Open Wide
Architecte Open Source

LiveCD X86 Xenomai

Xenomai sur architecture ARM9

Licence

Sommaire

- Présentation de Xenomai
- LiveCD X86 Xenomai
- Xenomai sur architecture ARM9

Première partie:

Présentation de Xenomai

Sommaire

- Introduction
- Historique
- Description fonctionnelle
- Architecture
- Adeos
- Domaines d'exécution
- Interfaces temps-réel
- Outils de mise au point

Introduction

- Xenomai, sous-système temps-réel de Linux
  - Faible latence
  - Espace utilisateur
  - RTOS générique + interfaces
  - Emulation de RTOS traditionnels
  - Complètement intégré à Linux
- Licence GPL (coeur), LGPL (interfaces)

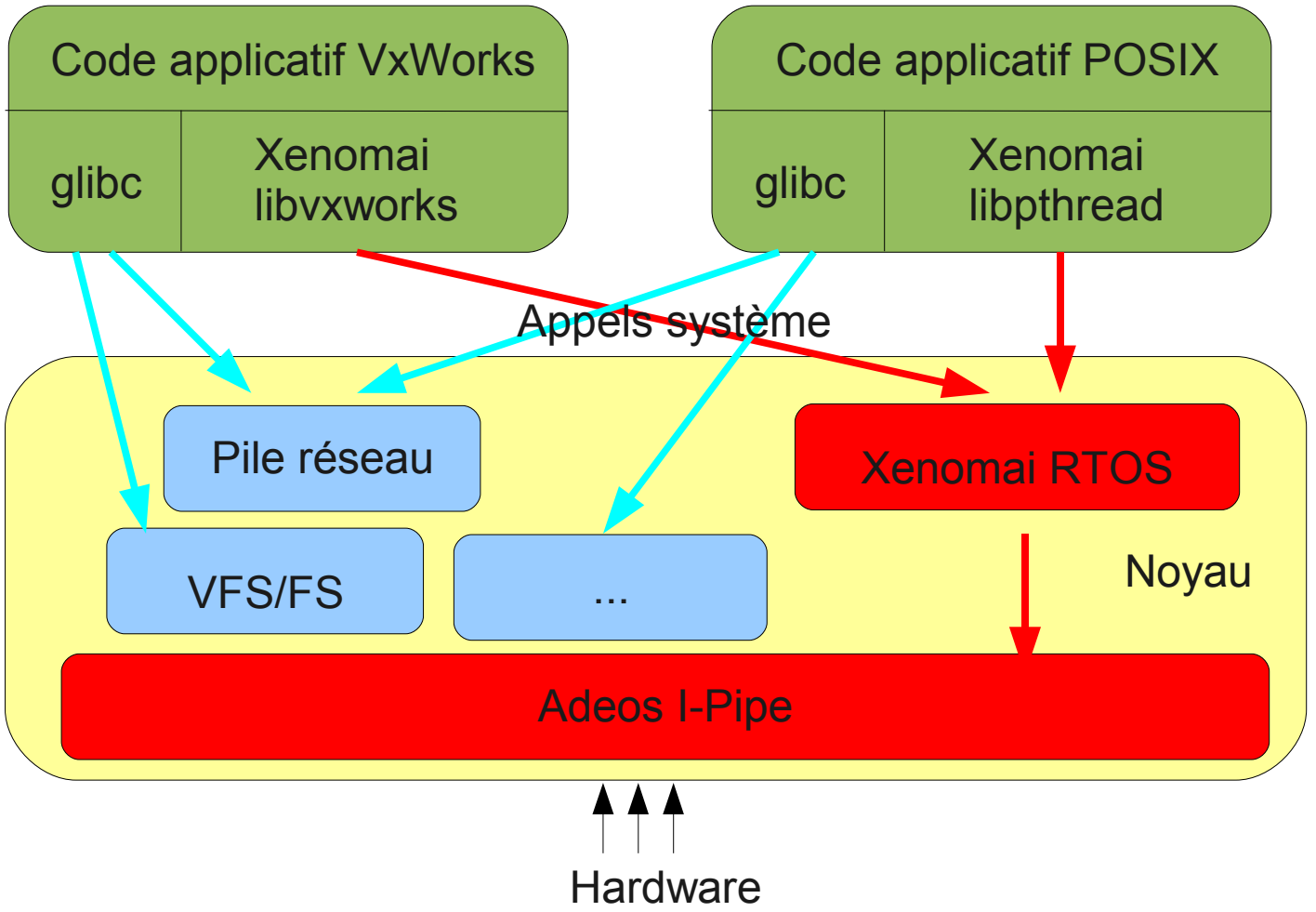
Xenomai Historique

- Fondation en 2001
  - Xenomai v0.5 – Septembre 2001
  - Xenomai v1.1.1 – Décembre 2002
- Intégration à RTAI en 2003
  - RTAI/fusion v0.1 – Juin 2004
- Indépendance reprise en 2005
  - Xenomai v2.0 – Octobre 2005 (base RTAI/fusion v0.9.1)
  - Xenomai v2.1 – Mars 2006
  - Xenomai v2.2 – Juillet 2006
  - Xenomai v2.3 – Décembre 2006
  - Xenomai v2.4 – Décembre 2007

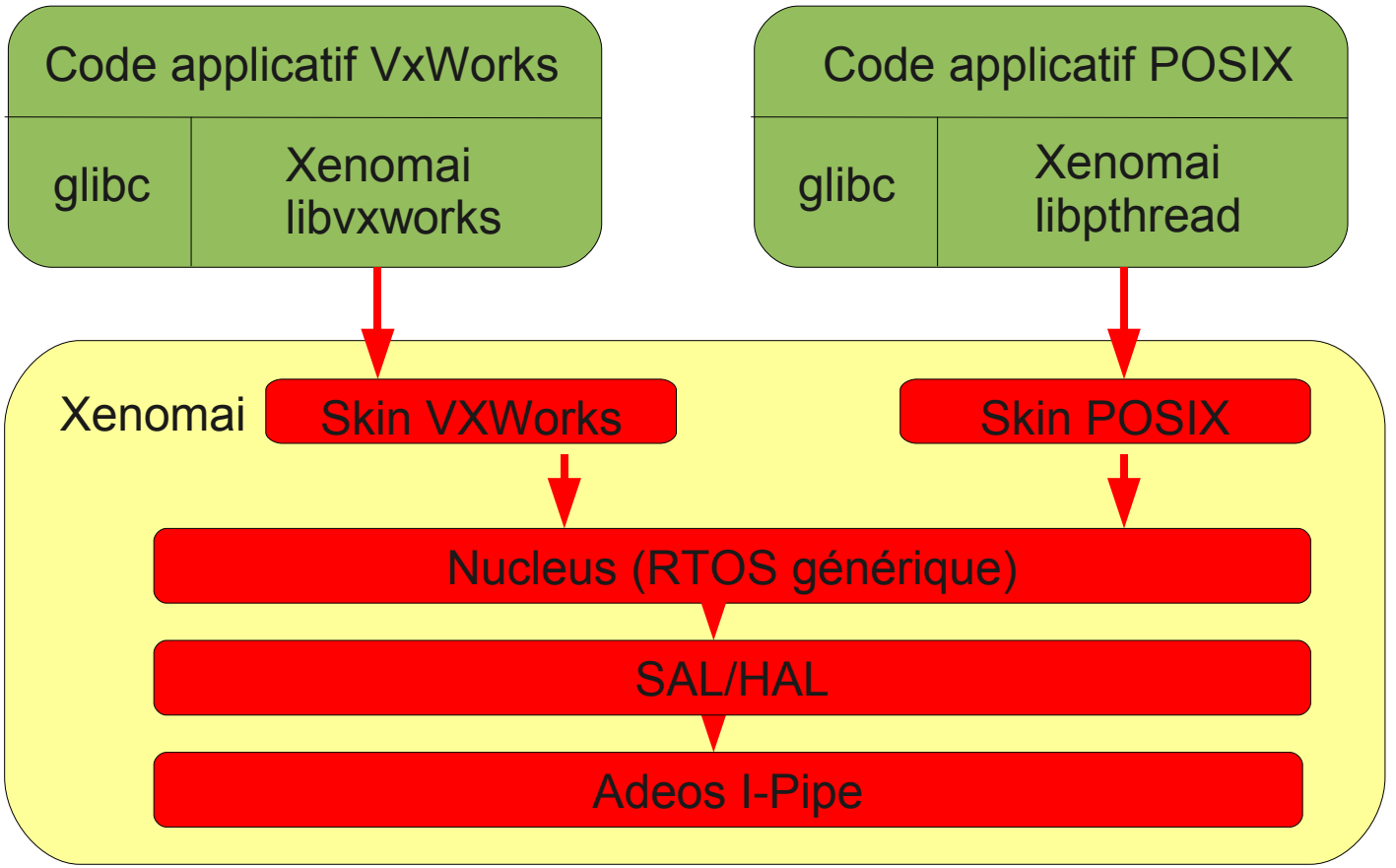
Xenomai
Description fonctionnelle (1/3)

