.

### Support

The Blackfin Koop is supported by both an active open source community as well as a dedicated team from Analog Devices Inc.

Les archives essentielles

A consulter absolument

## Quelques spécificités du projet Blackfin

- Le compilateur ajoute systématiquement un underscore devant chaque fonction. Ce comportement est obsolète mais les mainteneurs ne veulent pas changer ce qu'il faut dans l'assembleur. Cela crée de nombreuses difficultés dans la création des bibliothèques dynamiques.
- L'archive uclinux-dist est gigantesque car elle est composée d'une centaine de projets. Cela rend toutes les opérations très lentes. La maintenance est chaotique: la plupart des projets ne sont pas à jour.
- La configuration du noyau et des applications passe par un troisième programme de configuration qui rend les manipulations laborieuses et non intuitives.
- La bibliothèque DSP de VDSP++ a été partiellement portée dans la chaîne de construction gcc. Elle permet de simplifier la portage d'algorithmes existants, mais se limite pour l'instant principalement aux nombres complexes fractionnaires de 16 bits.

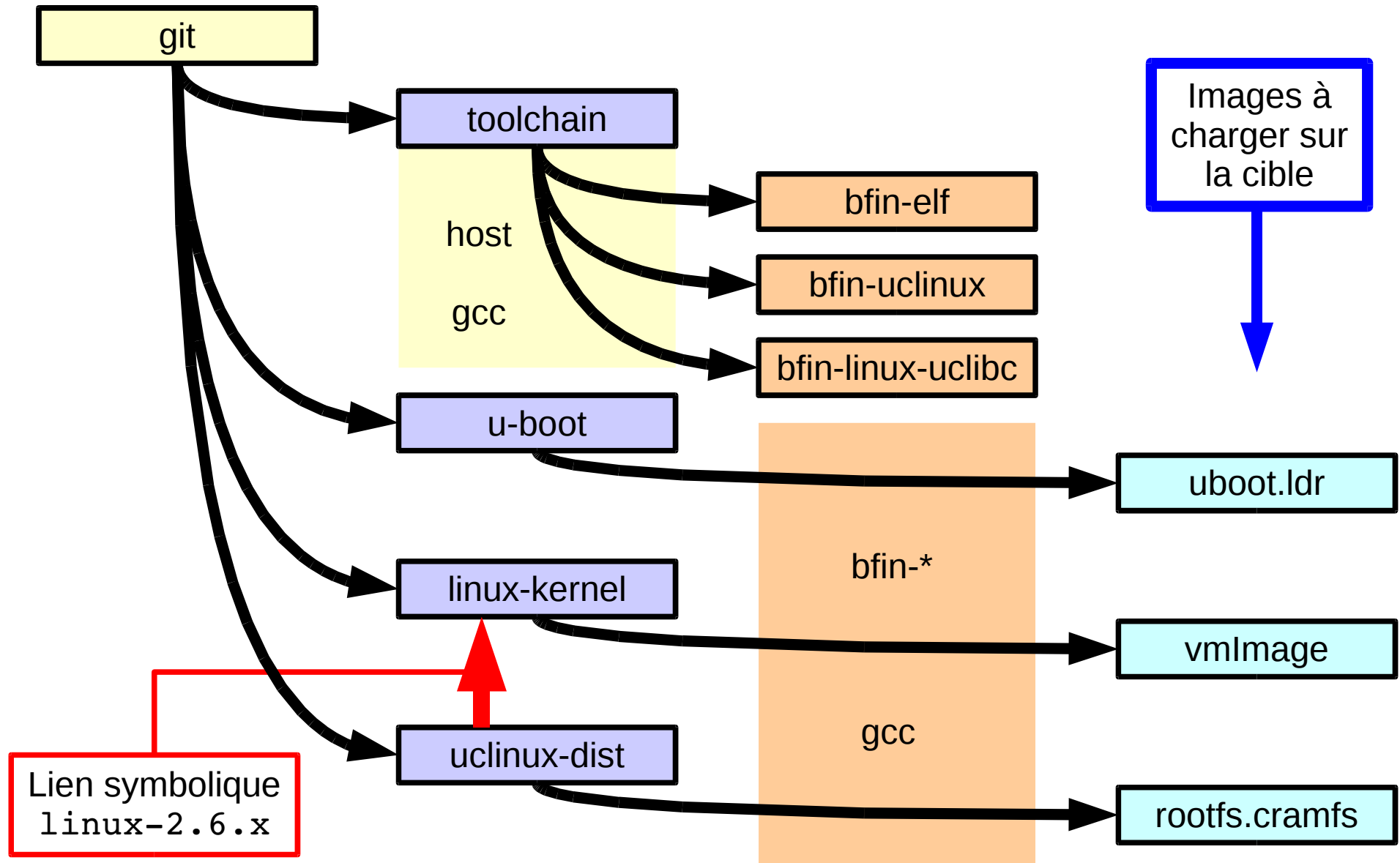
## Quelques spécificités du projet Blackfin

