

Notes de Cours INF 8601

Olivier Sirois

2017-10-10

Contents

1	Chapitre 1 - Introduction	2
1.1	Info	2
2	Chapitre 2 - Analyse de systèmes temps réelles.	4

Chapter 1

Chapitre 1 - Introduction

1.1 Outline

On retrouve les systèmes temps réels un peu partout. toutes les applications critiques contiennent une forme de systèmes/aspects temps réelles.

on retrouve les systèmes temps réelles dans:

- factory - usine
- drone
- AI
- medicale
- automobile
- jeux vidéos

Les systèmes ou les décisions doivent être fait dans un délai temporel sont techniquement considérés temps réelles. Cela s'applique à tous les systèmes rattachés aussi.

Les systèmes temps réeles sont généralement des systèmes qui interagisse avec des environnements externes.

Par nature, ils sont réactif. Mais normalement, on ne cherche pas seulement la bonne réponse, mais on veut l'avoir dans un temps données (spécifications). Ils contiennent aussi beaucoup de capteurs étant donnée qu'ils interagissent avec un environnement externes.

Definition : Deadline = temps ou il faut avoir une réponse (contraintes)

on peut différencier les systèmes temps réelles en deux différentes catégories :

- Hard real time systems : ils doivent répondre sans faute à leurs contraintes
- Soft real time systems : la réponse n'est pas aussi critique mais elle est quand même voulue

Définitions : Déterministe = les contraintes déterministes sont exprimées en termes de valeur fixe, i.e. aucune moyenne et autre valeur statistique

Normalement, lorsqu'on ne répond pas à une contrainte déterministe, sa équivalent à un système failure.

exemple : la porte de fermeture d'un train doit toujours marcher..

Définitions : Concurrences = trois types:

- True concurrency : plusieurs processeurs en parallèles
- pseudo concurrency : Model qui représente logiquement des activités en parallèle, sauf qu'il s'exécute généralement sur un processeur mono-cœur.
- physical concurrency : lorsqu'on parle d'un environnement qui évolue/la physique.

On peut aussi différencier l'environnement externe, dans le sens qu'on exploite normalement les patrons d'arrivées des événements. Comme par exemple un événement périodique ou aperiodique. Les systèmes temps réels ont généralement aussi un partage de ressources assez robustes.

Normalement, l'utilisation des techniques de programmation temps réels augmente considérablement la complexité de la programmation.

Définitions : Correct = le système donne un résultat qui est vrai, (qu'il calcule bien les choses)

Définition : Robuste = le système réagit bien à des conditions qui ne sont pas planifiées, même en présence de défaillances systèmes non-prévues.

Reliability = une mesure de comment souvent le système va défaillir

Fault tolerance = la reconnaissance et la gestion de fautes dans le système en les évitant avec certaines techniques

Criticalités = mesure d'importance d'une défaillance - λ crash λ division par 0

d'autres aspects assez importants sont la prédictibilité d'un système temps réel. l'interface human-machines doit être conçue de sorte qu'il prévienne l'erreur humaine et la confusion.

Évidemment, le design d'un systèmes temps réelles se réalise de la même manière qu'on logiciel traditionnelle (waterfall, spiral). On commence avec un énoncés des différents requis, un design préliminaire, on fais ensuite un design détaillé + réalisations + teest pour le logiciel et matériel avec finalement une intégration et une validation.

On travaille généralement avec plusieurs abstractions. On ne peut pas controler tout les détails de l'environnement en plus que les systèmes d'aujourd'hui sont massif.

INSÉRÉ CLASSIQUE MOORE'S LAW..

Cela fait en sorte qu'on ne peut pas suivre l'amélioration des chips. C'est pour sa qu'on se sert maintenant de languages haut niveau + conception automatique qui nous aide a suivre l'évolution technologique.

Les méthodologies aujourd'hui ont déjà beaucoup d'abstraction pris en compte. C'est méthodologies sont essentiels pour pouvoir réduire le gap de productivités.

Chapter 2

Chapitre 2 - Analyse de systèmes temps réelles.