

Systèmes temps réel

Fabien Calcado

Email: fabien.calcado@gmail.com

EFREI 2013 - 2014

1

Sommaire

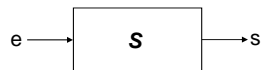
- Qu'est ce que le temps réel ?
- **Programmation multitâches**
 - Principe d'interruption
 - Gestion des interruptions sous UNIX
- **Caractéristiques des tâches dans les systèmes temps-réel**

Programmation multitâche et ordonnancement – EFREI

2

Qu'est ce que le temps-réel ?

• Système simple



- e : entrée du système S
 - S : processus
 - s : sortie du système S
- $s = F(e)$
→ non temps-réel (pas de notion de temps)

– Analyse

- Structure de données, Algorithmes

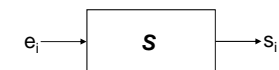
Etat 1 → Programme → Etat 2

Programmation multitâche et ordonnancement – EFREI

3

Qu'est ce que le temps-réel ?

• Système classique (*Multiple flots de données*)



- e_i : entrée du système S
 - s_i : sortie du système S
- $s_i = F(e_i)$
 $s_i = F(e_i, e_{i-1}, e_{i-2} \dots) = F(E_{i-1}, e_i)$

– Caractéristiques

- Non terminaison du système
- Cadencement événementiel
- Les traitements et les E / S se succèdent et s'entrelacent
- Le système peut *attendre* un événement

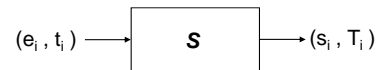
Programmation multitâche et ordonnancement – EFREI

4

Qu'est ce que le temps-réel ?



● Système temps-réel



- Les flots de données suivent des lois temporelles
- (e_i, t_i) : évènements reçus par le système S
- (s_i, T_i) : évènements émis par le système S

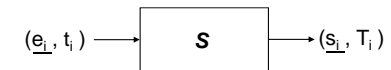
$$s_i = F(E_{i-1}, e_i, t_i)$$

$$T_i = G(E_{i-1}, e_i, t_i)$$

Qu'est ce que le temps-réel ?



● Système temps-réel numérique



Qu'est ce que le temps-réel ?

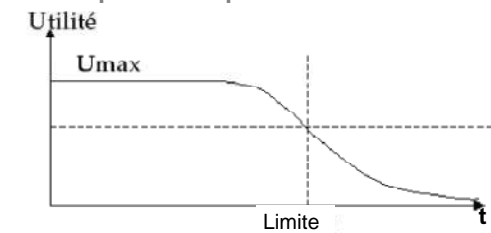
Différentes notions de « temps-réel »

- Temps-réel **strict**, dur ou critique (hard real-time system)
 - Une contrainte temporelle non respectée est considérée comme une défaillance du système
 - » Entraîne un risque de dysfonctionnement (risque d'incident)
 - » Soumis à de fortes contraintes de fiabilité
 - » Exemple : aéronautique, aérospatiale, transport ferroviaires, centrale nucléaire, gestion de salle de marchés, télé-médecine
- Temps-réel **souple**, mou (soft real-time system)
 - Dépassement des contraintes temporelles autorisées dans une certaine mesure
 - » Pas de conséquences catastrophiques (qualité de service)
 - » Exemple : téléphonie, visioconférence, jeux en réseau...

Qu'est ce que le temps-réel ?

Utilité temporelle :

- Validité de la réponse en fonction du temps
 - Utilité nulle : la réponse n'a plus d'intérêt...
- Cas du temps réel souple

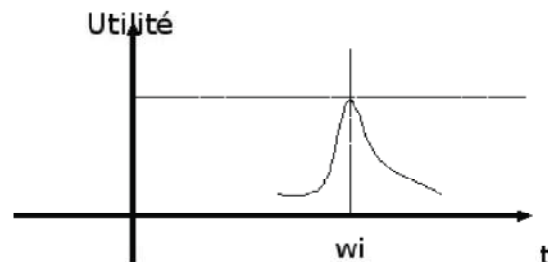


- Cas du temps réel strict ?