- Par défaut les objets sont compilés au format FLAT à la place du format ELF comme toutes les autres distributions. Cela rend impossible la compilation de nombreuses applications et librairie. Le format FLAT duplique les librairies pour chaque applications qui les utilisent ce qui gaspille de la place dans le système de fichier. Il existe des dérivés du format FLAT pour tenter d'éviter ces problèmes, mais ils ne fonctionnent pas bien. Utiliser le format ELF (FDPIC): il est standard; il fonctionne avec toutes les applications et librairies; et il n'est pas si lent à charger pour justifier le format FLAT.
- Pas de MMU. Cela signifie que certaines applications ou librairies ne sont pas supportées. Dans beaucoup de cas il est possible de les modifier pour contourner le problème, mais quelques cas n'ont pas de solution simple et exige un nouveau design adapté. Une des conséquences les plus visibles est l'impossibilité d'avoir une gestion satisfaisante des tâches avec le shell.

## Que définit le terme uClinux ?

- uClinux != un noyau Linux
  - Historiquement, cela n'était pas le cas au début du projet uClinux. Il fallait un noyau Linux spécialement modifié, appelé justement uClinux, pour supporter les processeurs sans MMU. Au début des années 2000, une nouvelle approche a été testée: inclure les architectures sans MMU (nommu) directement dans le noyau Linux. Après un laborieux travail, le support 'nommu' a été accepté dans le noyau Linux, suivit petit à petit par les architectures sans MMU. De fait, il n'existe plus de noyau uClinux.
- uClinux == distribution Linux basé sur la uClibc.
  - Le nom uClinux réfère actuellement à des distributions utilisant la librairie `uClibc` à la place de la librairie `glibc`.
- uClinux == site web [uclinux.org](http://uclinux.org)
  - Initialement le site de la communauté des développeurs du noyau uClinux. Il n'a plus cette utilité depuis le support nommu dans le noyau Linux. Il n'est plus que un pointeur sur d'autres sites d'architectures embarquées qui sont parfois un sous domaine, comme le Blackfin avec son site [blackfin.uclinux.org](http://blackfin.uclinux.org).

# Construction



## Installation de la chaîne de construction

- La création d'un firmware (noyau + système de fichier) exige un système Linux. Le plus simple est d'installer un PC avec une distribution Linux tel que Debian. Il existe toutefois deux autres possibilités: la virtualisation (par exemple avec VmWare Server) ou coLinux.
- La distribution Linux doit inclure un certain nombre de paquetages tel que la suite gcc pour une compilation native; make; yacc; bison; ncurses-dev; zlib1g-def; etc...
- La disponibilité des ces éléments est généralement vérifiée par le script qui fabrique la chaîne de construction mais il y a parfois des oublis.
- Remarque sur les chaînes distribuées en binaires:
  - Le site [blackfin.uclinux.org](http://blackfin.uclinux.org) distribue aussi des versions binaires de la chaîne de construction, mais cela peut être problématique:
    - Pas facile d'installer correctement les composants.
    - Pas compatible avec votre distribution (64 bits p.ex).
    - Pas de code source pour comprendre ce qui se passe.
    - Pas d'évolution au rythme des développements.

## Installation de la chaîne de construction

- Les archives `toolchain`, `u-boot`, `linux-kernel` et `uclinux-dist` doivent être chargées depuis le site `blackfin.uclinux.org` avec la commande `git clone`. Techniquement le site `blackfin.uclinux.org` utilise encore `subversion` en interne, mais cela va probablement disparaître un jour. Il est vivement recommandé de directement commencer avec `git`.
- Le script de fabrication de la chaîne de construction est invoqué avec la commande `toolchain/buildscript/BuildToolChain`.
- Le résultat de la fabrication se trouve dans les dossiers
  - `bfin-elf`
  - `bfin-uclinux`
  - `bfin-linux-uclibc`
- Chacun des dossiers correspond à un compilateur pour un type d'exécutable différent. Le compilateur `bfin-linux-uclibc` est celui qui produit les exécutables au format ELF qui est la norme sous UNIX.

