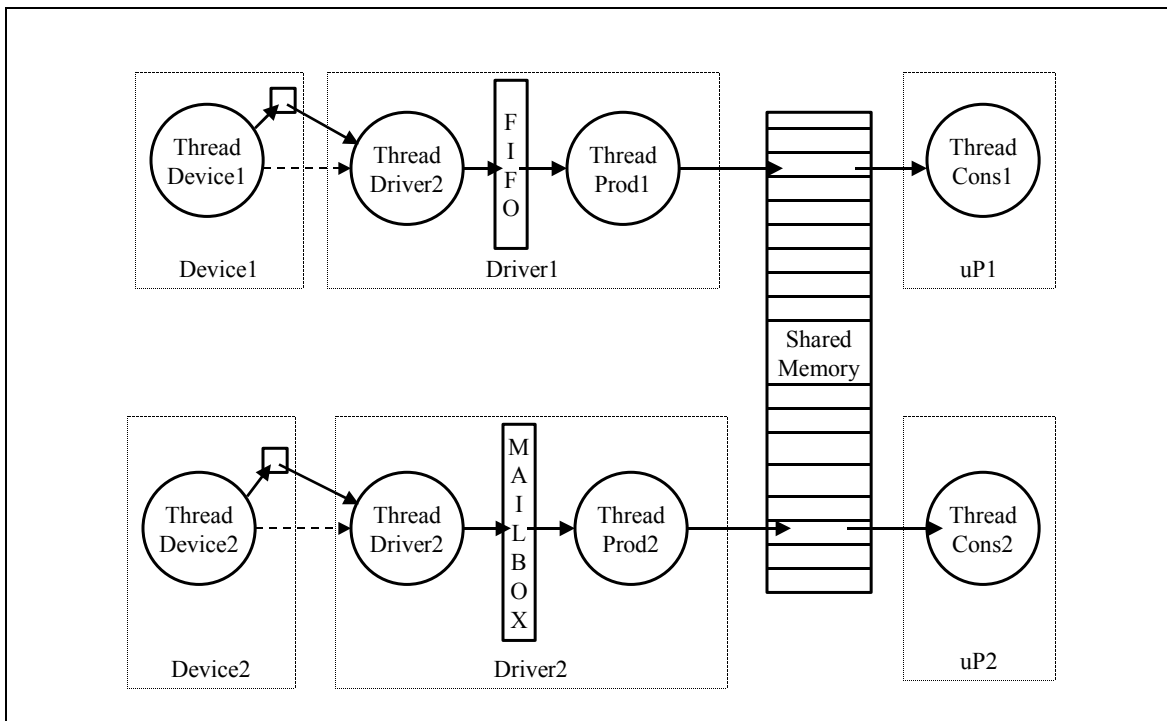


Exemple μ C

Description

Vous devez programmer un ensemble de Threads qui simulera l'échange de données entre deux périphériques et deux microprocesseurs. Les Threads devront être implantés conformément au schéma ci dessous.



Réalisation

En utilisant le RTOS micro C, vous devez réaliser 8 Threads (Task) :

- Deux Threads simulant les périphériques (device). Chacun des Threads produit régulièrement une série de valeurs (espacées-elles mêmes régulièrement) (utilisez deux Timer) qu'il place dans une variable partagée non-protégée (pas de mutex) sensée représenter un registre. Il envoie ensuite un évènement (simulant une interruption) au Thread simulant le pilote (driver). Pour réaliser cet évènement vous devez utiliser un semaphore ou les flags que vous placerez de manière à créer un Rendez-vous unilatéral (inspirez-vous du livre p 58-60).
- Deux Threads simulant les pilotes (drivers) à bas niveau. Chacun des Threads reçoit l'évènement ce Thread lit la valeur contenue dans la variable partagée (registre) et l'envoie dans un FIFO ou une MailBox (suivant le schéma). Ensuite il se remet en attente d'un évènement.
- Deux Threads producteurs simulant les pilotes (drivers) à haut niveau. Chacun des Threads reçoit une valeur par l'intermédiaire de son FIFO ou de sa MailBox, ce Thread la récupère et la l'écrit dans une mémoire partagée commune (avec le Thread de l'autre driver et les Threads consommateurs). Utilisez l'algorithme du Bounded Buffer (Producer-Consumer).
- Deux Threads consommateurs simulant des microprocesseurs. Chacun des Threads consomme de manière régulière les données contenues dans la mémoire partagée. Utilisez un Timer pour simuler le temps de consommation. Vous devez réaliser également un affichage permettant de suivre l'évolution du programme (pas seulement les variables fournies par le Thread de statistique). Vous pouvez utiliser une structure de message lorsque vous passez chaque valeur dans le FIFO, la MailBox et la mémoire partagée, afin d'y insérer des informations nécessaires au suivi de la donnée.

Une fois votre programme terminé, jouez avec les priorités, les Timers, la taille du Fifo et de la mémoire partagée. Tirez-s'en ensuite certaines constations.