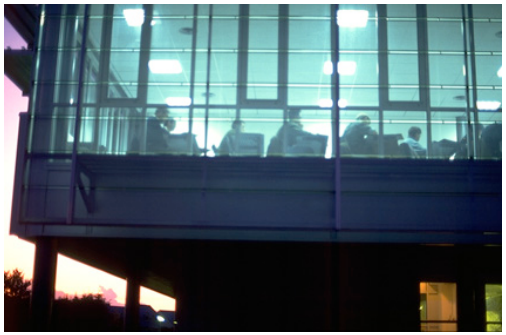


Cours A7 : Temps Réel

Pierre.Paradinas / @ / cnam.fr
Cnam/Cedric
Systèmes Enfouis et Embarqués (SEE)

Motivations

- Du jour :

  Exemples de STR

- Comprendre les mécanismes systèmes mis en oeuvre dans un système temps réel

Unix

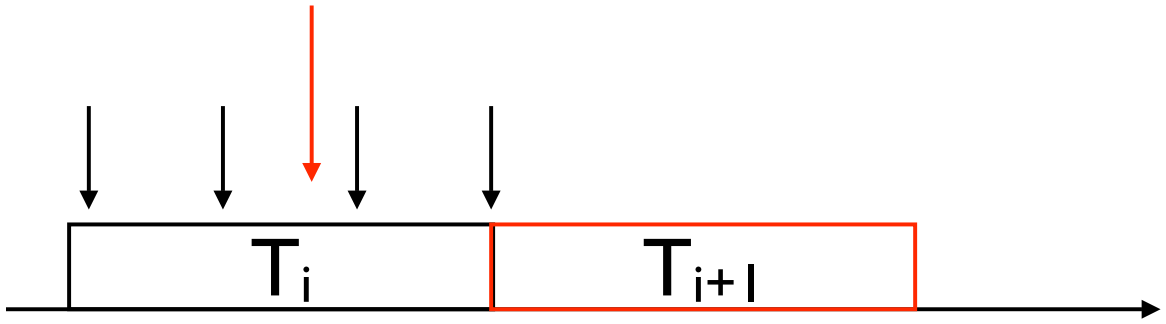
- Unix est-il un STR ?
  - Les tâches sont gérées sur le principe de “tourniquet”
    - cons : ne permet pas de respecter les échéances pour certaines tâches!
  - D’autres points font que Unix ne peut être utilisé pour faire du TR
    - aspects horloge et gestion du temps,
    - les E/S sont bloquantes,
    - lourdeur du noyau et de l’ensemble du système (difficulté liée à la taille dans des environnements contraints)

D'Unix à Unix-TR :

- Initialement :
  - "Tourniquet",
  - Introduction de priorité, 2 types de tâches :
    - Tâches gérées en tourniquet
    - Tâches gérées en priorité
    - Néanmoins, une tâche en "tourniquet" peut bloquer une tâche en "priorité" moins prioritaire,
- Préemption des tâches dans le noyau permet de remédier à ce problème

La préemption (D'Unix à Unix-TR)

- En standard Unix n'est pas préemptif,
- L'introduction de la préemption introduit un surcoût :
  - La tâche T_i arrive pendant l'exécution de la tâche T_i



La préemption (D'Unix à Unix-TR)

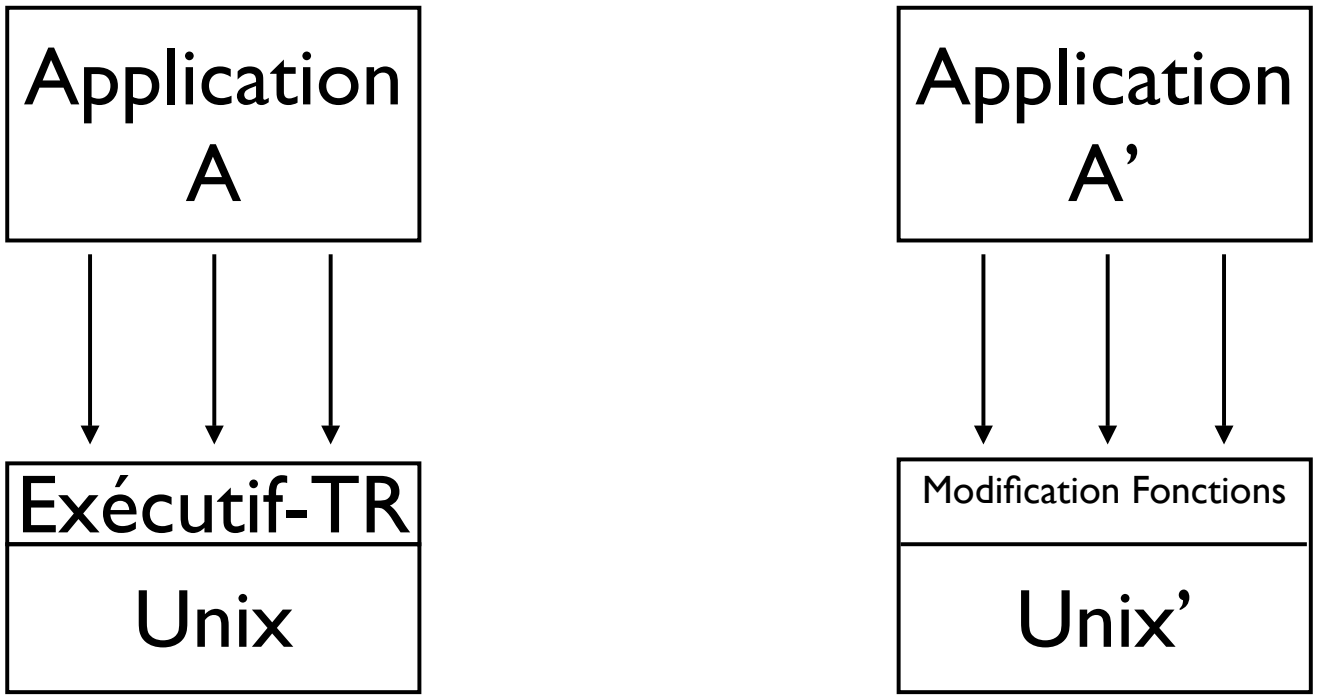
- Préemption à intervalle régulier
  - Si les points de préemption sont trop nombreux dans le noyau, alors il passe son temps dans les appels à l'ordonnanceur,
  - Si les points de préemption sont trop éloignés dans le noyau, alors des tâches ne pourront pas être traitées alors qu'elles sont prioritaires.
- Noyau complètement préemptif:
  - Pros : les interruptions interviennent au moment opportun,
  - Cons : nécessite la réécriture du noyau.