## Utilisation de la chaîne de construction

- Important: l'interpréteur de commande courant doit être configuré pour trouver les outils de cross-compilation Blackfin dans sa variable de recherche d'exécutable `PATH`. Le plus simple est d'écrire un script `setarch` comme ceci et de l'invoquer avec la commande `source setarch`:

```
export PATH=$PATH:`pwd -P`/bfin-uclinux/bin
export PATH=$PATH:`pwd -P`/bfin-linux-uclibc/bin
export PATH=$PATH:`pwd -P`/bfin-elf/bin
export STAGING=`pwd -P`/uclinux-dist/staging
export ROMFS=`pwd -P`/uclinux-dist/romfs
export STRIP=bfin-linux-uclibc-strip
unset LANG
```

- Ne pas fixer la langue, cela peut planter le compilateur!
- Attention: chaque instance d'interpréteur de commande doit être configuré. Il est possible d'invoquer la configuration depuis le script par défaut de l'interpréteur, mais cela peut causer des problèmes dans le cas où plusieurs cross-compilateurs sont installés.

## L'archive u-boot

- L'archive `u-boot` contient le dossier `u-boot-1.1.6` avec le code source du projet. Un certain nombre de boards sont directement supportés avec la commande `make xxxx_config`. Pour un board plus spécifique le plus simple est généralement d'adapter le fichier de configuration du board le plus ressemblant.
- Seules les boards des kits de développement de Analog Devices sont régulièrement testés
- La programmation de l'image `u-boot` sur la cible peut être faite:
  - Avec une interface JTAG, directement en flash ou en SDRAM.
  - A part les premiers processeurs de la famille Blackfin, il est aussi possible de charger `u-boot` en SDRAM via l'interface UART
- Dans le cas du chargement de `u-boot` en SDRAM, son exécution permet de charger via Ethernet+TFTP ou UART un `u-boot` définitif pour le programmer dans une mémoire flash.

