

Laboratoire de Programmation Temps Réel semestre printemps 2023 - 2024

Laboratoire 6 : Gestion de surcharge

Temps à disposition : 4 périodes

Le laboratoire peut être effectué par groupe de 2 personnes.

Objectifs

Le but de ce laboratoire est d'ajouter un système de gestion de surcharge au labo précédent. Les spécifications pour la gestion de surcharge sont les suivant :

- Votre système devra être capable de détecter la surcharge, c'est-à-dire, détecter que les tâches du traitement de la vidéo et de l'audio ont pris plus de temps que prévu et qu'une échéance a été manquée.
- Le traitement du flux vidéo devra effectuer le traitement grayscale implémenté au labo précédent. Cependant, en fonction d'une detection de surcharge temporaire, votre système implémentera dynamiquement une version dégradée du frame rate vidéo permettant ainsi de gérer la surcharge.
- Le traitement du flux audio ne devra en aucun cas être dégradée lors d'une surcharge du système. Notre système n'effectuera plus de FFT sur les données mais il devra tout mettre en oeuvre pour assurer le maintien du flux audio du microphone au speaker. Aucune gestion d'une surcharge de l'audio est à mettre en oeuvre pour ce laboratoire.

Etape 1 : Observation de la surcharge

Pour ce laboratoire, vous pouvez récupérer votre code développé pour le laboratoire précédent et le simplifier pour qu'il soit conforme aux attentes du laboratoire, ou bien utiliser la base que nous vous fournissons.

Le switch 0 permet de générer de la surcharge CPU grâce à une tâche de priorité 90. Si le switch est actif, alors la surcharge est exécutée, et s'il est inactif elle ne l'est pas.

Essayez votre code pour voir ce qui se passe en cas de surcharge. ⚠ Evidemment ceci part du principe que vos tâches ont une priorité inférieure à 90.

Dans le fichier `main.c`, le mécanisme de surcharge est déjà mis en place et fonctionnel. Vous avez des macros définies permettant de gérer à votre convenance le système de surcharge. La surcharge fonctionne de la manière suivante :

- `SW0` : active/désactive la surcharge CPU
- `KEY2` : augmentation de la charge CPU d'une valeur définie
- `KEY3` : diminution de la charge CPU d'une valeur définie

Trouvez la valeur minimal permettant de faire sauter le watchdog Xenomai en utilisant cette fonctionnalité.

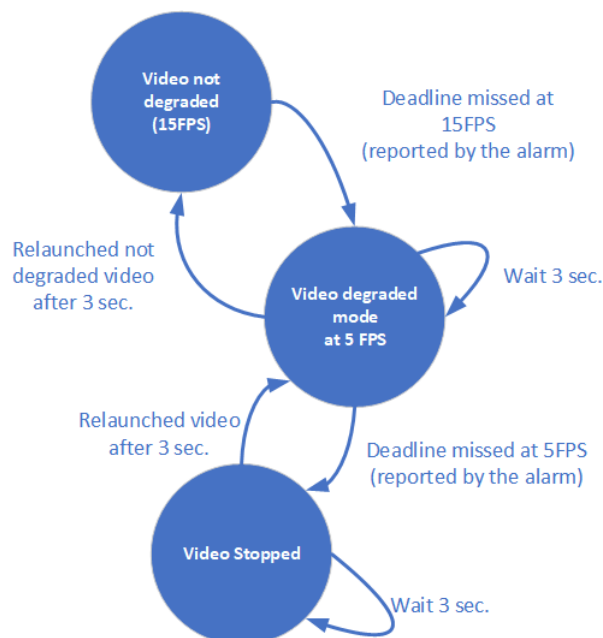
Etape 2 : Gestion de la surcharge avec une alarme

Pour la gestion et la détection de la surcharge, nous allons exploiter les alarmes offertes par Xenomai. Il faudra créer une alarme `one-shot` qui sera lancée (avec `rt_alarm_start`) au début du traitement de la vidéo et sera arrêtée (avec `rt_alarm_cancel`) si les tâches de la vidéo se terminent dans les délais. Si l'alarme n'est pas annulée avant son expiration, le handler sera appelé, indiquant qu'une échéance a été dépassée.

Assurez-vous que le `timeout` de votre alarme est légèrement supérieur à la période pour donner suffisamment de marge et éviter les fausses alertes dues à des variations mineures dans le temps d'exécution. De plus, n'oubliez pas que vous devrez considérer à ajuster la valeur de `timeout` pour le traitement de la vidéo qui aura des modes dégradés (voir étape 3).

Etape 3 : Fonctionnalité dégradée

La fonctionnalité dégradée de la vidéo sera basée sur le frame rate de la tâche pour l'acquisition des données. Cette approche devrait clairement diminuer la charge CPU.



Comme indiqué dans la figure ci-dessus, lors de la détection de surcharge, votre système activera en un premier temps un mode dégradé (`video_degraded_mode`) où le frame rate sera à 5FPS. Si la détection de surcharge est à nouveau détectée depuis le mode dégradé, la vidéo sera stoppée (`video_stopped`).

Lorsque votre système est dans le mode `video_degraded_mode` ou le mode `video_stopped`, celui-ci devra attendre 3 secondes avant de retourner dans l'état précédent. Vous pouvez utiliser à nouveau une alarme pour calculer ce temps passé dans un des modes.

Travail à effectuer

A vous de développer le code correspondant aux différentes étapes. Le code final devra être correctement commenté.

Un petit rapport présentant vos choix architecturaux ainsi que les tests effectués vous est également demandé.

- 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