Développement d'un driver de périphériques sous l'environnement RTLinux

Master : Informatique Industrielle

Université de Kasdi Merbah d'Ouargla
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique et des Technologies de l'Information
Présenté par : DJERROUDI Linda
linda.djerroudi@gmail.com
Dirigé par : M. Herrouz A.

INTRODUCTION

Le temps réel est une notion qui prend de plus en plus d'importance. Un exemple simple : on ne compte plus le nombre de sites Internet qui nous envoient, sur notre Smartphone, de l'information en temps réel.

Ce projet de recherche consiste à mener un développement en interaction forte avec le matériel sous GNU/Linux en tant que système embarqué. Dans ce but , nous avons adopté le Système d'Exploitation (SE) RTLinux qui utilise un noyau linux et s'appuie sur une architecture bien spécifique afin de supporter l'approche temps-réel.

DÉFINITIONS

- **Système temps réel :** système (matériel et logiciel) qui satisfait aux contraintes fonctionnelles et temporelles qui lui sont imposées.
- la notion de temps réel implique un comportement du système déterministe , i.e. il doit respecter le déterminisme logique (exactitude du résultat) et temporel (résultat obtenu dans le temps imparti).

Système temps réel dur

- Def : système devant respecter des échéances avec une tolérance nulle.
- utilité nulle des résultats obtenus après échéance.
- pénalité élevée : panne complète du système + danger pour l'intégrité physique du système ou des êtres humains.

VS

Systèmes Temps réel mou

- Def : système devant satisfaire des échéances temporelles mais avec un degré de flexibilité, tolérance sur l'échéance.
- une échéance non satisfaite : pas de destruction du système, a une pénalité relative au retard sur l'échéance (date limite)

La différence entre un système temps réel dur et mou est la pénalité en cas de non-respect de l'échéance.

- **Un SE** :
  - Ensemble de programmes assurant la gestion pour le partage des ressources attachées au CPU (dont le CPU)
  - Le cœur de ces programmes est le **noyau** qui regroupe les fonctionnalités les plus importantes
  - Le but principal d'un SE est d'interagir avec le matériel et permettre l'utilisation des composants de bas niveau

- **Un SE temps réel** : il répond aux caractéristiques suivantes :
  - une politique d'ordonnancement "temps réel" pour résoudre la concurrence des tâches pour le CPU ;
  - un temps de réponse (**latence**) assez court pour l'application visée.

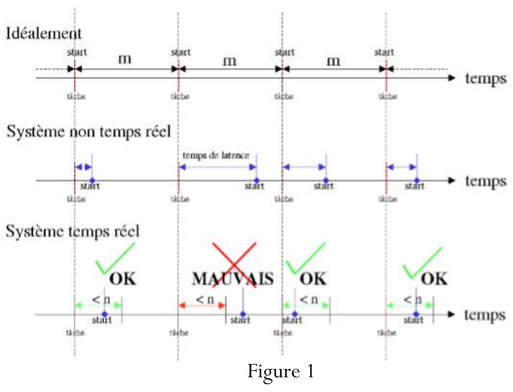

Figure 1

- **Ordonnancement préemptif :**
  - assurer l'ordonnancement des tâches et leurs échéances
  - **préemption** : capacité à interrompre une tâche au profit d'une autre
  - **priorité fixe** : chaque tâche a une priorité d'exécution invariante
  - **règle** : exécution de la tâche de plus haute priorité non-bloquée à chaque appel à l'ordonnanceur.

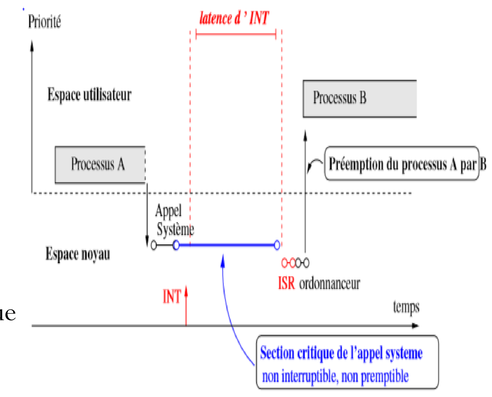

Figure 2

PROBLÉMATIQUE

- **Pourquoi linux ?**

Linux est considéré comme un système fiable, robuste et puissant. il est multiutilisateurs (plusieurs personnes peuvent travailler sur le même ordinateur), mais aussi multitâche (plusieurs applications peuvent être lancées en même temps sans qu'aucune n'affecte les autres). Il est également multiplateforme. Il peut être installé sur : x86, Sparc, PowerPC, Alpha et même sur les consoles de jeu, et il est aussi multiprocesseurs.

OR Linux reste un SE généraliste et il n'est pas conçu pour fournir un comportement temporel déterministe , c.à.d. qu'il n'est pas temps réel . Par ailleurs comme Linux est un noyau offrant beaucoup d'avantages (gratuit et open source , stable , bon temps de réponse moyen en situation de fortes charges, …), des solutions pour qu'il adosse le temps réel ont rapidement vu le jour en 1996 et n'ont pas cesser d'évoluer depuis.