# linux-kernel

```
drwxr-xr-x 22 jcdr jcdr 4096 Oct 7 23:30 .
drwxr-xr-x 20 jcdr jcdr 4096 Sep 27 15:44 ..
-rw-r--r-- 1 jcdr jcdr 62614 Oct 7 23:30 .config
drwxr-xr-x 8 jcdr jcdr 4096 Sep 25 13:37 .git
-rw-r--r-- 1 jcdr jcdr 867 Sep 25 13:37 .gitignore
-rw-r--r-- 1 jcdr jcdr 18693 Sep 25 13:37 COPYING
-rw-r--r-- 1 jcdr jcdr 89517 Sep 25 13:37 CREDITS
drwxr-xr-x 79 jcdr jcdr 12288 Sep 25 13:37 Documentation
-rw-r--r-- 1 jcdr jcdr 2465 Sep 25 13:37 Kbuild
-rw-r--r-- 1 jcdr jcdr 100983 Sep 25 13:37 MAINTAINERS
-rwxr-xr-x 1 jcdr jcdr 2468 Sep 25 13:37 MAKEALL
-rw-r--r-- 1 jcdr jcdr 55081 Sep 25 13:37 Makefile
-rw-r--r-- 1 jcdr jcdr 16930 Sep 25 13:37 README
-rw-r--r-- 1 jcdr jcdr 3172 Sep 25 13:37 REPORTING-BUGS
drwxr-xr-x 26 jcdr jcdr 4096 Sep 25 13:37 arch
drwxr-xr-x 2 jcdr jcdr 4096 Sep 25 13:37 block
drwxr-xr-x 3 jcdr jcdr 4096 Sep 25 13:37 crypto
drwxr-xr-x 79 jcdr jcdr 4096 Sep 25 13:37 drivers
drwxr-xr-x 63 jcdr jcdr 4096 Sep 25 13:37 fs
drwxr-xr-x 46 jcdr jcdr 4096 Oct 7 23:25 include
drwxr-xr-x 2 jcdr jcdr 4096 Sep 25 13:37 init
drwxr-xr-x 2 jcdr jcdr 4096 Sep 25 13:37 ipc
drwxr-xr-x 5 jcdr jcdr 4096 Sep 25 13:37 kernel
drwxr-xr-x 6 jcdr jcdr 4096 Sep 25 13:37 lib
-rw-r--r-- 1 jcdr jcdr 16 Sep 25 13:37 localversion-adi
drwxr-xr-x 2 jcdr jcdr 4096 Sep 25 13:37 mm
drwxr-xr-x 43 jcdr jcdr 4096 Sep 25 13:37 net
drwxr-xr-x 6 jcdr jcdr 4096 Sep 25 13:37 samples
drwxr-xr-x 9 jcdr jcdr 4096 Sep 25 13:37 scripts
drwxr-xr-x 5 jcdr jcdr 4096 Sep 25 13:37 security
drwxr-xr-x 21 jcdr jcdr 4096 Sep 25 13:37 sound
drwxr-xr-x 2 jcdr jcdr 4096 Sep 25 13:37 usr
drwxr-xr-x 3 jcdr jcdr 4096 Sep 25 13:37 virt
```

```
General setup --->
[*] Enable loadable module support --->
-* Enable the block layer --->
    Preemption Model (Preemptible Kernel (Low-Latency Desktop)) --->
[ ] Preemptible RCU (NEW)
    Blackfin Processor Options --->
    Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
    Executable file formats --->
    Power management options --->
    CPU Frequency scaling --->
    Networking --->
    Device Drivers --->
    File systems --->
    Kernel hacking --->
    Security options --->
-* Cryptographic API --->
    Library routines --->
---
    Load an Alternate Configuration File
    Save an Alternate Configuration File
```

```
Processor and Board Settings
CPU (BF533) --->
Silicon Rev (0.3) --->
System type (BF533-STAMP) --->
BF533/2/1 Specific Configuration --->
Board customizations --->
Blackfin Kernel Optimizations --->
Kernel executes from (RAM) --->
Memory model (Flat Memory) --->
[*] 64 bit Memory and IO resources (EXPERIMENTAL)
< > Enable Blackfin General Purpose Timers API (NEW)
[*] Enable DMA Support (NEW)
    Uncached SDRAM region (Enable 1M DMA region) --->
    *** Cache Support ***
[ ] Enable ICACHE (NEW)
[ ] Enable DCACHE (NEW)
[ ] Enable Instruction Cache Locking (NEW)
[ ] Enable the memory protection unit (EXPERIMENTAL) (NEW)
    *** Asynchronous Memory Configuration ***
EBIU_AMGCTL Global Control --->
EBIU_AMBCTL Control --->
```

## L'archive linux-kernel

- L'archive `linux-kernel` contient le code source du noyau Linux. Il est possible de configurer et compiler le noyau directement depuis ce dossier avec les commandes `'make menuconfig'` et `'make'`. Mais attention: les opérations effectuées depuis l'archive `uclinux-dist` peuvent affecter la configuration du noyau Linux ou provoquer la concaténation d'un système de fichier `initrd` à l'image du noyau.
- Utiliser la commande `make` depuis ce dossier permet de générer plus rapidement un noyau, mais demande plus d'expérience.

# uclinux-dist

```

drwxrwxrwx 14 jcdr jcdr 4096 Sep 19 22:31 .
drwxr-xr-x 12 jcdr jcdr 4096 Sep 25 13:10 ..
-rw-r--r-- 1 jcdr jcdr 3357 Sep 19 22:31 .config
drwxrwxrwx 8 jcdr jcdr 4096 Sep 19 22:31 .git
-rw-r--r-- 1 jcdr jcdr 18007 May 7 16:26 COPYING
drwxr-xr-x 3 jcdr jcdr 4096 May 7 16:26 Documentation
-rwxr-xr-x 1 jcdr jcdr 1477 May 7 16:26 MAKEALL
-rw-r--r-- 1 jcdr jcdr 14186 Aug 21 13:44 Makefile
-rw-r--r-- 1 jcdr jcdr 5193 May 7 16:26 README
-rw-r--r-- 1 jcdr jcdr 1743 May 7 16:26 SOURCE
drwxr-xr-x 11 jcdr jcdr 4096 May 7 16:26 bfin_patch
drwxr-xr-x 2 jcdr jcdr 4096 May 7 16:26 bin
drwxr-xr-x 3 jcdr jcdr 4096 Sep 19 22:31 config
lrwxrwxrwx 1 jcdr jcdr 45 Aug 21 15:47 config.arch -> vendors/AnalogDevices/BF533-EZKIT/config.arch
drwxr-xr-x 11 jcdr jcdr 4096 May 7 16:26 freeswan
drwxr-xr-x 87 jcdr jcdr 4096 Jun 16 11:26 lib
lrwxrwxrwx 1 jcdr jcdr 15 Aug 21 15:47 linux-2.6.x -> ../linux-kernel
drwxr-xr-x 13 jcdr jcdr 4096 May 7 16:28 openswan
-rw-r--r-- 1 jcdr jcdr 16566 May 7 16:28 release_notes
drwxr-xr-x 89 jcdr jcdr 4096 Jun 16 11:26 testsuites
drwxr-xr-x 6 jcdr jcdr 4096 Sep 19 22:31 tools
drwxr-xr-x 262 jcdr jcdr 12288 Sep 19 22:31 user
drwxr-xr-x 68 jcdr jcdr 4096 Aug 26 14:18 vendors

