HEIG VD | REDS Institut Reconfigurable and Embedded Digital Systems

Alexandre Corbaz, Catel Torres Arzur Gabriel

Laboratoire de Programmation Temps Réel

semestre printemps 2023 - 2024

Laboratoire 5 : Traitement des données

Temps à disposition : 6 périodes

Le laboratoire doit être effectué par groupe de 2 personnes.

Objectifs

L'objectif de ce laboratoire est de développer un système qui, utilisant la plateforme DE1-SoC, est capable de capter et de traiter des données audio et vidéo afin d'en extraire des informations utiles.

Nous examinerons et analyserons également les différents temps d'exécutions des tâches afin d'étudier et d'optimiser les performances temps réel et l'implémentation de celles-ci.

Préconditions au laboratoire 5

Afin d'augmenter le déterminisme des exécutions des programmes et de supporter l'analyse, sauf indication contraire, tous vos programmes devront être executés sur un seul CPU grâce à la commande `taskset`.

Mise en route de la carte DE1-SoC

N'oubliez pas de load les différents modules lorsque vous redémarrez la board, voici une commande pour faciliter le setup :

- `insmod /usr/custom_modules/de1_ioctl_module.ko`
- `insmod /usr/custom_modules/de1_audio_module.ko`
- `insmod /usr/custom_modules/de1_video_module.ko`

Etape 1 : Audio - Traitement et logging

✠ Le système à réaliser devra être capable de récupérer les données audio en entrée, d'appliquer une FFT sur ces données, et d'afficher des informations pertinentes, notamment concernant la fréquence principale. Pour ce faire, nous allons le décomposer en différentes tâches :

1. Tâche pour l'acquisition des données
2. Tâche pour le traitement des données
3. Tâche pour le monitoring/logging

Le flux de données va donc aller de la tâche d'acquisition vers la tâche de traitement, puis du traitement au monitoring. A vous de mettre en place la communication en utilisant une queue de messages temps réel.

La tâche d'acquisition doit être périodique, les autres tâches dépendent directement de la réception des données, il est donc recommandé de mettre en place des tâches qui ne sont pas périodiques pour traiter les données.

Le logging à mettre en place est relativement simple. Il faut afficher dans la console la fréquence fondamentale. La tâche de logging ne doit pas faire de traitement, simplement recevoir des messages qu'elle doit ensuite afficher.

✠ Enfin, vous utiliserez le switch 0 pour activer ou désactiver l'acquisition et le traitement du flux audio.

Etape 2 : Vidéo - Traitement

✠ Le code fourni vous propose une implémentation pour l'acquisition. L'objectif est d'adapter le code pour avoir une solution de traitement modulable. Pour la vidéo, nous vous demandons de mettre en place 3 manières de gérer les données à afficher :

— Affichage de la vidéo telle qu'elle est récupée du fichier, sans modifications.
— Affichage de la vidéo en grayscale.
— Affichage de la vidéo après qu'une convolution soit appliquée.

Ces différents modes doivent être gérés avec les switchs 8 et 9 de la board. Les switchs doivent être utilisés comme des entrées binaires permettant de choisir les modes dans l'ordre mentionné ci-dessus.

Dans le dossier `utils/`, vous trouverez des fichiers supplémentaires permettant d'utiliser les différents filtres nécessaires. Le nom des fichiers est assez explicite, nous vous laissons en prendre connaissance.

Nous vous demandons de gérer la synchronisation des tâches différemment de la méthode que vous avez utilisé pour la partie audio. Pour la vidéo, vous devrez gérer la synchronisation avec des "event".

Il est important que la tâche d'acquisition ne soit jamais polluée par le traitement des données. Il faut assurer que cette tâche récupère toujours les données à la fréquence souhaitée (15HZ). Si le traitement est trop demandant en ressources, cela ne doit pas avoir d'impact sur l'acquisition.

✠ Enfin, vous utiliserez le switch 1 pour activer ou désactiver la partie vidéo.

Etape 3 : Caractérisation des tâches et condition d'ordonnançabilité

Pour ce laboratoire, je vous demande de mesurer les performances de vos tâches et de prouver formellement l'ordonnançabilité de celle-ci avec la théorie vue au chapitre 5 sur l'ordonnancement RM (Rate Monotonic).

Pour ce faire, vous allez caractériser de manière individuelle le temps d'exécution du groupe des tâches pour l'audio, du groupe des tâches pour la vidéo et de la tâche de traitement des switches. Le but est de déterminer à partir de la distribution mesurée le pire cas du temps d'exécution.

- Nous allons considérer pour l'audio la tâche pour l'acquisition des données, la tâche pour le traitement des données et la tâche pour le logging comme une unique tâche. C'est une hypothèse valable sachant que les trois tâches ont une relation de précédence et la coordination est faîte à l'aide de mailbox.
- Nous allons faire les mêmes hypothèses pour la vidéo sachant que les trois tâches ont une relation de précédence et la coordination est faîte à l'aide de drapeaux (event). N'oubliez pas de caractériser la vidéo pour les différents traitements des données demandés (no processing, grayscale, convolution).

Libre à vous d'utiliser les méthodes de mesure que nous avons utilisées lors des précédents laboratoires.

✓ Veuillez documenter vos mesures (méthode, nombre de mesures, niveau de confiance, etc...) et l'approche utilisée afin de déterminer le pire temps d'exécution.

Etape 4 : Mise en œuvre et condition d'ordonnançabilité

Nous connaissons maintenant :

- le temps d'exécution des tâches
- les périodes que vous avez spécifié pour l'audio et les switches.

✓ Afin que notre ensemble de tâches soit ordonnançable avec l'algorithme Rate Monotonic, quelles sont les périodes min. pour la vidéo (avec les différents traitements des données) en appliquant la condition d'ordonnançabilité proposée par Liu et Layland.

✠ Veuillez mettre en œuvre votre système en respectant l'algorithme Rate Monotonic avec les priorités et les conditions précédemment déterminées.

✓ Quelle observations faîtes-vous pour les différents traitements des données vidéos ?

- Quelle est la charge du CPU retournée par la commande htop ?
- Votre système est-il capable de gérer les différents traitements vidéo de manière ordonnée ?
  - Si oui, quel est le framerate vidéo maximal auquel votre système continue de fonctionner correctement ?
  - Si non, dans quelle mesure devez-vous diminuer le framerate vidéo pour obtenir un système fonctionnel ?

Travail à effectuer

⚠ Le code que vous avez développé lors de l'étape 2, l'étape 3 et l'étape 4 est celui à rendre. Le code final devra être correctement commenté.

Tout le code dédié lors de l'étape 3 pour caractériser les tâches ne doit pas être rendu. L'approche utilisée à l'étape 3 afin de déterminer le pire temps d'exécution devra être documentée du le rapport.

Un rapport au format PDF vous est également demandé. Il devra contenir les réponses aux questions directes (indiquées par le symbole ✓), vos choix architecturaux ainsi que les tests effectués.

— Rendez le tout sur cyberlearn, dans un fichier compressé nommé `rendu.tar.gz`
  — Ce fichier doit être généré en lançant le script `ptr_rendu.sh`
  — Le script vérifie qu'un fichier `rapport.pdf` est présent à son niveau, ainsi qu'un dossier `code` également présent au même niveau.