- Adaptabilité
  - coeur de RTOS générique
  - spécialisation d'interfaces ou *skins*
- Intégration Linux forte
  - sous-système noyau
- Multi plates-formes
  - *arm, blackfin, i386, ia64, powerpc32, powerpc64*
  - *simulateur*
- *Modèle de programmation*
  - espace noyau
  - espace utilisateur standard

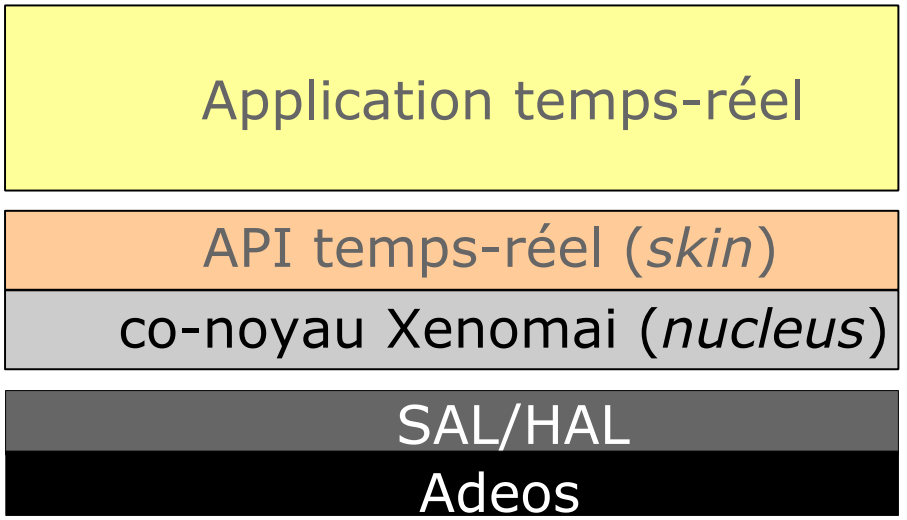
Xenomai
Description fonctionnelle (2/3)



Xenomai
Description fonctionnelle (3/3)



Xenomai
Architecture générale (1/2)



- couche API (POSIX, pSOS+, VRTX, VxWorks, uITRON ...)
- RTOS générique
- Dépendances matérielles

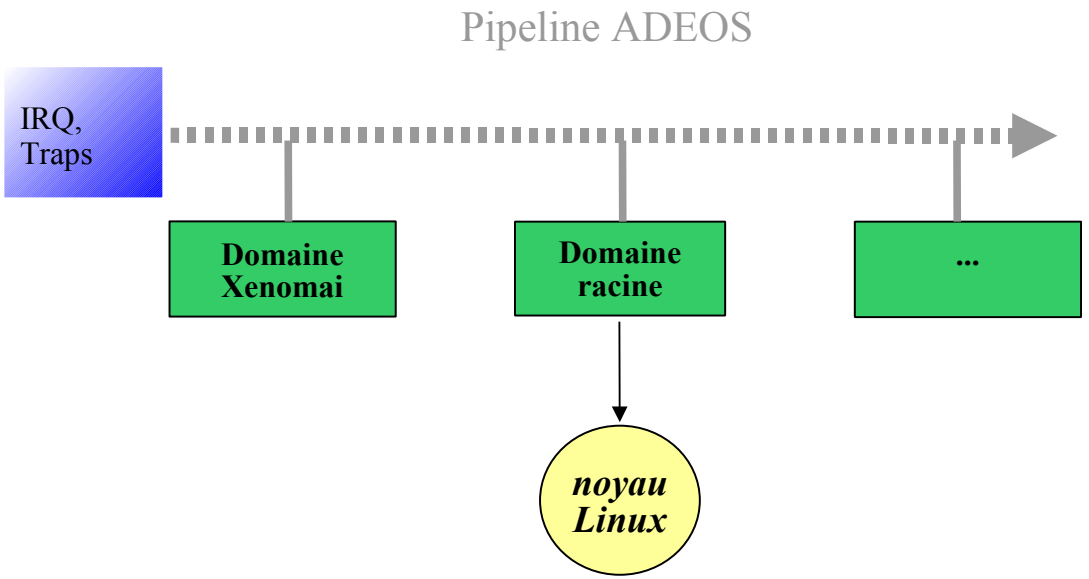
Xenomai
Architecture générale (2/2)