```

```

Library Configuration -->
Core Applications --->
Flash Tools --->
Filesystem Applications --->
Network Applications --->
Miscellaneous Applications --->
BusyBox --->
Tinylogin --->
MicroWindows --->
Games --->
Miscellaneous Configuration --->
Debug Builds --->
Blackfin test programs --->
Blackfin app programs --->
Blackfin canned demos --->
Blackfin device drivers (Experimental) --->
Blackfin build options --->

---
Load an Alternate Configuration File
Save an Alternate Configuration File

```

COPYING	bonnie++	crypto-tools	ethattach	gdbserver	iproute2	loattach	nsntp	openswan	proftpd	ser	stunnel	unrar
COPYING.LIB	boottools	ctorrent	etherwake	genext2fs	iptables	login	ntd-utils	openvpn	python	sethdhc	sysutils	unzip
DISTRIBUTION	bpalgin	cxctest	ethtool	gettyd	iputils	lpr	musicbox	p7zip	qspitest	setmac	tcpblast	unzoo
DirectFB-examples	br2684ctl	dagrab	ez-ipupdate	grep	iso_client	lrpstat	mysql	palmbot	quagga	setserial	tcpdump	user.mak
Kconfig	brcfg	dnns3-client	fakeidentd	hd	jffs-tools	lrzsz	nano	pam_ldap	radauth	sh	tcpwrappers	vconfig
Kconfig.local	bridge-utils	de2ts-cal	fconfig	horch	jpegview	lua	nbench	pam_radius	radvd	shadow	tcsh	version
Makefile	busybox	demo	fdisk	hostap	kendin-config	madplay	nessus	pam_smb	ramimage	shutils	telnet	vixie-cron
Makefile.local	cal	dhcp-isc	fileutils	httpd	klaxon	mail	net-snmp	pam_tacplus	rdate	siproxd	telnetd	vplay
a60	camserv	dhcpcd	flashw	httptunnel	l2tpd	mathtest	net-tools	pciutils	readprofile	slattach	tftp	vpnled
agetty	can4linux	dhcpcd-new	flatfsd	hwclock	l7-protocols	mawk	netcat	pcmcia-cs	recover	smbmount	tftpd	w3cam
appWeb	chi_client	dhcpl	flthdr	i2c-tools	lcd	mbus	netflash	perl	reiserfsprogs	smtpclient	threaddemos	wget
arj	cgi_generic	dhystone	fnord	ifattach	ledcon	mgetty	netperf	ping	rootloader	snmpd	thttpd	wheatstone
asterisk	cgihlml	diald	freeradius	inetd	levee	microwin	netstat-nat	play	routed	snort	tinylogin	wireless_tools
at	cksum	directvnc	freeswan	init	lha	mii-tool-fec	nooom	playrt	rp-pppoe	speedtouch	tingtcl	wlan-ng
bash	clamav	discard	frob-led	ip-sentinel	lilo	mini_httpd	ntp	plug	rrdtool	squashfs	tip	yaffs-utils
bind	clawsmtpl	dnsmasq	frox	ipchains	links	mount	ntpclient	pop3proxy	rsync	squid	tpt	zebra
blkfin-apps	clock	dnsmasq2	fswcert	iperf	linphone	mp3play	null	portnmap	rsyslogd	ssh	traceroute	
blkfin-drvs	contrack	dosfstools	ftp	ipfwadm	linux-igd	mpc	nsh	pppd	rtc-ds1302	sslwrap	tripwire	
blkfin-test	cpu	dropbear	ftpd	ipmasqadm	lirc	mpc2x	oggplay	pptp-client	rtc-41t11	stp	udev	
bluez-utils	cramfs	e2fsprogs	games	ipportfw	lissa	mpd	openmpd	samba	samba	strace	ulogd	
boa	cron	elvis-tiny	gdb	ipredir	lm_sensors	nsh	openssl	procs	sash	stty	ulogd2	