Qu'est ce que le temps-réel ?

Utilité temporelle dans le cas critique:

- Exactitude du résultat n'est pas suffisante, son instant de production est tout aussi important
 - w_i : instant optimal



Qu'est ce que le temps-réel ?

Caractère temps-réel d'un système

- Découle de ses spécifications
 - Les lois d'arrivées des entrées doivent être strictement définies et connues.
 - Les résultats fournis sont très souvent spécifiés de manières incomplètes:
 - » interactions sur les données communes
 - » utilités des résultats
 - Deux systèmes ayant la même mise en œuvre peuvent être considéré temps-réel ou non vis-à-vis de leurs spécifications

Qu'est ce que le temps-réel ?

■ Spécificités des applications temps-réel

- Contraintes de temps à satisfaire
 - Défaillance du système en cas de non respect
- Systèmes dédiés
 - Logiciel et matériel spécifiques
- Contraintes industrielles
 - Systèmes embarqués (poids, taille, énergie...)
 - Contraintes d'environnement

Qu'est ce que le temps-réel ?

■ Le terme temps-réel est souvent employé à tort et à travers !

- Systèmes interactifs
 - On cherche à avoir les contraintes temporelles les plus petites
- Messagerie électronique
 - Contraintes de temps peuvent être négligées
- Par abus de langage, un système fondé sur des techniques courantes en temps-réel est parfois dit « temps-réel »
 - Système multi-utilisateurs

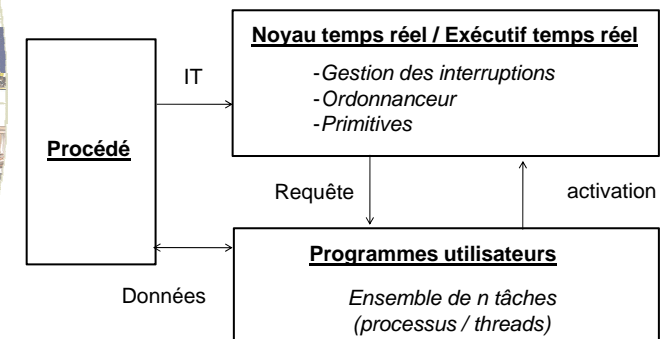
Qu'est ce que le temps-réel ?

■ Caractéristiques importantes d'un système temps-réel

- Déterministe (determinism)
 - Pour un contexte donné le comportement est toujours le même
 - » Aucune incertitude sur le comportement du système
- Prédicibilité (predictability)
 - 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 temps au pire cas (worst case execution time ou WCET)
- Fiabilité (reliability)
 - Capacité d'un système à réaliser et maintenir ses fonctionnalités dans des conditions normales d'utilisation
 - » En temps réel cela concerne le respect des contraintes de temps
 - » Dans le cas ou le système reste fiable même si certaines pannes sont apparues, on parle de tolérance aux fautes

Qu'est ce que le temps-réel ?

■ Schéma simplifié d'un système temps réel



Sommaire



- Qu'est ce que le temps réel ?
- Gestion des interruptions
- Caractéristiques des tâches dans les systèmes temps-réel

Programmation multitâche



Principe d'interruption

- Indique à l'OS l'occurrence d'un événement interne ou externe, attendu ou non
 - Interruption hiérarchisée (différent niveau de priorité)
- Les interruptions sont le seul moyen pour l'OS de reprendre la main de manière forcée sur le processeur
 - TP n°2 : gestion d'interruptions dites logiciel
- Les instructions machines s'exécutent séquentiellement
 - Saut
 - Exception (instruction invalide)
 - Trap (interruption demandée)
 - Interruption matérielle ou logicielle (événement matériel / logiciel)

Programmation multitâche



Atomicité des instructions

- Point interruptible
 - Le CPU possède un état observable (donc mémorisable) seulement à des instants précis où les valeurs des registres (registre d'état, compteur ordinal...) sont valides (stables)
- Une instruction assembleur s'exécute atomiquement
 - Une interruption ne peut survenir qu'entre deux instructions
 - » Attention, l'exécution d'une instruction se fait en plusieurs cycles
 - » Atomicité des instructions garantie entre des tâches ordonnées préemptivement

