**Un système d’exploitation, c’est quoi ?**

Un système d'exploitation est un ensemble de programmes qui gère le matériel d'un ordinateur.

Il offre des services pour le rendre exploitable par les applications informatiques.

Il sert d'intermédiaire entre ces logiciels et le matériel informatique

*Ressources : Processeur, Mémoire (physique et virtuelle), Périphériques, Fichiers, Réseau, La communication inter-processus en évitant les interférences entre les logiciels, Contrôle d'accès, Logiciels utilitaires, Interface utilisateur*

**UN RTOS, c’est :**

* Un système d'exploitation pour lequel le temps de réaction sur un événement est déterminé par avance et contenu dans des limites temporelles min et max.
* Une partie du système temps réel. Lui seul ne peut garantir les contraintes temporelles d’un système. Un RTOS facilite la mise en œuvre d’un système temps réel.
* Utilisé dans des systèmes embarqués.
* Une application centrale avec des contraintes temps réel et une collection des services sans de tell contraintes.

**Un RTOS, ce n’est pas :**

* Un système temps réel.
* Quelque chose qui assure la mise en œuvre d’un système temps réel.
* Nécessairement très performant ou très rapide.

*Exemples de systèmes utilisant un RTOS : Automobile : Système antiblocage (ABS), contrôle moteur. Avionique : La majorité des systèmes de navigation. Industrie : Automatismes. Consoles de jeu vidéo. Caméras numériques.*

**Faiblesses temporelles d’un RTOS :**

Un OS doit assurer que les tâches seront toutes exécutées. Il essaye de partager équitablement le temps d'exécution entre les tâches.

* Les mécanismes d'accès aux ressources partagées entraînent des incertitudes temporelles.
* La gestion des entrées/sorties crée des temps morts.
* La gestion des interruptions n'est pas toujours déterminée.
* La gestion de mémoire virtuelle et de caches mémoire conduit à des fluctuations dans le temps d'exécution des fonctions du système
* La gestion des timers pour la gestion du temps n'a pas une résolution assez fine.

**Composants d’un RTOS :**

Il doit avoir les services suivants :

* Gestion de la mémoire,
* Gestion des interruptions et des entrées/sorties physiques,
* Entrées/sorties logiques et gestion des périphériques,
* Communication,
* Gestion et ordonnancement des tâches,
* Synchronisation,
* Gestion du temps

**Classement des contraintes temporelles**

***Systèmes temps-réel à contraintes dures*** : n'acceptent aucun compromis sur la durée de traitement des données. Le non-respect d’une contrainte de temps a des conséquences graves (freinage). On a besoin de garanties

***Systèmes temps-réel à contraintes souples*** : on peut tolérer le non-respect occasionnel d’une contrainte de temps (la perte d'une image dans un film)

**Interactions avec le monde extérieur**

***Scrutation cyclique (polling) :***

Ce principe est très simple à programmer avec un petit nombre de périphériques. Le temps de réaction du système facile à déterminer. Ce temps correspond au traitement d'une boucle.

Si le système satisfait aux contraintes de temps alors le système est temps réel.

Les inconvénients, la maintenance est difficile, il faut s'assurer le respect des contraintes de temps à chaque correction au code. Un ajout de périphérique remet en cause tout le système. Il faut revoir l'intégralité du code.

***Interruptions :***

Une interruption traite la réaction à contraintes temporelles. Ensuite, on continue d’exécuter les autres traitements.

**Les Tâches**

Un système à réponse événementielle passe la plupart de son temps à attendre des événements.

Pour résoudre ce problème, pendant une attente, il faut que le processeur exécute un autre programme. Ce mécanisme s'appelle multi-tâche. Le système peut offrir plus de services mais la complexité de la gestion des ressources du système augmente.

**Une tâche est un programme élémentaire exécuté par le processeur**

Elle est constituée de trois zones :

1. Code : contient les instructions du programme,
2. Données : contient les données globales du programme,
3. Pile : contient la pile du programme, pour les données temporaires.

**Contexte d’une tâche**

Chaque tâche a un contexte qui contient le code, les données, la pile mais aussi les registres et l'état du processus et autres infos

État du processus

1. Active : la tâche possède le processeur et s'exécute,
2. Prête : la tâche ne possède pas le processeur mais on peut le lui attribuer à tout instant,
3. Bloquée : la tâche demande l'accès à une ressource partagée déjà utilisée,
4. Endormie : la tâche se met en sommeil pour une certaine durée.