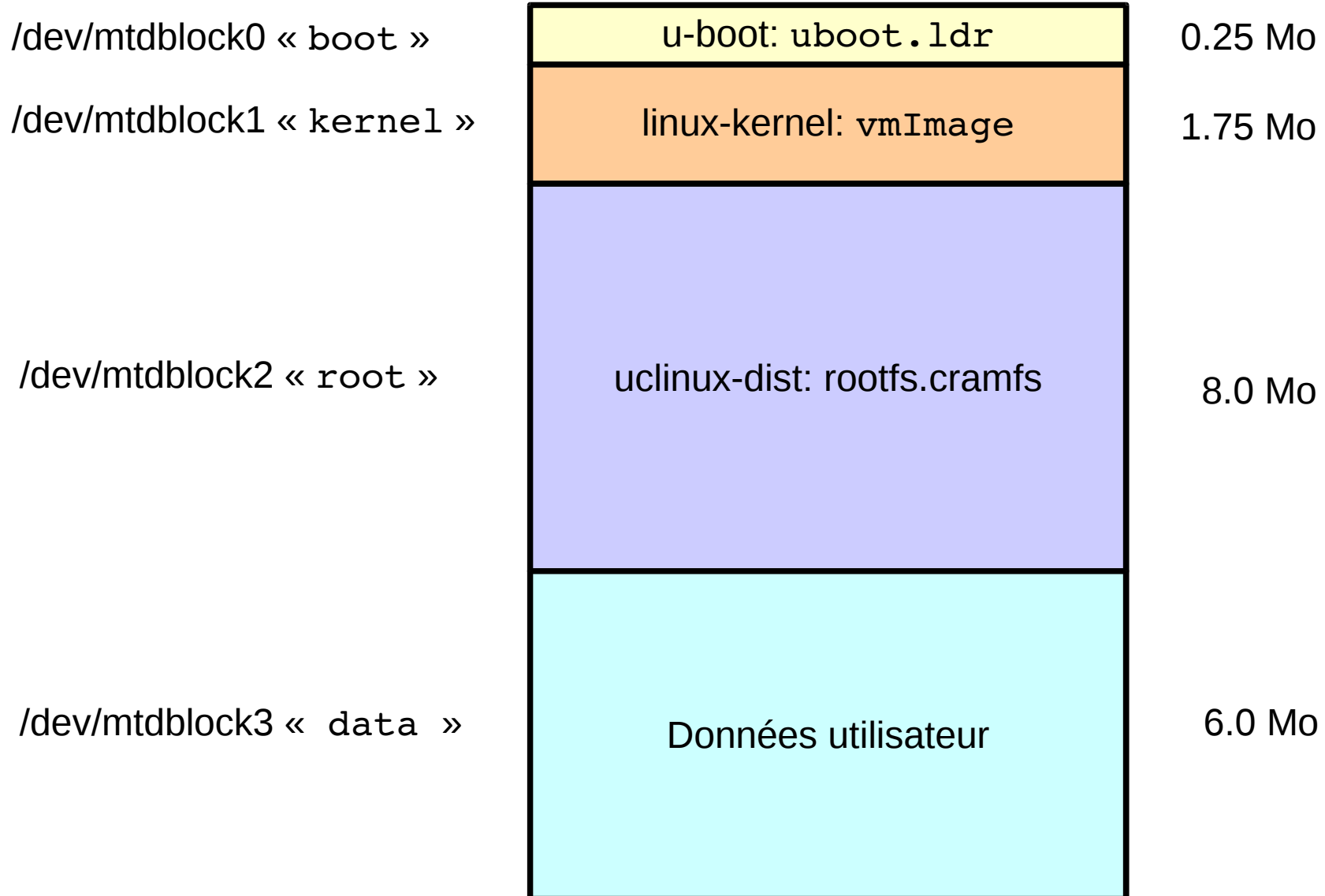
## L'archive uclinux-dist

- L'archive `uclinux-dist` contient les codes sources de nombreuses bibliothèques et applications, ainsi que les scripts permettant de générer des images de différents type de systèmes de fichier racine.
- Pour être fonctionnelle, cette archive doit absolument contenir un lien `linux-2.6.x` sur l'archive `linux-kernel`. Avec ce lien, les commandes `'make menuconfig'` et `'make'` vont configurer et compiler le noyau Linux et le système de fichier racine. C'est la méthode la plus simple pour débuter, bien que lente.
- Les images résultantes sont dans le dossier `images` et dépendent de la configuration choisie. L'image du noyau Linux est généralement séparée de l'image du système de fichier racine, sauf dans le cas `initrd`.

## Image initrd ou système de fichier ?

- Une image `initrd` est la concaténation du noyau Linux et du système de fichier racine dans un seul gros fichier. C'est pratique pour tester rapidement un système complet, mais gaspille énormément de mémoire SDRAM car elle doit contenir tout le système de fichier.
- Habituellement le noyau Linux et le système de fichier racine sont programmés dans différentes partitions de la mémoire flash, après la partition contenant `u-boot`. Seule le noyau Linux est recopié entièrement en mémoire SDRAM.
- Le système de fichier reste dans la mémoire flash ce qui permet un gain de mémoire SDRAM important.
- Le noyau utilise un cache en SDRAM pour accélérer les échanges avec le système de fichier en mémoire flash.

# Partitions en mémoire flash



## Partitionnement de la mémoire flash

- Le but est de pouvoir enregistrer dans la même mémoire flash les images suivantes:
  - Le chargeur d'amorçage u-boot, avec son environnement.
  - Le noyau Linux, compressé pour gagner de la place.
  - Le système de fichier racine, en lecture seule et compressé.