- Interfaces temps-réel, ou *skins*
  - absence d'API centrale et unique
  - spécialisations d'un RTOS générique
  - outil de migration
- RTOS générique
  - objets temps-réel réutilisables
- Couches d'abstraction d'architecture hôte
  - System Abstraction Layer (SAL)
  - Hardware Abstraction Layer (HAL)
- Adeos

Adeos (1/2)

- Virtualisation de ressources
  - interruptions externes et simulées
  - exceptions processeur
  - appels système
- Architecture type *pipeline* d'événements
  - organisation en domaines (Xenomai, Linux...)
  - notifications d'événements selon priorité
- API générique
- Portabilité
  - x86, ppc, ppc64, ia64, arm, blackfin

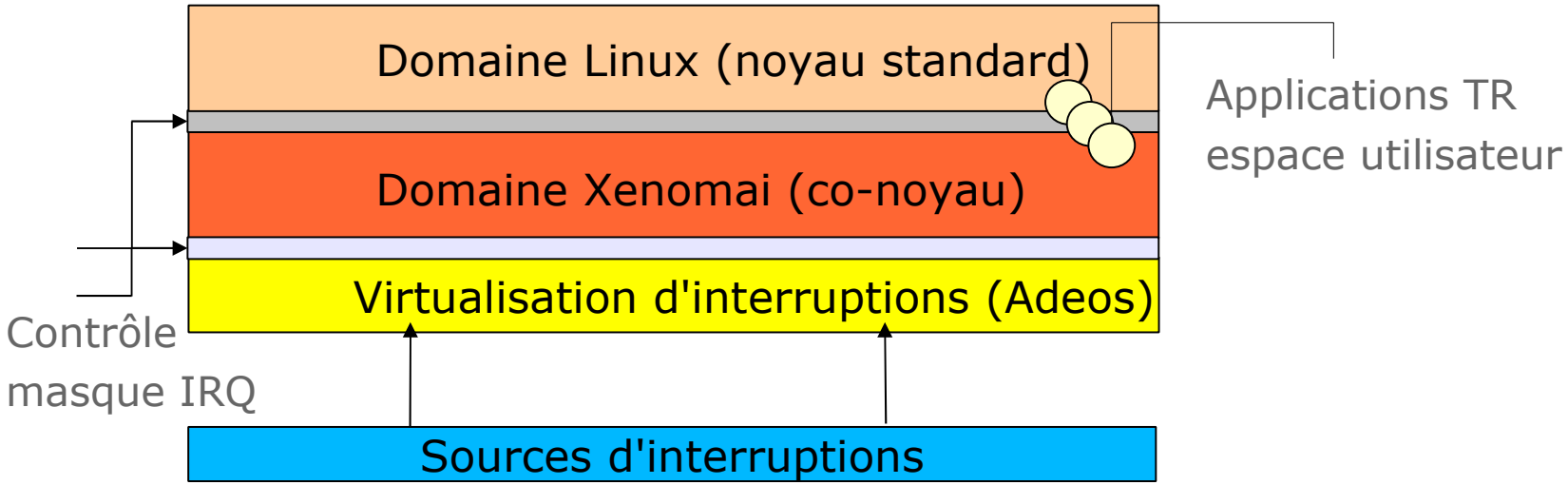
Adeos (2/2)


