

Plan du cours

- ④ **Taxonomie des systèmes informatiques**
- ④ **Systèmes temps réel**
- ④ **Spécificités des OS pour le temps réel**
- ④ **L'OS Xenomai pour le temps réel**
- ④ **Systèmes embarqués**
- ④ **Linux pour l'embarqué**
- ④ **Marché des OS pour le temps réel et l'embarqué**
- ④ **Modélisation d'applications temps réel avec UML 2.x**

Quelques dates clés

- **1999** : Linux embarqué apparaît pour la 1ère fois officiellement à une exposition *Linux World*



- **2000** : création du consortium Linux embarqué (*Embedded Linux Consortium*) regroupant :

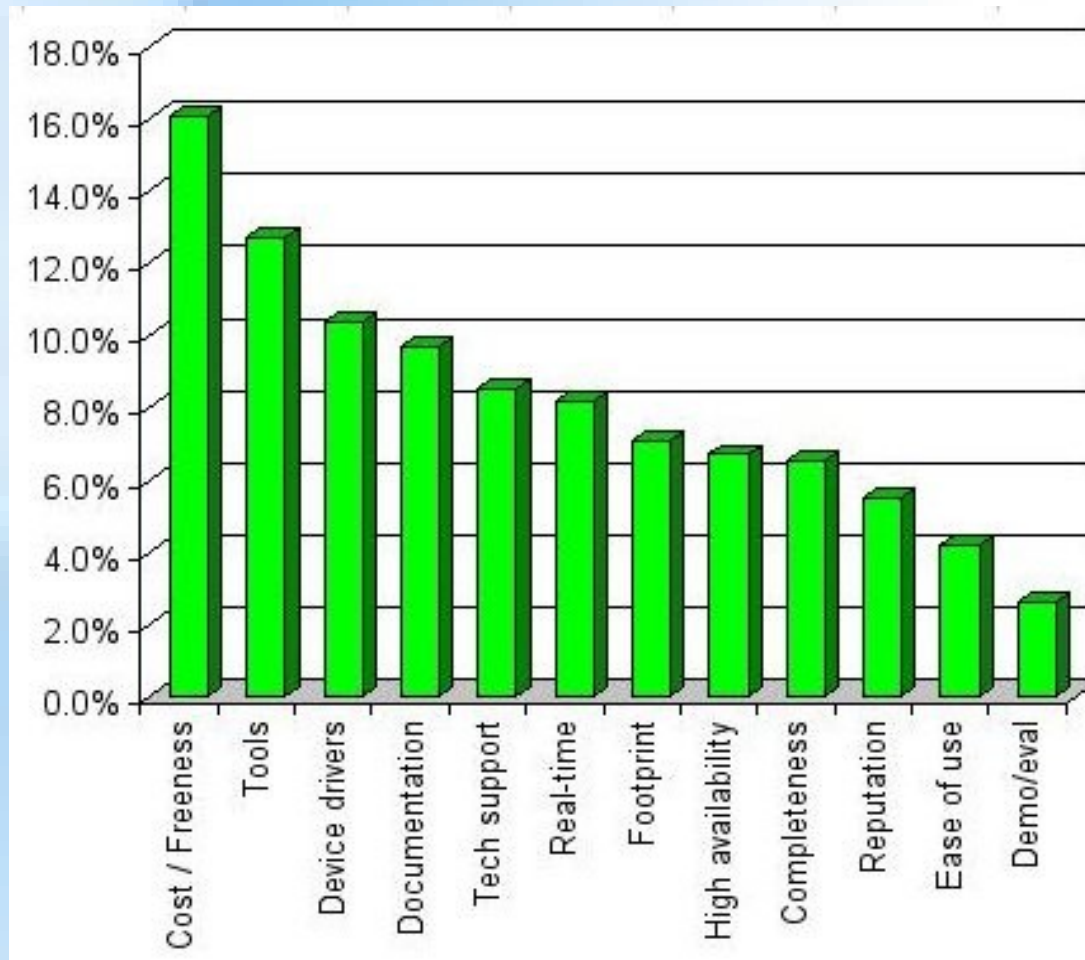


- des éditeurs de distributions Linux
- des éditeurs de systèmes temps-réel propriétaires
- des fabricants de composants

Description

- **Linux embarqué = adaptation du noyau Linux** à un système embarqué
- **Les fonctionnalités** du noyau Linux embarqué sont **adaptées** aux capacités du système embarqué :
 - Moins de mémoire requise
 - Moins de services disponibles
 - Boot depuis une mémoire ROM (FLASH)
 - Pas de clavier ou de souris requis
- Une version de Linux embarqué peut être **configurée « à la carte »** pour fonctionner sur une plateforme donnée

Linux embarqué, une solution attrayante ?