  - Le système de fichier des données persistantes.
- Avec le partitionnement, une simple mémoire flash SPI de type M25P64 ou M25P128 permet de contenir tout ce qui est nécessaire au fonctionnement du système.
- Remarque: prévoir une quantité suffisante de SDRAM.
  - 16 Mo est un minimum.
  - 64 Mo au maximum sur certain Blackfin a cause d'un bug silicium.

## Systeme de fichier en lecture seulement

- Dans le but de garantir l'intégrité du système de fichier racine, il est recommandé de le monter en lecture seulement. Ceci implique l'utilisation d'un système de fichier en SDRAM (`tmpfs`) pour les données temporaires. Le respect des bonnes habitudes UNIX par les applications fait que ceci est simple à mettre en place pour le dossier `/tmp`. Pour le dossier `/etc` qui contient tout les fichiers de configuration, il y a plusieurs approches possibles dans le cas où des fichiers doivent être dynamiques:
  - Créer un dossier `/tmp/etc` en système de fichier `tmpfs` et y recopier le contenu de `/etc`. Puis remonter le dossier `/tmp/etc` dans le dossier `/etc`. Avec cette méthode, tout les fichiers sont dynamiques.
  - Une variante de cette méthode est de sauver et de restaurer le contenu du dossier `/etc` dans un système de fichier persistant, par exemple dans la partition `data`. Il est aussi possible de directement monter un dossier `etc` de la partition persistante dans `/etc`.

## Fichiers de configuration dynamiques

- - Créer un dossier `/tmp/etc` et y copier l'état initial des fichiers dynamiques depuis, par exemple, un dossier `/etc.init`. Dans le dossier `/etc` ces fichiers sont remplacés par des liens symboliques dans le dossier `/tmp/etc`. Cette méthode permet de contrôler précisément quels fichiers sont dynamiques.
- Chaque méthode a ses avantages et ses inconvénients. Les critères de sélections sont propres à chaque système en fonction de son utilisation.

