



## W6 - Windows XP temps réel avec RTX

*Programmation multitâche et temps réel critique de Windows XP*

### Objectifs

- Maîtriser le modèle multi-tâches multi-process de Windows
- Synchroniser des tâches et des processus entre eux
- Communiquer entre tâches et processus
- Accéder aux ports d'E/S et aux cartes mémoire
- Implémenter les interruptions
- Maîtriser l'ordonnancement et les IPCs déterministes de RTX
- Gérer l'allocation de mémoire dynamique en temps réel
- Intercepter les "blue screen" du kernel
- Communiquer entre RTX et WIN32 par mémoire partagée
- Debugger les applications RTX

### Matériel

- un PC par binôme avec RTX et Visual C++
- corrigés des exercices sur CDROM
- support de cours

### Pré-requis

- Connaissance du langage C obligatoire (niveau [cours L2](#))
- Connaissance de Windows CE (niveau [cours W3](#))

### Environnement du cours

- Cours théorique
  - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
  - Cours dispensé via le système de visioconférence Teams (si à distance)
  - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Au début de chaque demi-journée une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

### Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

# Plan du cours

## Architecture système de Windows XP

- Architectures et Machines Virtuelles
- Description des espaces mémoire User et Kernel
  - suppression des espaces de swap
  - APIs d'allocations des pages kernel et USER
- Mécanisme client serveur des appels systèmes
  - optimisation des appels systèmes (Native API)
- Allocation des ressources et objets système

## Programmation User et Kernel

- Couche de portabilité WIN32
- Agences kernel et leurs APIs
- Outils de développement et de debug WIN32 et kernel
  - Visual Studio
  - Platform SDK ( Software Development Kit)
  - DDK ( Driver Development Kit)
- Bugs à l'exécution
  - programmation des exceptions WIN32
  - exceptions kernel (blue screen) et interprétation

## Programmation du multitâche Windows XP

- Objets du multitâche (processus, job objects, thread kernel et user, fiber)
- Zone d'allocation des tâches (TLS)
- Mécanismes d'ordonnancement
- Algorithmes de répartition multi processeur SMP
- APIs d'affinité Multiprocesseur
- Spectre des priorités WIN32 et KERNEL
- Ordonnancement dynamiques et temps réel des tâches

## Les IPCs Windows XP

- Outils de synchronisation inter-tâches
  - mutex, sémaphores, événements, thread pooling, instructions insécables, sections critiques
- Outils de communication interprocessus
  - mémoire partagée, pipes nommés & anonymes, boîte aux lettres
  - mémoire partagée entre espaces kernel et USER

## Présentation de RTX

- Architecture du sous système RTSS
- Spécifications des APIs temps réels RTAPI
- Différences entre les APIs WIN32 et les RTAPIs
- Environnement de développement RTX
- Intégration des outils RTX sous Visual Studio
- Debug intégré et distant de RTX

## La programmation de RTX

- Génération de tâches RTX et WIN32RTX

- ordonnanceur temps réel de RTX
- spectre de priorités des tâches RTX
- processus de boot de d'arrêt de la machine
- Espaces d'adressages des tâches
- Verrouillage d'espace mémoire et allocation dynamique déterministe
- Communication par mémoire partagée
- Debug des applications RTX
  - debugger de Visual Studio
  - utilisation de Windbg 5.0
- Gestion des exceptions
  - exceptions C++
  - exceptions structurées
- Interception des CRASH systèmes
  - gestion du GSOD (Green Screen of Death)
- Gestion du temps système
- Horloges et timers temps réels
- Architecture des interruptions
- Interception et isolation des interruptions par le HAL RTX
- Temps de latences matériels et logiciels de RTX

## **Programmation avancée de RTX**

- DLLs RTX
- Architecture des drivers RTX
- Points d'entrée standard des drivers RTX
- Accès aux périphériques
  - ports d'entrées sorties
  - registres mappés en mémoire
  - énumération du bus PCI
  - translation d'adresses