Rapport de stage Ingénieur

-

Implémentation d'un ordonnanceur temps réel sur plateforme multi-cœur hétérogène

-

BELPOIS Vincent

2023

Table des matières

Présentation du stage

0.1 Le L.I.A.S.

0.2 Le sujet du stage

Mon stage s'intéresse à l'implémentation d'un ordonanceur sur plateforme hétérogène[1]

1 Compatibilité de différents systèmes d'exploitation avec la carte ROCK960

1.1 Présentation de la carte de développement

1.2 Etude des versions de Linux compatibles

2 LITMUS^{RT}

2.1 Présentation de LITMUST^{RT}

2.2 Présentation de *feather-trace*

2.3 Implémentation d'un ordonanceur EDF partitioné

2.3.1 Algorithme considéré

On cherche alors pour commencer a implémenter un algorithme d'ordonancement simple afin de se familiariser avec les méthodes et fonctions fourni par LITMUS^{RT}. J'ai donc choisi un algorithme partitioné pour la simplicité d'ordonancement par processeur que cela offre. Un algorithme EDF (*Earliest Deadline First*) est alors choisi pour la simplicité du choix de la tache a exécuter. Comme son nom l'indique, on choisi à chaque instant la tache ayant l'échéance la plus proche. On nomera par la suite cet algorithme P-EDF (*Partitionned Earliest Deadline First*).

Pour montrer le fonctionnement de cet algorithme, si l'on se place sur un même processeur, on peut visualiser l'éxécution de deux tache periodiques :

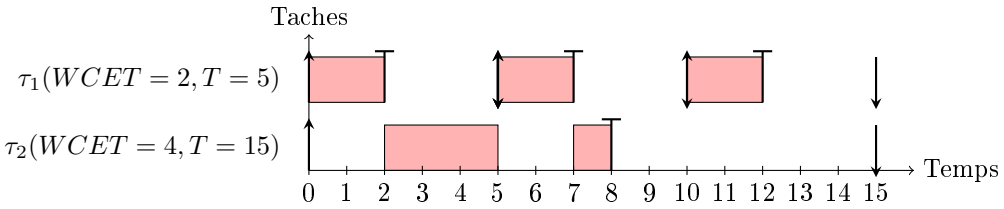

FIGURE 1 – Exemple de EDF à 2 taches

On a ici une première tache τ_1 avec un pire temps d'éxécution (*Worst Case Execution Time*) de 2 et une période de 5, et une seconde tache τ_2 avec un pire temps d'éxécution de 4 et une période de 15. On a alors préemption de la τ_2 à $t = 5$ afin d'éxécuter τ_1. Cela est dû au réveil de la tâche τ_1 (représenté par la flêche montante) et a la date d'échéance plus proche de cette dernière.

2.3.2 Implémentation

Expliquer ce qu'est un module dans le noyau linux.
- Montrer ce qui est propre a la définition du module
- Montrer l'emplacement des fichiers que l'on va créer dans le noyau
- Montrer les modification du make file

```c
#include <stdio.h>

int main() {
    printf("Hello, world!");
    return 0;
}
```

$$\frac{a}{b} = 1,3$$

Références

[1] Antoine Bertout, Joël Goossens, Emmanuel Grolleau, and Xavier Poczekajlo. Workload assignment for global real-time scheduling on unrelated multicore platforms. In *Proceedings of the 28th International Conference on Real-Time Networks and Systems*, pages 139–148, 2020.

Table des figures

Glossaire

processeur Ca c'est la définition. 3