Snapshot of the Embedded Linux market - March, 2004

Points forts de Linux pour l'embarqué

- **Fiabilité**
- **Faible coût**
- **Performances**
- **Adaptabilité**
- **Configurabilité**
- **Portabilité**
- **Ouverture**
- **Modularité**



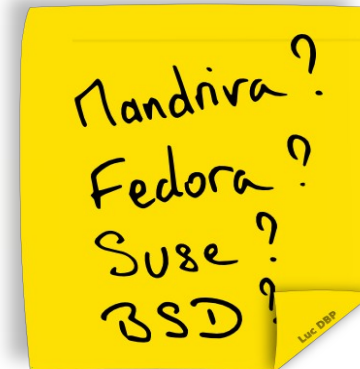
Points faibles de Linux pour l'embarqué

- **Crédibilité de l'open-source**
- **Méconnaissance des licences**
- **Support temps-réel dur**
- **Outils de développement**
- **Support technique**



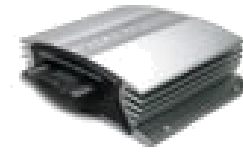
Les offres Linux embarqué

- **Distributions Linux classiques** (taille RAM > 64Mo, taille ROM >150Mo)
- **Linux embarqué avec support commercial**
- **Linux embarqué gratuit**



Produits à base de Linux embarqué (1)

- **Phatnoise PhatBox** -- un autoradio avec disque dur



- **D-Link MediaLounge DSM-320** -- un lecteur multimédia sans fil



- **iRiver PMC-120** -- un lecteur multimedia portable



- **Linksys Wireless-G WRT54G** -- Routeur sans fil-G à haut débit



Produits à base de Linux embarqué (2)

- **Nokia 770 Internet Tablet** -- un PDA-tablet PC
- **Motorola MotoRokr EM35** -- un téléphone mobile
- **Tivo Series 2** -- un magnétoscope numérique
- **TomTom ONE** -- un navigateur GPS pour automobile



Quelques dates clés

- **1999** : Linux embarqué apparaît pour la **1ère fois officiellement** à une exposition *Linux World*



- **2000** : création du consortium Linux embarqué (*Embedded Linux Consortium*) regroupant :