Programmation multitâche



Gestion d'interruption

- Gestionnaire
 - Programme appelé par le processeur lorsque l'interruption est rencontrée
 - Chargé de gérer les conséquences de l'occurrence de l'interruption
 - Responsable de sauvegarder le contexte d'exécution avant le traitement (routine d'interruption) puis de le rétablir
 - » On ne doit pas perturber l'exécution du programme interrompu
- **Changement de contexte ou commutation de tâche**
 - Opération provoquée par le système à chaque fois que le processeur doit être affecté à une tâche autre que celle qu'il exécutait avant ce changement

Programmation multitâche

• Etapes de la gestion des interruptions

- Arrivé d'une interruption non masquée
- Sauvegarde du contexte d'exécution de la tâche courante
- Activation du gestionnaire d'interruptions
- Traitement de l'interruption
 - En fonction de la nature de l'interruption et des priorités entre interruptions
- Reprendre la tâche interrompue (ou élire une nouvelle tâche → commutation)

Programmation multitâche

• Condition de prise en compte

- Une demande d'interruption n'est pas automatiquement traitée dès qu'elle survient
 - Interruption incidente n'est pas masquée
 - CPU est à un point interruptible
 - Pas de niveau d'interruption plus prioritaire en attente ou en cours de traitement
- Du temps de réponse à une interruption et de la constance de ce temps de réponse dépendra la qualité d'un système temps réel

Programmation multitâche

• Caractéristiques des tâches

- **Tâche non-préemptible** : tâche en cours d'exécution qui ne peut pas être interrompue
- **Tâche préemptible** : tâche en cours d'exécution qui peut être interrompue (au profit d'une autre tâche)

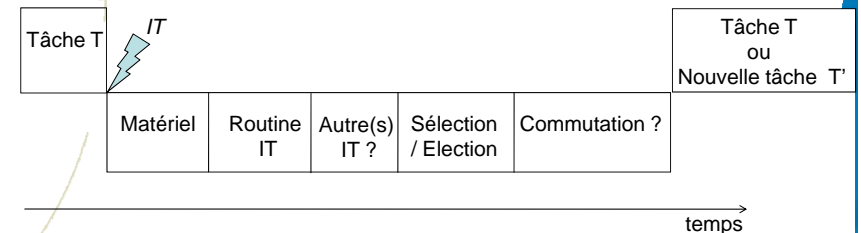
• Contraintes sur certaines ressources de calcul

- Ressources multiples ou non
- Ressources **réquisitionnables** (CPU)
 - » Signifie que la ressource est préemptible
- Ressources **non-réquisitionnables** (DSP)
 - » Différent de non-préemptible, le traitement peut être abandonné

Programmation multitâche

• Coût temporel du traitement des interruptions

- Souvent ignoré...
 - Devient problématique dans un contexte temps réel !



Programmation multitâche

● Interruptions sous UNIX

- Un « signal » désigne une interruption ou un déroutement
 - Soit un évènement externe au process/thread (tâche)
 - » frappe au clavier, signal émis par un autre thread à l'aide de la primitive Kill, signal d'horloge...
 - Soit un évènement interne au process/thread (tâche)
 - » Correspond à une erreur (erreur virgule flottante, protection mémoire...)
- Possible de masquer ces signaux
- Signal en attente d'être pris en compte
 - Bit associé dans le registre des signaux en attente est à 1
 - Si un autre signal du même type arrive il est perdu ! (voir TP)

Programmation multitâche

● Interruptions sous UNIX

- Prise en compte d'un signal = exécution d'une fonction spécifique appelé « handler »
 - Routine prédéfinie dans le système
 - » Traitement standard ou par défaut
 - Routine mise en place par l'utilisateur pour personnaliser le traitement de ce signal
 - » TP n2

Programmation multitâche

● Système multitâche

- Objectif général (système classique)
 - Optimisation de l'emploi des ressources
- Objectifs temps réel
 - Produire des résultats (justes) à des dates données
 - Gérer des échelles de temps différentes
 - » Imposée par la cohabitation de périodes différentes
- Rôle de l'OS
 - Interface avec le matériel
 - Ordonnancement des différentes tâches
 - Communication entre les tâches (au sens large)
 - Cohabitation de tâches devant s'ignorer
 - » Partage du CPU, du disque...

