

INFORMATIQUE DU TEMPS REEL

Manipulation « Caractéristiques temporelles de Windows XP »

29 janvier 2007

Objectifs

Cet ensemble de manipulations permet d'observer certains aspects du comportement temporel de Windows XP.

Outils et recommandations générales

Vous disposez d'un PC qui va vous servir pour exécuter vos programmes (la machine observée) et d'un PC que vous allez utiliser comme analyseur logique. Les deux sont liés par un câble comportant deux prises mâles de type subD à 25 broches. La petite carte d'adaptation est à mettre du côté du PC observé. Dans la première partie du laboratoire (manipulations 1 à 4), le PC observé doit être un PC ayant un identificateur (étiquette en haut à droite du moniteur type ICBC07PCxx) pair. L'analyseur doit utiliser un PC ayant un numéro impair.

Les fichiers dont vous aurez besoin sont disponibles dans le répertoire \\icwfile\Studdata\RealTime\. Copiez le répertoire ManipsNTTR sur la racine de votre disque local. Vous trouverez à l'intérieur de ManipsNTTR un répertoire par question de cette manipulation. S'y trouvent les fichiers dont vous avez besoin, dont les fichiers manipN.sln qui sont les fichiers de projet à ouvrir avec Visual .NET.

L'application RTLancher (dans C:\ManipsNTTR\RTLancher) est une application qui permet de lancer une autre application en priorité Temps Réel. L'utilisation de cette application est simple. Lancez l'application depuis un « Explorer » en double cliquant sur l'icône « RTLancher », puis tapez le nom de la manip que vous voulez lancer, par exemple « manip1 » pour lancer la manip1 en temps réel (par défaut les projets visual studio ont été configurés pour que les exécutables soient créés là où se trouve RTLancher).

Afin d'éviter les risques de blocage de la machine, essayez toujours votre application en la lançant directement (pas depuis Visual Studio) avant de la lancer avec le RTLancher

Pour démarrer l'analyseur, il faut « rebooter » le PC et presser F8 quand vous y êtes invité. Choisir l'option « boot disquette temps reel ». Le système démarre alors en mode DOS. Dès que vous avez l'invite, tapez la commande logic.exe. Cela lance l'analyseur.

L'analyseur permet de visualiser l'état de 5 des 8 lignes du port parallèle d'un PC. Dans les programmes que vous allez créer ou utiliser, vous trouverez des appels à WriteParallelPort. Le dernier paramètre de cet appel correspond à la valeur qui est mise à la sortie du port parallèle. La correspondance entre les bits de cette valeur et les canaux de l'analyseur est la suivante :

Valeur en C++	Ligne de donnée sur port parallèle	Canal sur analyseur
0x01	0	1
0x02	1	0
0x04	2	2
0x08	3	3
0x10	4	4

L'analyseur vous permet de choisir les paramètres de fonctionnement :

- la fréquence d'échantillonnage (sample rate) en nombre d'échantillons par seconde (commande R ou r). Utiliser la valeur maximale pour la manipulation.
- le mode (condition) de déclenchement (commande T ou t)
- le canal sur lequel le mode de déclenchement (trigger input) est appliqué (commande I ou i)
- le nombre d'échantillons (trigger duration) pendant lequel le signal doit rester stable après que la condition de déclenchement a été trouvée (commande d ou D)

La mesure proprement dite est démarrée par la commande S ou s (Start/Stop). Le logiciel indique alors si la condition de déclenchement a été trouvée ou non. Dès que le nombre d'échantillons a atteint la taille du tampon, le message « buffer full » est affiché. Vous pouvez arrêter la mesure avant cela en utilisant la commande S (ou s). L'affichage du contenu du tampon peut être obtenu par la commande V ou v (View Trace). Vous avez alors diverses commandes qui permettent de modifier l'affichage (échelle, début). Notez que dès que le tampon est plein, le système n'autorise plus de mesure. Il faut vider le tampon avant toute nouvelle mesure par la commande C ou c (Clear buffer).

Il arrive que l'analyseur bloque complètement. La seule solution est alors d'appuyer sur le reset du PC et de rebooter selon la procédure indiquée plus haut.

Manipulation 1 :

Utilisez le projet «Manip1 » dans le dossier « C:\manipsNTTR\manip1 ». Ce projet consiste en une application qui démarre un thread qui exécute une boucle incrémentant un compteur. On cherche ici à mettre en évidence la différence de comportement entre l'exécution du même programme en priorité temps réel et en priorité non temps réel.

- Observez la structure du programme et les primitives utilisées pour créer et démarrer un thread.
- Exécutez le programme directement. Que se passe-t-il lorsque vous déplacez la souris ?
- Exécutez le programme en utilisant le RTLaucher. Que se passe-t-il ? Pourquoi ?

L'erreur « Could not open Parallel Port! » signifie que le driver du port parallèle n'a pas été installé. Il suffit alors de copier le fichier portio.sys (qui se trouve dans le répertoire ManipsNTTR) dans le répertoire C:\WINDOWS\system32\drivers\ puis de localiser le fichier TempReelPortIo (qui se trouve dans le même répertoire de portio.sys) de cliquer sur ce fichier avec le bouton de droite de la souris et d'utiliser la commande « merge ». Il faut ensuite redémarrer l'ordinateur. Tout devrait alors rentrer dans l'ordre.