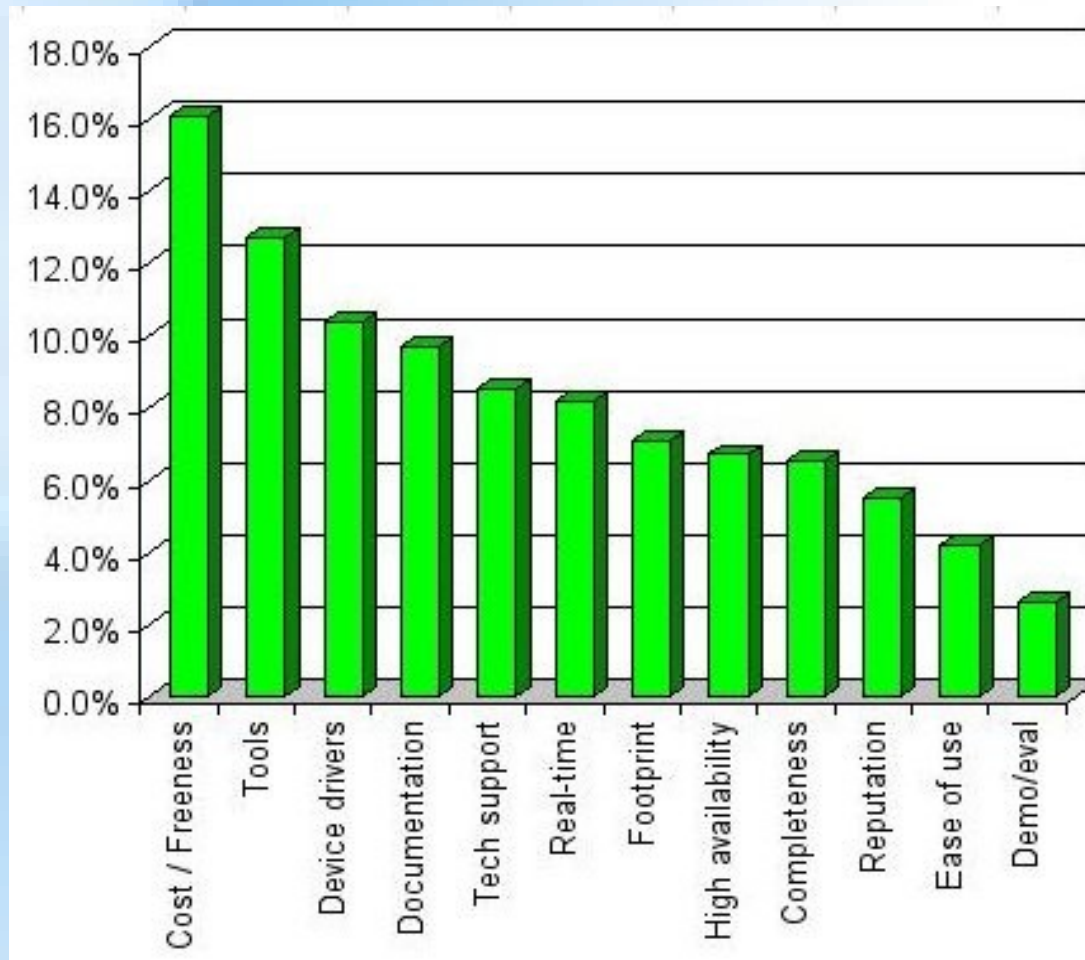
- des éditeurs de distributions Linux
- des éditeurs de systèmes temps-réel propriétaires
- des fabricants de composants

Description

Linux embarqué = Utilisation du noyau Linux et de composants logiciels libres pour mettre en œuvre un système embarqué

- **Les fonctionnalités** du noyau Linux embarqué sont **adaptées** aux capacités du système embarqué :
 - Moins de mémoire requise
 - Moins de services disponibles
 - Moins de pilotes de périphériques
 - Boot depuis une mémoire ROM (FLASH)
- Une version de Linux embarqué peut être **configurée « à la carte »** pour fonctionner sur une plateforme donnée

Linux embarqué, une solution attrayante ?



Snapshot of the Embedded Linux market - March, 2004

12

Points forts de Linux pour l'embarqué

- **Fiabilité**
- **Faible coût**
- **Performances**
- **Adaptabilité**
- **Configurabilité**
- **Portabilité**
- **Ouverture**
- **Modularité**



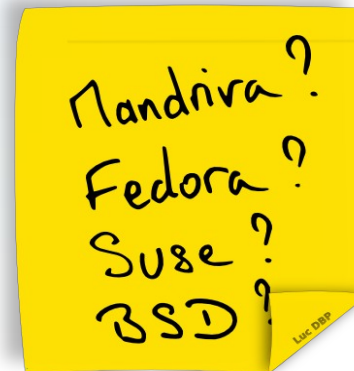
Points faibles de Linux pour l'embarqué

- **Crédibilité de l'open-source**
- **Méconnaissance des licences**
- **Support temps-réel dur**
- **Outils de développement**
- **Support technique**



Les offres Linux embarqué

- Distributions Linux classiques (taille RAM > 64Mo, taille ROM >150Mo)
- Linux embarqué avec support commercial
- Linux embarqué gratuit



Produits à base de Linux embarqué (1)

- **GPS** – Tomtom et Garmin



- **Routeurs personnels** – Linksys, Freebox, Livebox



- **PDA** – Zaurus, Nokia N8x0

- **Téléviseurs, caméscopes, lecteurs de DVDs** – Sony, Philips



- **Téléphone** – Nokia N900, OpenMoko, Palm Pre



Produits à base de Linux embarqué (2)