Programmation multitâche

● Intérêt d'une conception multitâche

- Initialement : optimiser l'utilisation du matériel
 - Plusieurs fonctionnalités sur un même ordinateur
 - Paralléliser les I / O avec le CPU
 - Optimisation des WCET des diverses tâches à accomplir (dans un cadre non temps réel : temps d'exécution moyen)
 - Profiter des multiples ressources de calcul
 - Architectures multiprocesseur
- Simplifier la conception
- Traiter des événements asynchrones
 - Interruptions à des instants pas entièrement déterminés
 - Notion d'importance ou d'échéance

Programmation multitâche

● Intérêt d'une conception multitâche

– Gérer un objectif temps-réel global

- Des échelles de temps différentes
 - » *Traitements courts passent avant les longs*
- Des degrés de criticité différents
 - » *Les traitements critiques passent en premier (si on ne peut pas faire autrement)*

– Objectifs souvent contradictoires

- Maximiser le nombre de fonctionnalités correctement acquittées
- Minimiser le coût (**dimensionnement**)

Programmation multitâche

● Autres caractéristiques des tâches

– Niveau d'urgence

- Exprime l'urgence des données produites par la tâche
- Définie par l'échéance de la tâche

– Niveau d'importance

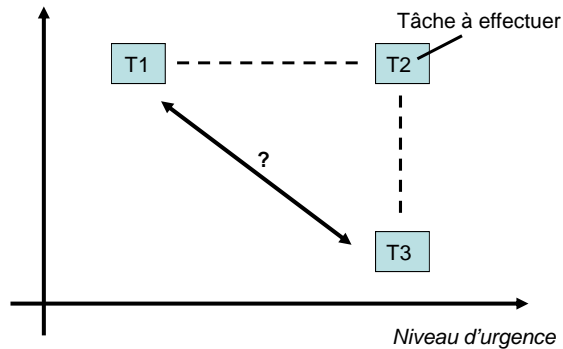
- Permet d'introduire, pour un ensemble de tâches, la capacité de résister aux fautes temporelles de certaines d'entre elles (défaillance)
- Le système doit pouvoir supprimer l'exécution de certaines tâches
 - » *Continuer à exécuter les tâches primordiales (mode dégradé)*

– Permet de distinguer deux tâches de même importance ou de même urgence

- Dédire les priorités des tâches

Programmation multitâche

Niveau d'importance



Programmation multitâche

● Classification des tâches par importance

– Critiques

- Doivent **toujours** être assurées (garantir des propriétés de **sûreté**)

– Essentielles

- Doivent être assurées autant que possible
 - » *i.e. au moins de temps en temps (garantir des propriétés de **vivacité**)*

→ **Dans tous les cas, pour un systèmes temps-réel, il faut assurer la **ponctualité** de tous les traitements**

– Sûreté

- Montrer qu'une chose ne peut pas se produire

– Vivacité

- Montrer qu'une chose se produira au bout d'un certain temps

– Ponctualité

- Montrer que les traitements se terminent à temps



• Ordonnancer correctement un système temps réel multitâche

- 100% des tâches critiques doivent respecter leurs contraintes (temporelles)
 - → preuve
- Pour les tâches essentielles
 - → faire au mieux (best effort)
- Différence entre système temps réel dur et mou
 - Dur : aucune faute temporelle acceptée
 - » *Dégâts catastrophiques*
 - Mou : une faute temporelle est acceptable
 - » *Dégâts dont le coût est estimée tolérable rapportée à sa probabilité d'occurrence*