Pipeline ADEOS
IRQ,
Traps
Domaine
Xenomai
Domaine
racine
...
noyau
Linux

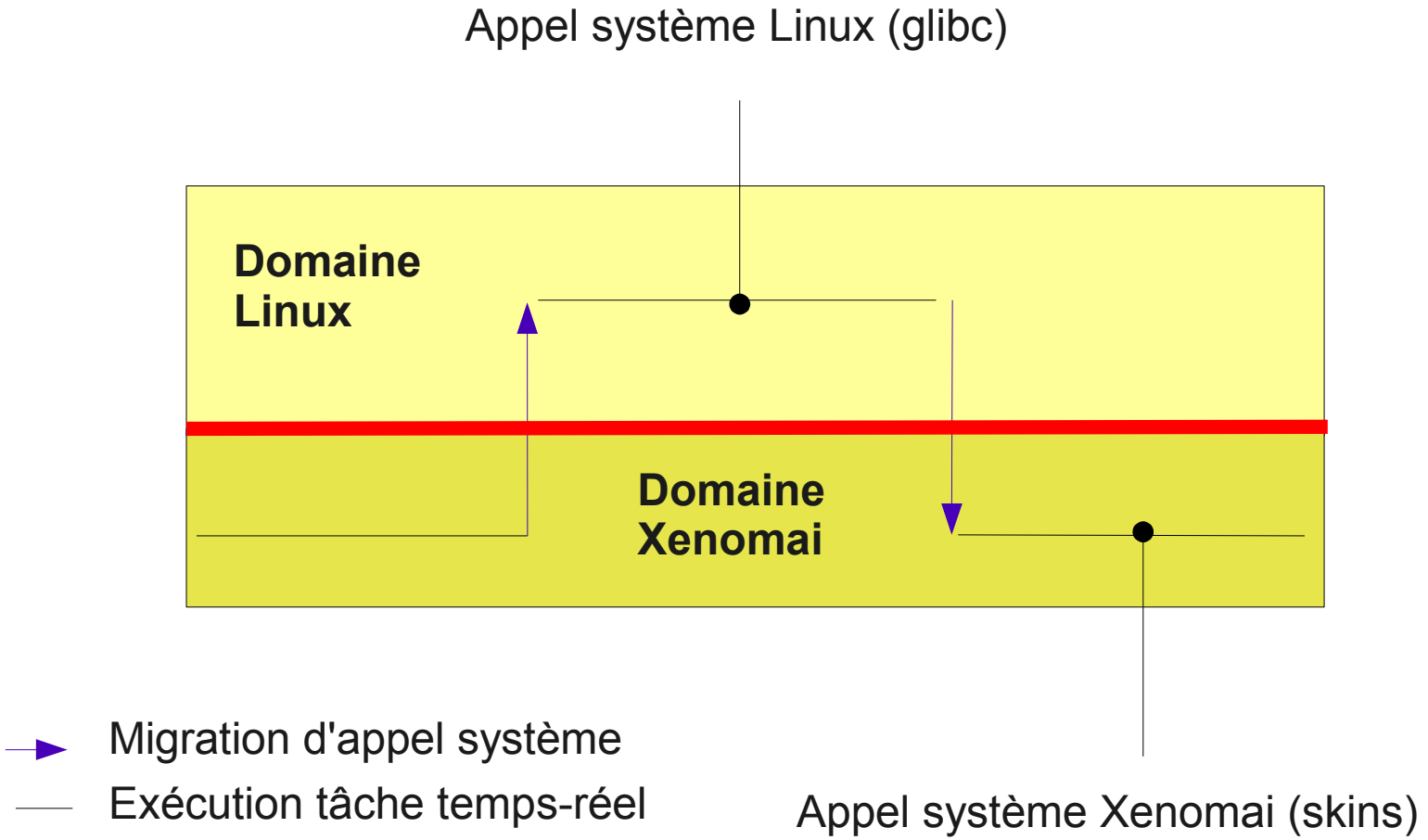
Xenomai
Domaines d'exécution (1/3)

- Dualité d'ordonnancement
  - Domaine Xenomai, déterministe
  - Domaine Linux, non déterministe
- Exécution d'une tâche temps-réel
  - Mode primaire (domaine Xenomai)
  - Mode secondaire (domaine Linux)
- Migration de modes
  - Automatique sur appels système

Xenomai
Domaines d'exécution (2/3)