- Eoliennes

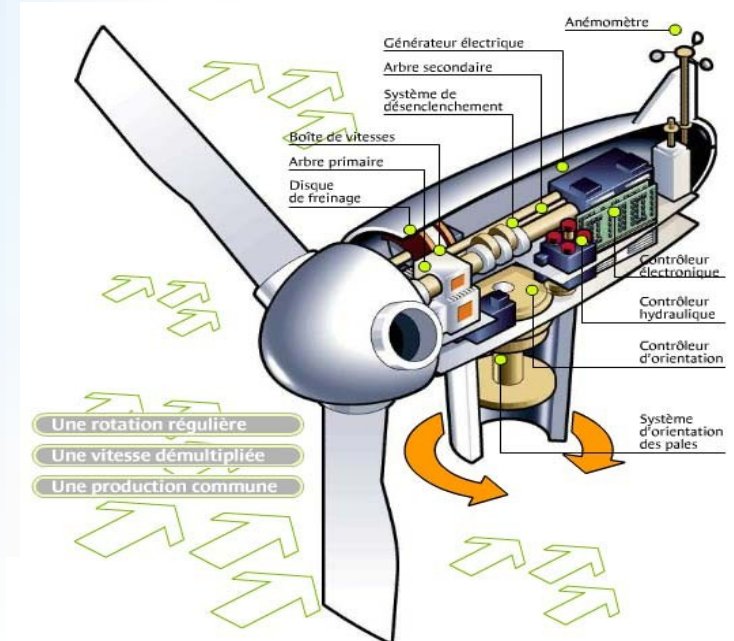


- Parc-mètres

- Terminaux de paiement

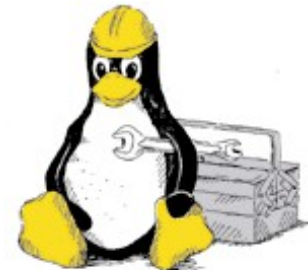
- Trains

- Avions

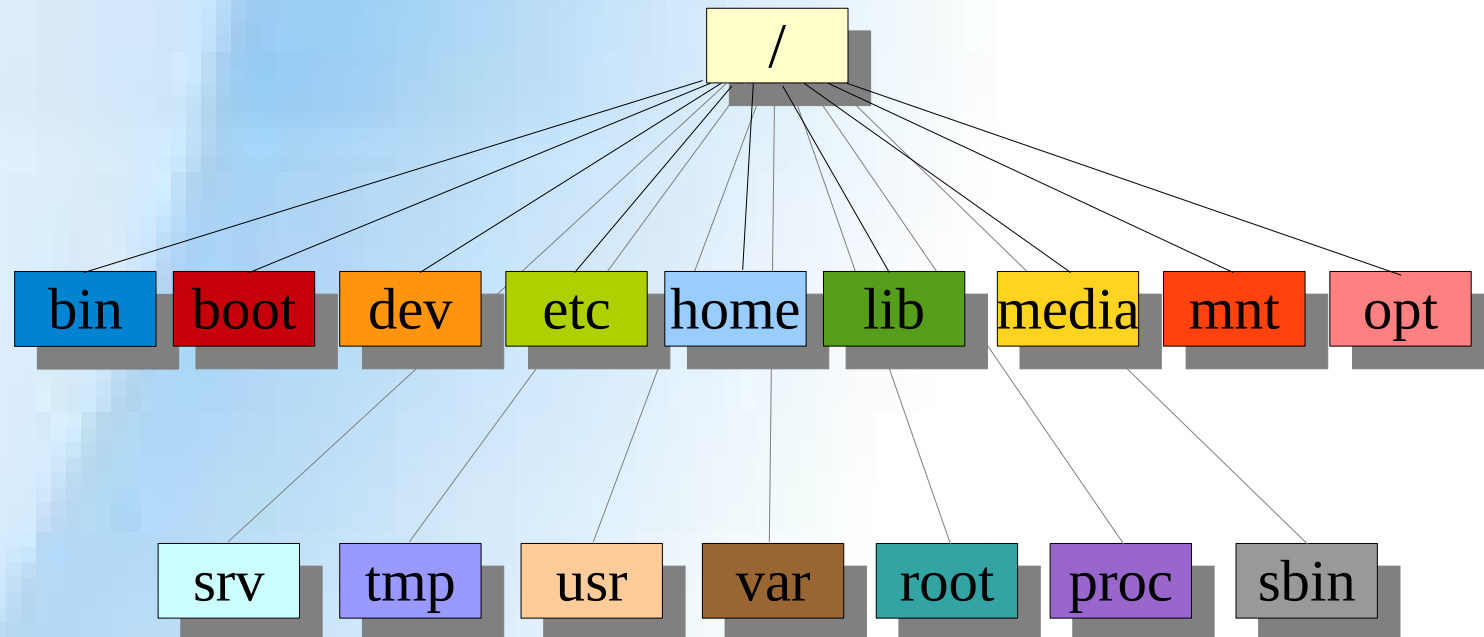


Méthodologie de construction d'un Linux embarqué

- **Rappels sur la structure de Linux**
- **Méthodologie générale**
- **Construction du système de base**

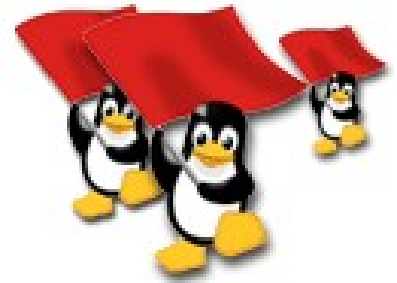


L'arborescence d'un système Linux



Généralités

- Un système Linux est composé de 3 parties :
 - le **noyau** (kernel)
 - la **libc**, bibliothèque principale pour les applications
 - les **applications** (ou commandes)
- Le système Linux est chargé en mémoire grâce à un logiciel situé dans le *MBR* :
 - le **bootstrap**, programme (chargeur) de démarrage



La séquence de démarrage

- Elle est toujours la même :
 - chargement du système par **un chargeur de démarrage**
 - chargement du **noyau Linux**
 - lancement du **programme d'initialisation /sbin/init**

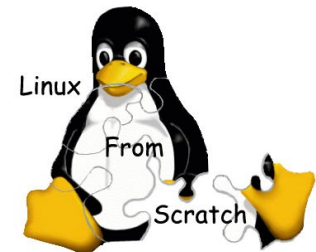
Différentes méthodes de construction

- **Distribution standard**

- Utilisation d'une distribution GNU/Linux du commerce
- Pb : espace disque conséquent (*minimum 100Mo*)

- **LFS : Linux From Scratch**

- Construction du système GNU/Linux en se basant *uniquement sur les sources des programmes*



- **LFS sur la base de composants existants**

- Construction du système *en se basant sur un assemblage de programmes et bibliothèques compilées*

Les différentes phases de construction

- Préparation de l'arborescence du système
- Récupération/Configuration/Compilation de **uClibc**