Manipulation 2 :

Utilisez le projet «Manip2 » dans le dossier « C:\manipsNTTR\manip2 ». Le programme correspondant démarre un thread qui exécute une boucle faisant changer l'état (« toggle ») de la ligne 2 de données du port parallèle. L'objectif est de constater que votre thread peut cesser de s'exécuter durant quelques millisecondes. Vous aurez ici besoin de l'analyseur logique (**paramètres de l'analyseur: rate max, input 2, level 1, duration 20**)

- Observez la structure du programme et les primitives utilisées.
- Observez la fonction d'exécution du thread et prévoyez quel sera le comportement du programme.
- Exécutez le programme directement. Que se passe-t-il ? Qu'observe-t-on à l'analyseur ?
- Maintenant il faut programmer l'analyseur pour qu'il ne se déclenche que si la ligne de donnée 2 ne change pas pendant un temps supérieur à 1ms. Que se passe-t-il ?
- Exécutez le programme en utilisant le RTLaucher. Que se passe-t-il ? Pourquoi ?

Manipulation 3 :

Utilisez le projet «Manip3 » dans le dossier « C:\manipsNTTR\manip3 ». Ce projet consiste en une application qui démarre 2 threads s'exécutant en parallèle. Ces deux threads partagent un Sémaphore qui gère l'accès au port parallèle du PC. Le but de cette manipulation est d'observer quelle est la politique utilisée par XP pour gérer les files d'attente des sémaphores. On peut déterminer quel est le dernier thread exécuté en regardant l'état des lignes de données 2 et 3 du port parallèle. Si la ligne de données 2 est active le dernier thread qui s'est exécuté est le thread 1. Si la ligne de données 3 est active, c'est le thread 2 qui a été exécuté en dernier. L'objectif est de déterminer la politique d'octroi des sémaphores. (Paramètres de l'analyseur: rate 50000, input 3, rising edge, duration 1)

- Observez la structure du programme et les primitives utilisées.
- Observez la fonction d'exécution du thread et prévoyez quel sera le comportement du programme.
- Exécutez le programme directement. Que se passe-t-il ? Qu'observe-t-on à l'analyseur ?
- Exécutez le programme en utilisant le RTLaucher. Que se passe-t-il ?
- Modifier les priorités relatives des threads. Quel changement observez-vous sur l'analyseur ?
- Modifier l'ordre dans lequel les threads demandent le sémaphore. Quel changement observez-vous sur l'analyseur ?
- Peut-on changer la priorité après que le thread est bloqué en attente du sémaphore ?

Manipulation 4 :

Utilisez le projet «Manip4 » dans le dossier « C:\manipsNTTR\manip4 ». Ce projet consiste en une application qui démarre 3 threads s'exécutant en parallèle. Deux de ces 3 threads (thread1 et thread2) se partagent un sémaphore qui gère l'accès au port parallèle du PC. Le troisième thread (thread3) simule une tâche avec des calculs intensifs. Les trois threads sont à la même priorité. Le but de cette manipulation est de constater l'influence du « time-slice » de XP sur l'exécution des threads (paramètres de l'analyseur: rate 10000, input 2, rising, duration 1)

- Observez la structure du programme et les primitives utilisées.
- Observez la fonction d'exécution du thread et prévoyez quel sera le comportement du programme.
- Exécutez le programme directement. Que se passe-t-il ? Qu'observe-t-on à l'analyseur ?
- Exécutez le programme en utilisant le RTLauncher. Que se passe-t-il ? Pourquoi ?
- modifier le programme de manière à obtenir un comportement prévisible.

Manipulation Multithreaded :

Vous allez maintenant observer l'influence du « multithreading » sur le comportement des programmes. Pour ce faire, vous allez permuter les rôles des 2 PC que vous utilisez. Rebooter le PC impair avec Windows XP et le PC pair avec l'analyseur (cf. instructions plus haut). Veuillez aussi permuter le câble parallèle en mettant la petite carte du côté du PC tournant sous Windows. Ce dernier fonctionne maintenant en « multithreaded » ce qui est vu par Windows XP comme un bi-processeur.

- Reprenez la manipulation 1 et exécutez le programme à nouveau à partir de RTLauncher. Quelle différence observez-vous par rapport à l'exécution sans multithreading ? Expliquer la différence.
- Ajoutez un deuxième thread avec la priorité 31. Comment se comporte le programme lorsque vous l'exécutez avec RTLauncher ?
- Reprenez la manipulation 3 et exécutez le programme à nouveau à partir de RTLauncher (mêmes réglages de l'analyseur que lors de cette manipulation). Quelle différence observez-vous par rapport à l'exécution sans multithreading ? Pourquoi ?
- Reprenez la manipulation 4 et exécutez le programme à nouveau directement et à partir de RTLauncher (mêmes réglages de l'analyseur que lors de cette manipulation). Quelle différence observez-vous par rapport à l'exécution sans multithreading ? Pourquoi ?

Manipulation 5 :

L'objectif des manipulations 5 à 8 est de vous amener progressivement vers une solution possible pour créer un comportement temps réel prévisible sous Windows XP. A vous d'utiliser l'analyseur pour observer le comportement de votre programme. N'oubliez pas de comparer le comportement dans les priorités non temps réel et dans les priorités temps réel (avec RTLauncher). Il y a parfois des surprises.

Utilisez le projet «Manip5 » dans le dossier « C:\manipsNTTR\manip5 ». Utilisez ce projet pour créer une application qui possède un thread « Dispatcher » qui est à haute priorité. Ce thread se déclenche périodiquement et contrôle l'exécution d'autres threads périodiques.

Manipulation 6 :

Modifiez l'application réalisée dans la manipulation précédente pour que le « dispatcher » active les threads en fonction de leurs priorités.

Manipulation 7 :

Modifiez l'application réalisée dans la manipulation précédente pour que le « dispatcher » laisse du temps à XP pour s'exécuter.

Manipulation 8 :

Proposer un moyen de réaliser un « dispatcher » guidé par une politique d'ordonnancement Earliest Deadline First (EDF).