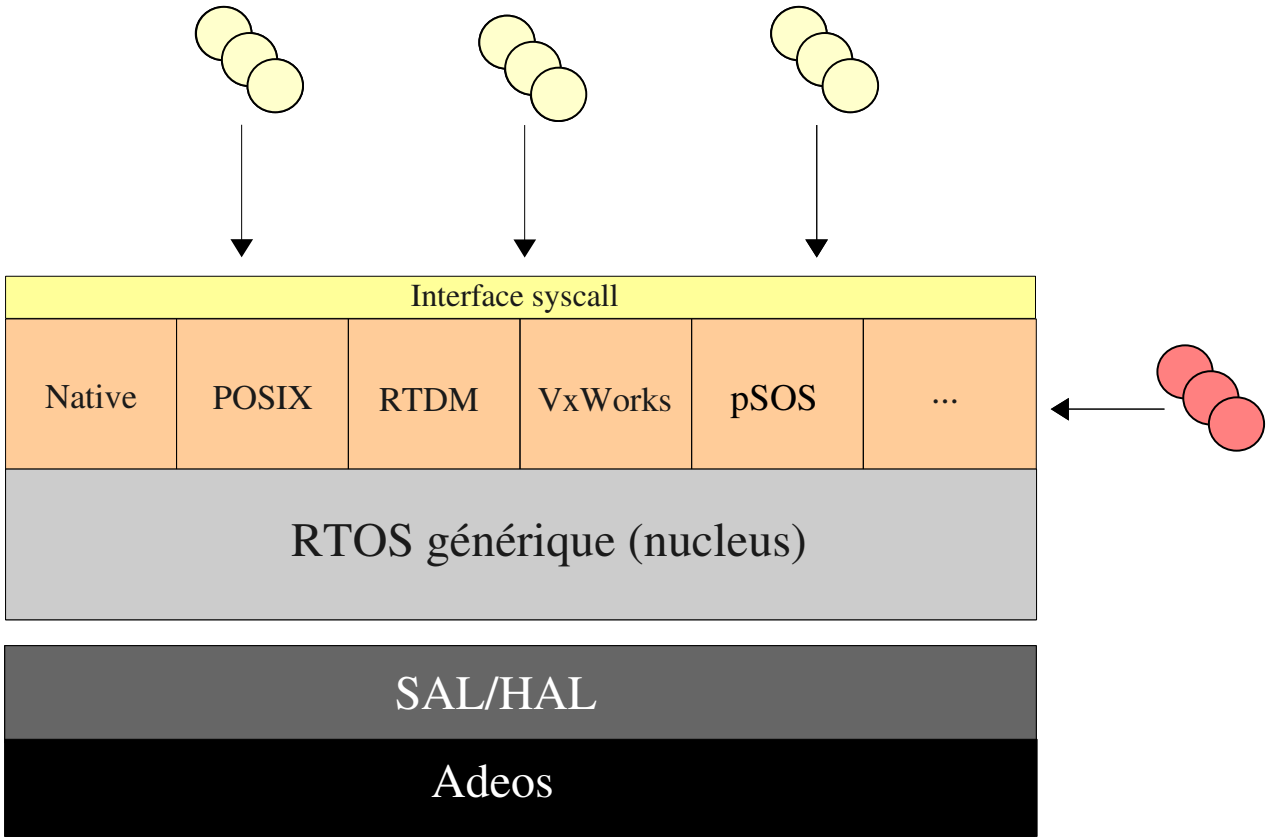
Xenomai
Domaines d'exécution (3/3)



Xenomai
Interfaces temps-réel (1/2)

- Personnalités temps-réel
  - *native*
  - *posix*
  - *psos+*
  - *rtai*
  - *uitron*
  - *vrtx*
  - *vxworks*
- API pour pilotes temps-réel de périphériques
  - *rtdm*

Xenomai
Interfaces temps-réel (2/2)



Applications en espace utilisateur

Applications en espace noyau

Xenomai
Outils de mise au point

- Watchdog
- Interface /proc/xenomai
- GDB
- Simulateur Xenomai
- Traceur I-Pipe
- KGDB

Liens & Documentation

- Xenomai
  http://www.xenomai.org
- Adeos
  http://home.gna.org/adeos/
- Interactions entre Adeos et Xenomai
  http://www.xenomai.org/documentation/branches/v2.3.x/pdf/Life-with-Adeos-rev-B.pdf
- Manifeste du projet Xenomai
  http://www.xenomai.org/documentation/branches/v2.3.x/pdf/xenomai.pdf
- Racine de documentation des interfaces
  http://www.xenomai.org/documentation/trunk/html/api/index.html
- Spécification POSIX de référence pour Xenomai
  http://www.opengroup.org/onlinepubs/009695399/

Deuxième partie:

LiveCD X86 Xenomai

Sommaire

- Objectifs
- Fonctionnement général
- Étapes de création de l'image bootable

- Linux : `initramfs`
- BusyBox : `init` et `mdev`
- Création de l'image bootable
- Tests

- Liens et documentation

Objectifs

- Outil de démonstration simple
- Vérification du fonctionnement de Xenomai
- Évaluation des performances de Xenomai sur une machine (latence, ...)