- Récupération/Configuration/Compilation de **Busybox**
- Récupération/Configuration/Compilation d'un **noyau Linux**
- Récupération/Configuration de **SysLinux**
- **Pré-test avec Qemu et installation sur le système cible**



Préparation de l'arborescence du système (1)

- Création des répertoires

```
mkdir monLinuxEmbarque  
cd monLinuxEmbarque
```

```
mkdir -p bin lib root home proc mnt sys tmp boot  
mkdir -p usr/{lib,local,games,share,bin,sbin}  
mkdir -p var/{cache,lib,lock,log,run,spool}  
mkdir -p media/{cdrom,flash,usbdisk}  
mkdir -p dev/{input,pts,shm,net,usb}  
mkdir -p etc/init.d
```


Préparation de l'arborescence du système (2)

- Spécification des interpréteurs de commande valides

nano /etc/shells

```
# /etc/shells : valid login shells  
  
/bin/sh  
/bin/bash  
/bin/ash  
/bin/tcsh
```

Préparation de l'arborescence du système (3)

- Définition du fichier de configuration pour init

nano /etc/inittab

```
# /etc/inittab : init configuration file for Ma-distribution-embarquee

::sysinit:/etc/init.d/rcS
::askfirst:/bin/sh
::restart:/sbin/init
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a -r
```

- Définition de la liste des systèmes de fichiers montables

nano /etc/fstab

```
# /etc/fstab : information about static file system
proc      /proc          proc      defaults      0            0
sysfs     /sys              sysfs     defaults      0            0
devpts    /dev/pts          devpts    defaults      0            0
tmpfs     /dev/shm          tmpfs     defaults      0            0
```

