

Chapitre 02:

Introduction aux systèmes temps réels

(Real time systems)

1. Définition:

Système: ensemble d' « activités » correspondant à un ou plusieurs traitements effectués en séquence ou en concurrence. Les traitements communiquent éventuellement entre eux. Le système est en interaction avec son environnement.

Système temps réel (STR): système dont le comportement dépend non seulement de l'exactitude des traitements effectués, mais également du temps où les résultats de ces traitements sont fournis, c.-à-d. qu'un retard dans la production d'un résultat est considéré comme une erreur.

Un STR dit critique s'il y a des conséquences d'une défaillance du système et concerne la sûreté de fonctionnement. Exemple: contrôle aérien.

Un STR est soumis à des contraintes temporelles, il n'est pas forcément rapide. L'échelle de temps peut varier selon le contexte.

Exemples:

- Réaction chimique (des heures),
- Chaîne de fabrication (des minutes),
- Surveillance de centrales nucléaires (secondes),
- Suivi d'une missile/Radar (microseconde)
- Robotique,
- Fourniture d'images et son pour le multimédia,
- Suivi opératoire en milieu médical.

Dans un système temps réel, un résultat de calcul mathématiquement exact mais arrivant au-delà d'une échéance prédéfinie est **un résultat faux**.

Un système temps réel est composé d'un système contrôlé (procédé) et d'un système de contrôle:

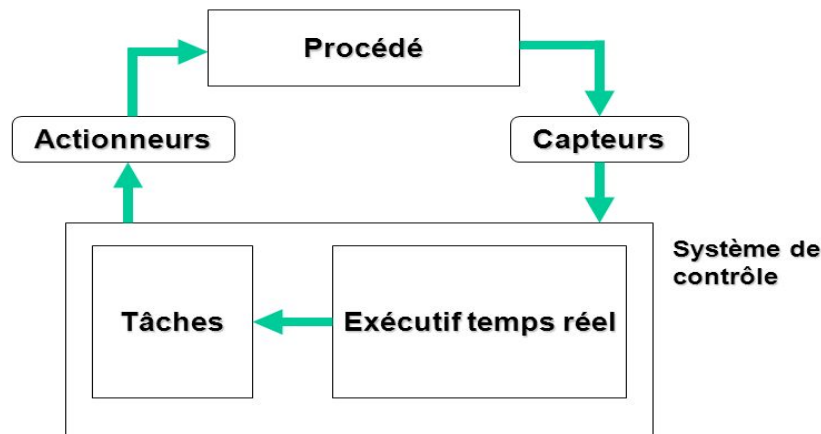


Figure 2.1. Cycle d'un système temps réel.

2. Classification des systèmes temps réels:

2.1. En termes d'interaction système/environnement :

2.1.1. Systèmes synchrones: A des moments déterminés par une référence de temps interne au système : systèmes pilotés par le temps (time driven system), programmés pour exécuter les actions/lectures à intervalles réguliers fixés par un timer (fonctionnement séquentiel).

Exemple: Feux rouge.

2.1.2. Systèmes asynchrones : A des moments déterminés par l'environnement lui-même: il attend les sollicitations et réagit à celles-ci: système piloté par les évènements (event driven system).

Exemple: Système de refroidissement d'un central électrique.

2.2. En termes de tolérance de dépassement:

2.2.1. Temps réel dur/strict (hard real time): ne tolère aucun dépassement de ces contraintes, ce qui est souvent le cas lorsque de tels dépassements peuvent conduire à des situations critiques ou des exceptions. Un système temps réel strict doit respecter des limites temporelles données même dans la pire des situations d'exécution possibles.

Exemples :

- Pilote automatique d'avion,
- Système de surveillance de centrale nucléaire.

2.2.2. Temps réel lâche/souple/mou (soft real time): s'accommode de dépassements des contraintes temporelles dans certaines limites. Un système temps réel souple doit respecter ses limites pour une moyenne de ses exécutions. On tolère un dépassement exceptionnel, qui pourra être compensé à court terme.

Exemples: - Visioconférence,
- mesure de température d'une chambre froide.

3. Caractéristiques d'un système temps réel :

3.1. Prévisibilité : Un STR doit être conçu tel que ses performances soient définies dans le pire cas.

Les performances de l'application doivent être définies dans tous les cas possibles de façon à assurer le respect des contraintes de temps. On parle de pire cas. Pour cela, il est nécessaire de connaître avec précision les paramètres des tâches : temps global de calcul de chaque activité, périodicité, date de réveil, etc.

3.2. Déterminisme : Enlever toute incertitude sur le comportement des activités :

- Dans un STR dur : on cherche à ce que toutes les échéances soient respectées.
- Dans un STR mou : minimiser le retard moyen des activités par exemple.

3.2.1. Sources de non-déterminisme:

- Charge de calculs (variations des durées d'exécution des activités),
- Entrées/sorties (temps de réaction, durée des communications),
- Interruptions (temps de réaction du système),
- Fautes et exceptions matérielles ou logicielles.

3.3. Fiabilité:

- Minimiser l'intervention humaine directe (systèmes temps réel embarqués),
- Conception tolérante aux fautes, pour garantir le comportement du système et de ses composants.

4. Limites des systèmes classiques pour le temps réel:

- politiques d'ordonnancement visent le partage équitable du temps d'exécution. Pas adaptées à des tâches plus critiques que d'autres,
- la gestion des E/S engendre de longues attentes (parfois non bornées),
- la gestion des interruptions n'est pas optimisée,

- les mécanismes de gestion de la mémoire virtuelle ne sont pas optimaux,
- les temporisateurs qui organisent le temps n'ont pas une résolution assez fine.

5. La structure d'un STR:

Un système TR est constitué de deux couches essentielles, la couche « matériel » et la couche « logiciel »

5.1. Hardware (matériels): peut avoir plusieurs configurations suivant l'importance de l'application et le cout à investir. Cependant, la structure de base est la même.

5.2. Software (logiciel): doit permettre l'exécution des tâches attribuées au calculateur, et de répondre en temps optimal à toutes les actions/lectures.

5.3. Compromis hardware/software:

- Capacités et rapidité (Mémoire, cpu, etc...)
- Algorithme (complexité)
- Gestion du hardware (ordonnancement, gestion de la mémoire, préemption).