Fonctionnement général

- Noyau Linux minimaliste
  - Pas d'affichage graphique évolué (console)
  - Pas besoin de gérer les disques dur (initramfs)
  - ...
- Distribution basée sur BusyBox
- Procédure de boot simple : pas de détection de matériel nécessaire (`init` fourni par BusyBox)

Étapes de création de l'image bootable

- Patch, configuration et compilation du noyau
- Préparation du rootfs de la distribution
  - Compilation et installation de Xenomai & BusyBox
  - Copie des bibliothèques dynamiques
- Création d'une archive contenant le rootfs
- Création de l'image ISO (Noyau Linux + GRUB + initramfs_data.gz)

- Dans tout les noyaux de la branche 2.6
  - Rootfs monté en mémoire vive (/)
  - Basé sur tmpfs (ou ramfs)
- Décompression d'une archive cpio dans ce rootfs
- Si /init existe il est exécuté

- Possibilité de stocker l'intégralité du système dans cette archive

- Création de l'archive cpio compressée en gzip

```
cd rootfs &&
   find . | cpio -o -H newc | gzip > ../initramfs.gz
```

BusyBox
> init et mdev

- Utilisation /sbin/init fourni par BusyBox
- Nécessite un /dev valide

- /init monte et rempli /dev...

```
#! /bin/sh
/bin/mount -t sysfs sysfs /sys
/sbin/mdev -s
/bin/mkdir /dev/pts
exec /sbin/init
```

- ... avant de donner la main à /sbin/init de BusyBox

GNU GRUB

- Utilisation de GNU GRUB comme bootloader

- Fichier de configuration
  `xenolive/boot/grub/menu.lst`

```
default  0
timeout  5
title    Xenomai 2.4.x (2.6.x-adeos)
kernel   (cd)/boot/vmlinuz-2.6.x-adeos rw
initrd   (cd)/boot/initramfs_data.gz
```

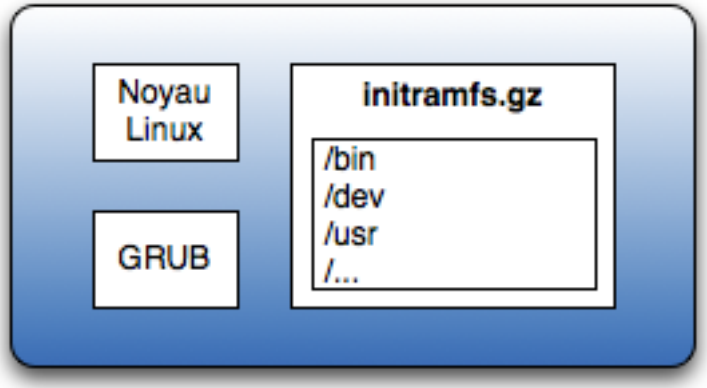
- Utilisation du stage2_eltorito

```
cp /usr/lib/grub/i386-pc/stage2_eltorito        \
   xenolive/boot/grub
```

Création de l'image bootable

- Génération de l'image bootable ISO 9660

```
mkisofs -R -b boot/grub/stage2_eltorito          \
   -no-emul-boot -boot-load-size 4               \
   -boot-info-table -o xenolive.iso xenolive
```



Tests

- Possibilité de tester avec QEMU
  - Fabrice Bellard
  - http://bellard.org/qemu/
  - Simulateur de processeurs / machines
  - Pas de comportement temps-réel
  - `qemu -boot d -cdrom xenolive.iso`
- Boot réel après gravure sur CD

Liens & Documentation

- Documentation du noyau sur initramfs
  linux-2.6.25/Documentation/filesystems/ramfs-rootfs-initramfs.txt

- CD bootable avec GNU GRUB
  http://www.gnu.org/software/grub/manual/html_node/Making-a-GRUB-bootable-CD-ROM.html

- Extension ElTorito de la norme ISO 9660
  http://en.wikipedia.org/wiki/El_Torito_(CD-ROM_standard)

Troisième partie:

Xenomai sur architecture ARM9

Sommaire

- Présentation du matériel
- Construction d'une chaîne de compilation croisée
- Compilation du noyau
- Compilation de BusyBox
- Compilation de Xenomai
- Bibliothèques dynamiques et finalisation du rootfs
- Configuration réseau (NFS / TFTP)
- Configuration de U-Boot
- Liens et Documentation

Présentation du matériel

- Carte basée sur un processeur Atmel AT91RM9200
- Ethernet 100 Mbps
- USB Host / Device
- 64 Mo de RAM
- 8 / 16 Mo de flash
- Bootloader : U-Boot

Chaîne de compilation croisée

- Crosstool 0.43 de Dan Kegel
  - Nécessite Flex et Bison
  - Versions actuelles : gcc 3.4.5 et glibc 2.3.6
  - Possibilité d'éditer le script pour choisir d'autres versions
- Test du compilateur

```
arm-softfloat-linux-gnu-gcc -o hello hello.c
```

Compilation du noyau

- Noyau Vanilla 2.6.25.x
- Patch noyau 2.6 pour AT91RM9200
- Patch noyau pour notre carte (option à activer lors de la configuration du noyau)
- Xenomai 2.4.4 (Adeos + kernelspace)

```
./scripts/prepare-kernel.sh –linux=/chemin_noyau/ --arch=arm            \
    --adeos=ksrc/arch/arm/patches/adeos-ipipe-2.6.x-arm-1.8-xx.patch

make ARCH=arm menuconfig

make ARCH=arm CROSS_COMPILE=arm-softfloat-linux-gnu- uImage
```

Compilation de BusyBox

- BusyBox 1.10.4
- Système de configuration et compilation similaire au noyau

```
make ARCH=arm menuconfig

make ARCH=arm                                          \
   CROSS_COMPILE=arm-softfloat-linux-gnu-              \
   CONFIG_PREFIX=/chemin_rootfs/rootfs install
```

Compilation de Xenomai

- Xenomai 2.4.4
- Bibliothèque et exemples
- Configuration et compilation à l'aide des Autotools

```
./configure --build=i686-pc-linux-gnu  \
   --host=arm-softfloat-linux-gnu      \
   --enable-arm-mach=at91rm9200

make DESTDIR=/chemin_rootfs/rootfs install
```

N.B. : Ne pas utiliser `--prefix=<chemin>`

Bibliothèques dynamiques & RootFS

- Bibliothèques dynamiques

```
cd /chemin_rootfs/rootfs && mkdir lib
mklibs --target arm-softfloat-linux-gnu -D                    \
    -L /chemin_crosstool/arm-softfloat-linux-gnu/lib   \
    -L usr/xenomai/lib -d lib bin/* usr/xenomai/bin/*
```

- Finalisation du rootfs

```
mkdir /chemin_rootfs/rootfs/dev
cd /chemin_rootfs/rootfs/dev
sudo MAKEDEV -v generic console
```

Configuration réseau (NFS / TFTP)

- Valider le fonctionnement de nfsd et tftp
  - portmap actif et lié à la bonne interface

    ```
    netstat -taupen | grep 111
    ```

  - Détail sur les connections en cas de problème

    ```
    tail -f /var/log/syslog
    ```

- Configuration du serveur NFS
  - Ajouter dans /etc/exports :

    ```
    /chemin_rootfs/rootfs
        *(rw,no_root_squash,no_all_squash,sync,no_subtree_check)
    ```

  - Exécuter la commande

    ```
    exportfs -a
    ```

- Copie du noyau dans le répertoire du serveur tftp

Configuration de U-Boot

- Paramètres passés au noyau

```
set bootargs root=/dev/nfs console=ttyS0,115200
   nfsroot=192.168.3.xxx:/chemin_rootfs/rootfs
   ip=192.168.3.250:::255.255.255.0:::off
```

- Spécification du démarrage par tftp

```
set bootcmd tftpboot 2000000 uImage
```

- Démarrage

```
boot
```

Liens et Documentation

- Crosstool
  http://www.kegel.com/crosstool/
- Noyau Linux
  http://www.kernel.org
- Patch AT91 pour Linux 2.6
  http://maxim.org.za/at91_26.html
- BusyBox
  http://busybox.net/
- U-Boot (mkimage)
  http://www.denx.de/wiki/UBoot/