Préparation de l'arborescence du système (4)

- Script de démarrage init

nano /etc/init.d/rcS

```
# /etc/init.d.rcS : rcS initial script
# !/bin/sh

echo "Processing /etc/init.d/rcS..."
/bin/mount proc
/bin/mount -a
```

chmod +x /etc/init.d/rcS

uClibc

- Bibliothèque C légère conçue pour de petits systèmes embarqués (≈ 400 Ko)
- Récupération

```
wget http://www.uclibc.org/downloads/uClibc-0.9.33.tar.bz2
```

- Copie des librairies minimales

```
cp lib/{libcrypt.so.1,libm.so.6,libc.so.6} $PATH-monLinuxEmbarque/lib  
cp lib/lib-linux.so.2 $PATH-monLinuxEmbarque/lib
```

Busybox (1)

- **Commandes disponibles** (<500 Ko compilés statiquement avec uClibc)

acpid, addgroup, adduser, adjtimex, ar, arp, arping, ash, awk, basename, beep, blkid, brctl, bunzip2, bzip2, cal, cat, catv, chat, chattr, chgrp, chmod, chown, chpasswd, chpst, chroot, chrt, chvt, cksum, clear, cmp, comm, cp, cpio, crond, crontab, cryptpw, cut, date, dc, dd, deallocvt, delgroup, deluser, depmod, devmem, df, dhcprelay, diff, dirname, dmesg, dnsd, dnsdomainname, dos2unix, dpkg, du, dumpkmap, dumpleases, echo, ed, egrep, eject, env, envdir, envuidgid, expand, expr, fakeidentd, false, fbset, fbsplash, fdflush, fdformat, fdisk, fgrep, find, findfs, flash_lock, flash_unlock, fold, free, freeramdisk, fsck, fsck.minix, fsync, ftpd, ftpget, ftpput, fuser, getopt, getty, grep, gunzip, gzip, hd, hdparm, head, hexdump, hostid, hostname, httpd, hush, hwclock, id, ifconfig, ifdown, ifenslave, ifplugd, ifup, inetd, init, inotifyd, insmod, install, ionice, ip, ipaddr, ipcalc, ipcrm, ipcs, iplink, iproute, iprule, iptunnel, kbd_mode, kill, killall, killall5, klogd, last, length, less, linux32, linux64, linuxrc, ln, loadfont, loadkmap, logger, login, logname, logread, losetup, lpd, lpq, lpr, ls, lsattr, lsmod, lzmacat, lzop, lzopcat, makemime, man, md5sum, mdev, msg, microcom, mkdir, mkdosfs, mkfifo, mkfs.minix, mkfs.vfat, mknod, mkpasswd, mkswap, mktemp, modprobe, more, mount, mountpoint, mt, mv, nameif, nc, netstat, nice, nmeter, nohup, nslookup, od, openvt, passwd, patch, pgrep, pidof, ping, ping6, pipe_progress, pivot_root, pkill, popmaildir, printenv, printf, ps, pscan, pwd, raidautorun, rdate, rdev, readlink, readprofile, realpath, reformime, renice, reset, resize, rm, rmdir, rmmod, route, rpm, rpm2cpio, rtcwake, run-parts, runlevel, runsv, runsvdir, rx, script, scriptreplay, sed, sendmail, seq, setarch, setconsole, setfont, setkeycodes, setlogcons, setsid, setuidgid, sh, sha1sum, sha256sum, sha512sum, showkey, slattach, sleep, softlimit, sort, split, start-stop-daemon, stat, strings, stty, su, sulogin, sum, sv, svlogd, swapoff, swapon, switch_root, sync, sysctl, syslogd, tac, tail, tar, taskset, tcpsvd, tee, telnet, telnetd, test, tftp, tftpd, time, timeout, top, touch, tr, traceroute, true, tty, ttysize, udhcpc, udhcpd, udpsvd, umount, uname, uncompress, unexpand, uniq, unix2dos, unlzma, unlzop, unzip, uptime, usleep, uudecode, uuencode, vconfig, vi, vlock, volname, watch, watchdog, wc, wget, which, who, whoami, xargs, yes, zcat, zcip

Busybox (2)