## BF-561 avec Linux-SMP

- Le processeur BF-561 inclut deux processeurs Blackfin. Depuis 2007, un noyau Linux existe spécifiquement pour faire fonctionner les deux processeurs en mode Symetric Multi Processor (SMP). Ce développement est maintenant stable au point qu'il a été déplacé dans une branche du noyau Linux du site [blackfin.uclinux.org](http://blackfin.uclinux.org). Sa fusion avec le tronc reste encore une tâche complexe.
- Le noyau SMP pour BF-561 ne peut techniquement pas se comporter comme un vrai noyau SMP, c'est pourquoi il est généralement appelé SMP-like. Le problème principal est que les caches des deux processeurs ne sont pas cohérentes entre elles. Cela implique que les threads qui communiquent entre eux doivent fonctionner sur le même processeur. La fonction `sched_setaffinity()` permet de garantir le fonctionnement de tous les threads d'une application sur un processeur donné. Pour profiter du second processeur, il faut y lancer une autre application.

## **BF-561: le problème d'arbitrage**

- Un second problème que pose le manque de cohérence des caches est que les écritures ne peuvent se faire que avec la police write-through qui provoque une écriture en mémoire pour chaque écriture en cache. Cela utilise une grande partie de la bande passante au détriment de la performance lorsqu'il y a deux processeurs pour un seul contrôleur mémoire.
- Le BF-561 a un arbitrage fixe qui donne systématiquement la priorité au processeur 0 (coeur A) et bloque le processeur 1 (coeur B). La conséquence est que en pleine charge, le processeur 1 exécute deux fois moins d'instructions en en temps donné que le processeur 0. C'est comme si le coeur A est à 600MHz et le coeur B est à 300MHz.

## L'architecture logicielle au laboratoire MIS-SE

- Le logiciel coûte plus cher que le matériel. Il doit être maintenu pendant des années tout en continuant à évoluer constamment.
- Le code doit donc être le plus générique possible et testé dans le plus grand nombre de conditions possibles.
  - Ces critères sont les mieux satisfaits avec les codes de projets Open Source maintenus par les grandes communautés actives. Les utiliser en priorité.
  - Pour ce qui n'existe pas en Open Source, programmer avec l'API GNU/Linux permet de coder et tester sur un PC avant de porter sur la cible. Cela oblige d'être un minimum générique et augmente la couverture des tests.
- Programmer plusieurs petites applications simples communicant entre elle avec des protocoles établis plutôt qu'une seule grande application.
  - Simplifie les spécifications et augmente la qualité des codes.
  - Répartition triviale des tâches entre programmeurs.
  - Meilleure réutilisation des codes déjà créés.
- Utiliser abondamment `git` pour archiver les changements de version.

## Quelques truc pour coder efficacement

- Utiliser les constructions C99 supportées par gcc
  - Déclaration de variable au milieu du code:
    - `for (int i = 10; i; --i) { buffer[i] = val; }`
  - Tableaux sur la pile initialisés par des variables:
    - `char buffer[] = { [2] = 0x22, [9] = addr, [4] = val };`
  - Une structure imbriquée est garantie alignée sur le début.
    - Permet de faire un héritage simple.
- Compiler avec un niveau d'attention élevé:
  - Options: `-Wall -Werror -pedantic`
- Tester les préconditions à d'une fonction et son résultat d'invocation.
  - Tout comportement non analysé doit terminer immédiatement le programme avec un message incluant assez d'information pour faire du debug.
- La librairie `glib` pour simplifier les opérations de base:
  - Gestion puissante des strings, listes, quark, fichiers, exécution, options en argument, byte order, atomicité, threads, synchronisation, boucle applicative, canaux d'entrée/sortie, queue, internationalisation, date et temps, complétion automatique, fichier de configuration, recherche de fichier, etc...
- Le protocole D-Bus pour communiquer entre applications.

## Conclusion

- UNIX, GNU, Linux, distributions, etc...
  - C'est un système vaste qui ne s'apprend pas en un jour.
  - Mais il offre un environnement fiable, complet et bien testé.
  - Très large communauté de développeurs impliqués.
- Un processeur sans MMU oblige l'utilisation de code spécifique
  - uClibc remplace la glibc; busybox remplace nombre d'applications GNU.
  - Quelques limitations, mais reste généralement compatible.
- La chaîne de construction Blackfin
  - Quelques particularités auquel il faut faire attention.
- Linux SMP du BF-561 fonctionne, mais avec de grosses limitations.
- L'utilisation de code Open Source permet réellement d'augmenter la qualité du système tout en développant plus rapidement.
  - Permet de coder d'une façon plus générique et de mieux tester.
  - La réutilisation de codes existant réduit les coûts de développement.
  - La différence entre un PC et un système embarqué tend à diminuer.