

SYS3 - RTLinux

Pierre BETTENS
pbettens (à) heb.be

ESI - École Supérieure d'Informatique

18/09/2006

- PbT
- Contrat didactique
 - 25h
 - Evaluation
 - Examen OM (Oral Machine)
 - *à préciser pour les détails*
- Attentes
 - Compréhension des concepts temps réels
 - Utilisation de l'API POSIX RTLinux

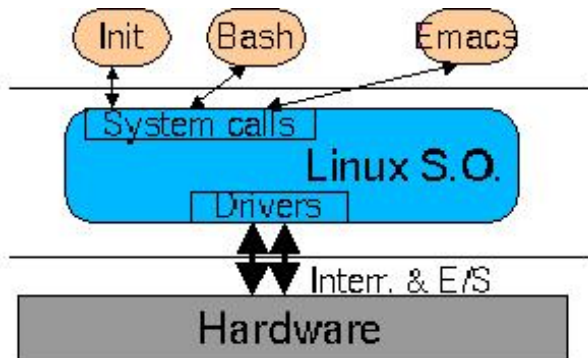
- Définition
 - Un système 'temps réel' est un système d'information dont les corrections ne dépendent pas uniquement du résultat logique des algorithmes mais aussi de l'instant où ces résultats ont été produits."
 - *Un résultat juste mais hors délai est un résultat faux*
- Position du problème
- Exemple d'utilisation
 - Aviation, aérospatial
 - Sciences
 - Multimédia
 - Télécommunication
 - ...
 - <http://fsmlabs.com>

- OS vs real time OS
 - Doit être simple
 - Doit être optimisé
 - Doit être prévisible
 - Capable de répartir les temps entre les *threads*
 - Offrir un mécanisme de communication entre les processus (sémaphores, ...)
 - Gérer les interruptions
 - Preemption

- Justification du choix
 - Licence GPL et The Open RTLinux Patent License,
 - Linux, environnement connu et stable
 - Il existe une version gratuite ... permet les tests
 - API est documentée
 - Simple à mettre en oeuvre
- Installation
 - Installation d'un noyau patché
 - Configuration et compilation
 - Configuration de RTLinux et installation
 - Chargement des modules RTLinux

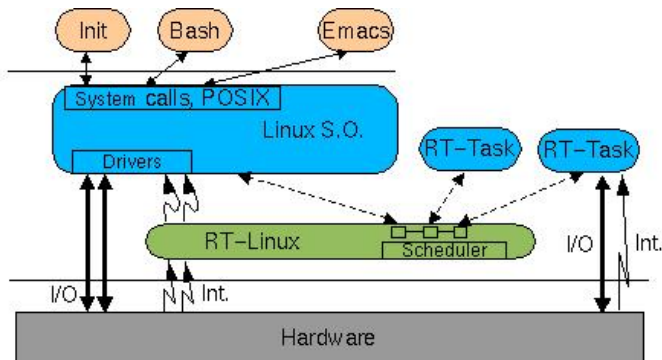
RTLinux - OS temps réel

Schémas de fonctionnement



RTLinux - OS temps réel

Schémas de fonctionnement



RTLinux - OS temps réel

Notions de modules

- Définition
 - Un module est un fragment de l'OS que l'on peut ajouter ou soustraire au moment de l'exécution.
- Exemples
 - Accès au *devices*
 - Support USB, ...
- Création
 - `init_module`
 - `cleanup_module`
- Commandes associées
 - `insmod`
 - `lsmod`
 - `rmmmod`
 - `dmesg`

RTLinux - OS temps réel

Notions de modules

- *Modus operenti*
 - Ecriture du code C (`module.c`)
 - Compilation (`module.o`)
 - Insertion du module dans le noyau
- Un module chargé fait partie intégrante du noyau
 - Il peut utiliser toutes les **fonctions** et accéder à toutes les **variables** du noyau
 - Le code du module est exécuté avec les **privilèges** maximum
 - Il n'est pas possible de paginer la mémoire réservée pour le module.

- Structure d'un module

```
#define MODULE
#include <linux/module.h>
#include <linux/cons.h>

int init_module(void) {
    // code execute a l'insertion
    return 0;
}

void cleanup_module(void) {
    // code execute lors rmmod
}
```

- Définition
 - Des *threads* sont des processus légers qui partagent un espace d'adressage commun
- Mise en oeuvre (utilisation de l'API standard)
 - `pthread_create()`
 - ...
- Synchronisation entre *threads*
 - Utilisation des sémaphores, *mutex*
- Communication entre *threads*
 - Utilisation des pipes, fifos
 - Définition
 - Tampon d'octet de taille fixe sur lesquels on peut lire/écrire
 - Même principe qu'un *pipe*

- Un programme temps réel est un module chargé dans le noyau
- Un module est un programme C, compilé, .o
 - `int init_module() ;`
 - `void cleanup_module() ;`
- Ce programme est composé de un ou plusieurs *threads*

- **freemind** - <http://freemind.sourceforge.net>
Génération d'un Mind Map
- Génération des slides sur base du Map freemind
 - Scripts Perl
 - freemind2beamer.pl
 - Format PDF
 - **L^AT_EX**
 - package *beamer*