- Récupération

```
wget http://busybox.net/downloads/busybox-1.20.2.tar.bz2
```

- Extraction

```
tar xjf busybox-1.20.2.tar.bz2
```

- Configuration

```
make menuconfig
```

- Compilation

```
make
```

Busybox (3)

- Pré-installation (*dans le sous-répertoire `_install/`*)

`make install`

- Copie des fichiers compilés par Busybox

`cp -a _install/* $PATH-monLinuxEmbarque`

- Création des nœuds d'accès aux périphériques matériels

`mkdevs.sh`

Le noyau Linux (1)

- Récupération

```
wget ftp://ftp.kernel.org/pub/linux/kernel/v3.x/linux-3.5.2.tar.bz2
```

- Extraction

```
tar xjf linux-3.5.2.tar.bz2
```

- Configuration

```
make menuconfig
```

- Compilation du noyau

```
make bzImage
```


Le noyau Linux (2)

- Compilation des modules chargeables dynamiquement

```
make modules
```

- Installation des modules chargeables dynamiquement compilés

```
make INSTALL_MOD_PATH=$PATH-monLinuxEmbarque/lib modules_install
```

- Copie du noyau compressé

```
cp arch/i386/boot/bzImage $PATH-monLinuxEmbarque/boot
```

- Création du système de fichier compressé au format gzip et archivé via cpio

```
find . -print | cpio -o -H newc | gzip -9 >
```

```
PATH-monLinuxEmbarque/boot/rootfs.gz
```

Syslinux

- Récupération

```
wget ftp://ftp.kernel.org/pub/linux/utils/boot/syslinux/syslinux-4.05.tar.gz
```

- Copie

```
cp isolinux.bin boot/isolinux
```

- Configuration

```
nano boot/isolinux.cfg
```

```
display display.txt
default ma-distrib-embarquee
label ma-distrib-embarquee
    kernel /boot/bzImage
    append initrd=boot/rootfs.gz root=/dev/XXX rootdelay=10
implicit 0
prompt 1
timeout 80
```

```
nano boot/display.txt
```

```
/*          \ | /
              ( o o )
+-----o00-{_}-00o-----
Ma distribution embarquee

<ENTER> to boot.
```

Pré-test du système avec Qemu

- Création d'une image iso

```
genisoimage -R -o cd.iso  
-b $PATH-monLinuxEmbarque/boot/isolinux/isolinux.bin  
-no-emul-boot -boot-load-size 4 -V "Live"  
-input-charset iso8859-1 -boot-info-table live
```

- Emulation du boot sur la cible

```
qemu-system-i386 -cdrom cd.iso
```

Installation sur le système cible (ex: une clé USB)

- Transfert de l'arborescence sur la cible

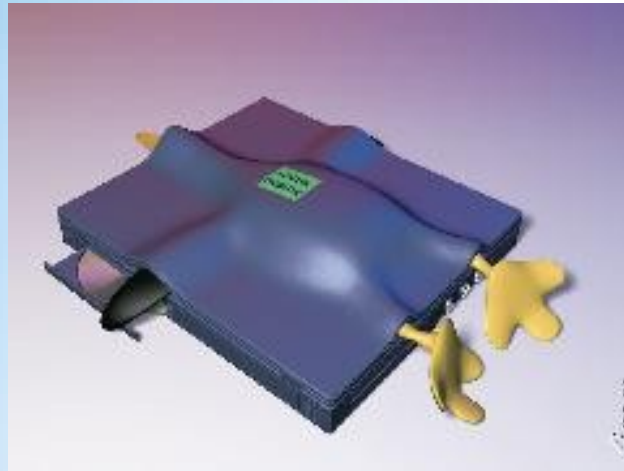
```
cp -Rdp $PATH-monLinuxEmbarque/* /dev/sdbX
```

```
cp -Rdp $PATH-monLinuxEmbarque/*.* /dev/sdbX
```

- Installation du bootloader

```
syslinux -s /dev/sdbX
```

Test du Linux embarqué



Références

- **Portail Linux temps réel** : <http://www.realtimelinuxfoundation.org>
- **Portail Linux embarqué** : <http://www.linuxdevices.com>
- **Portail communautaire Xenomai** : <http://www.xenomai.org>
- **Logiciel Libre** : www.framasoft.net