Dans ce but , les développeurs ont adopté 2 approches :

- ✓ Le co-noyau (RTLinux , RTAI , Xenomai, …)
- ✓ Noyau complètement préemptible (patch RT_PREEMPT)

SOLUTION

- Parmi les premières solutions Système d'Exploitation temps réel utilisant l'architecture micro-noyau sous un noyau linux figure RTLinux.
- RTLinux est une extension temps-réel dur du noyau Linux basé sur l'architecture µ-noyau, il met en avant le fait d'exécuter le noyau Linux en tant que tâche entièrement préemptible.

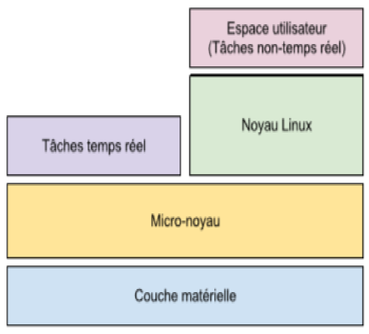

Figure 3

DÉMARCHE ENTREPRISE

le but du projet consiste à proposer une solution qui prend complètement en charge le temps-réel. Pour ce faire, nous entrevoyons :

1. L'installation de la version RedHat 8.0 de Linux et son noyau 2.4.18, ainsi que RTLinux 3.2 et son noyau auxquels une partition mémoire a été réservée sur la machine.
2. Ensuite , on procède à **patcher** le noyau RTLinux sous le noyau de linux en utilisant les directives adéquates.

Ceci nous permet de construire un nouveau noyau différent de celui de linux avec un gestionnaire de tâches. Le noyau Linux tourne indépendamment au dessus de ce µ-noyau et partage le CPU avec les autres tâches temps réel.

Plus précisément, Linux devient une tâche en arrière plan, ne s'exécutant que lorsqu'aucune tâche temps réel n'est active.

3. Apres avoir patcher, on recompile le système et redémarre le système.

Cette approche va aboutir à la production d'un nouveau système apte à contrôler toute sorte de machinerie industrielle, comme les robots, les systèmes d'acquisition de données et d'autres instruments et machines soumises au facteur du temps.

En guise de mise en œuvre de l'environnement obtenu, nous allons construire un driver de pilote simple, montrer l'utilisation des ressources RTLinux et écrire les programmes de test pour ce pilote.

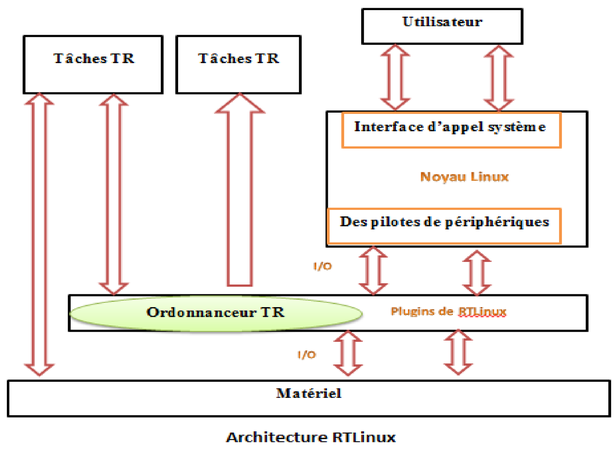

Figure 4

CONCLUSION

Linux est non seulement une parfaite plateforme pour l'expérimentation et caractérisation des algorithmes temps réel, mais il a su aussi supporter le temps réel dans ses deux formes.

Notre travail présente une mise en œuvre complète de ce pouvoir de linux, utilisant sa version RedHat et reposant sur l'environnement RTLinux.

Compte tenu de ce qui précède, Linux peut se prévaloir d'être un concurrent pour de nombreux systèmes d'exploitation Temps Réel.

RÉFÉRENCES

1. Comment Concevoir et Réaliser un Noyau Temps Réel? Mémoire d'un projet de fin d'études. Université Mouloud MAMMERI de Tizi-Ouzou. Présenté par M. Toufik SARNI. Dirigé par Mme Samia Bouzefrane (CNAM – Paris) & M. Lounes Djema (UMMTO)
2. FERRE, N., (2002). *Linux temps réel.* URL : http://nferre.free.fr/rtl/linux-rt_sl_r.pdf.
3. P. E. McKenney. (2011) RCU Concepts. [en ligne]. URL: http://www.ibm.com/developerworks/library/l-real-time-linux/
4. http://fr.slideshare.net/mounayangui/rtlinux

LISTE DES FIGURES

1. **Figure 1:** Tâche périodique et temps réel .
2. **Figure 2:** La préemption dans l'espace noyau.
3. **Figure 3:** l'approche Micro-noyau .
4. **Figure 4:** l'Architecture RTLinux.