Noyau préemptif

- En rendant le noyau préemptif (une tâche est interrompue pour laisser une tâche plus prioritaire), on doit assurer :
  - L'intégrité des données (accès concurrents) : résolus par des techniques d'exclusion mutuelle
  - La réentrance des programmes de gestion E/S
- Deux solutions :
  - Addition d'un exécutif Temps Réel au dessus de Unix
  - Réécriture de fonction

Les deux solutions

- L'application A est écrite avec un environnement de développement spécifique à l'exécutif TR.
- L'application A' est écrite en tenant compte des nouvelles interfaces pour des fonctions spécifiques.



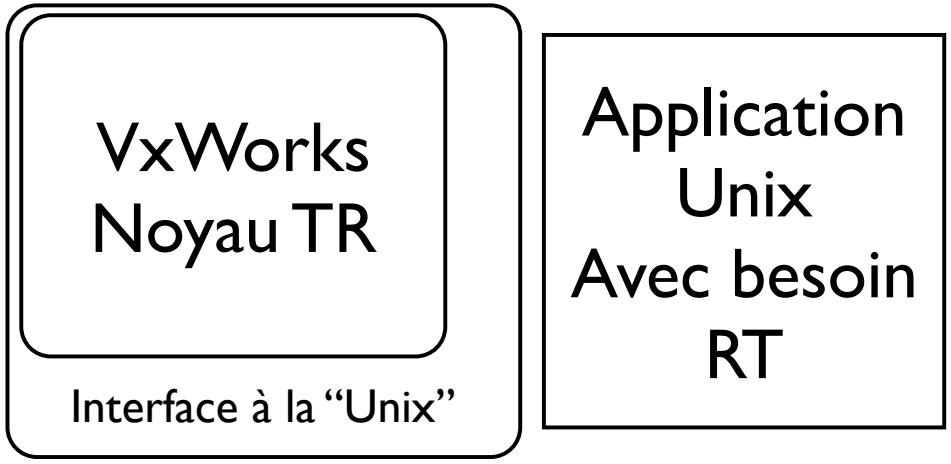
Le poids des processus Unix

- Les processus sont “lourds”, le changement de contexte à un coût important.
- Idée : introduire la notion de processus léger (“light weight process”),
- Exemple :
  - En architecture C/S, un processus léger sera le reflet d’un client, plusieurs clients seront tratés dans un même processus.

Les processus légers

- Les processus léger seront gérés avec des services du type :
  - création, activation via leur ordonnanceur, destruction,
  - partage des ressources,
  - synchronisation,
  - interruptions.
- Ils s'exécutent en général dans une zone mémoire réservée au/par le processus créateur.

Les Unix TR (les approches)



Les Unix TR (les approches)

- Unix System V : insertion de points de préemption
- Réécriture du noyau :
  - Constructeur (préemptif pas obligatoirement TR) : IBM (AIX), Sun (Solaris) {a racheté puis vendu Chorus!}, LYNX OS,
  - POSIX.1 est un standard de l'IEEE, del'ISO/IEC 9945 et de l'Open Group :
    - POSIX.1 (édition en 2004) : POSIX services de base habituellement disponible dans UNIX®
    - POSIX.1b: real-time extensions
    - POSIX.1c: thread extensions
  - Linux RT {projet "open source"}

Chorus

- D’après
  - Programming Under ChorusOS
  - Jean-Marie Rifflet
  - Novembre 2000

Les micro-noyaux

- Le noyau Unix est très gros : plusieurs millions de lignes,
- Approche de la construction initiale : unique,
- De plus en plus de fonctionnalités,
- Idée :
  - Partir d'un noyau minimal (communication entre les tâches),
  - Ajouter sur celui-ci les fonctions nécessaires en fonction des besoins (SGF, distribution,...).
- Exemple :
  - Mach (MIT (tbc)) base de l'actuel MacOSX
  - Chorus/Jaluna

Les constituants de Chorus :

- Identifiant unique
- Capacité
- Identifiant local
- Acteurs : est un groupe de ressources. Par exemple, un espace d'adresses ou ports de communication
- Threads
- Messages
- Ports, groupe de ports
- Région mémoire

Exemple de Chorus

Schéma général

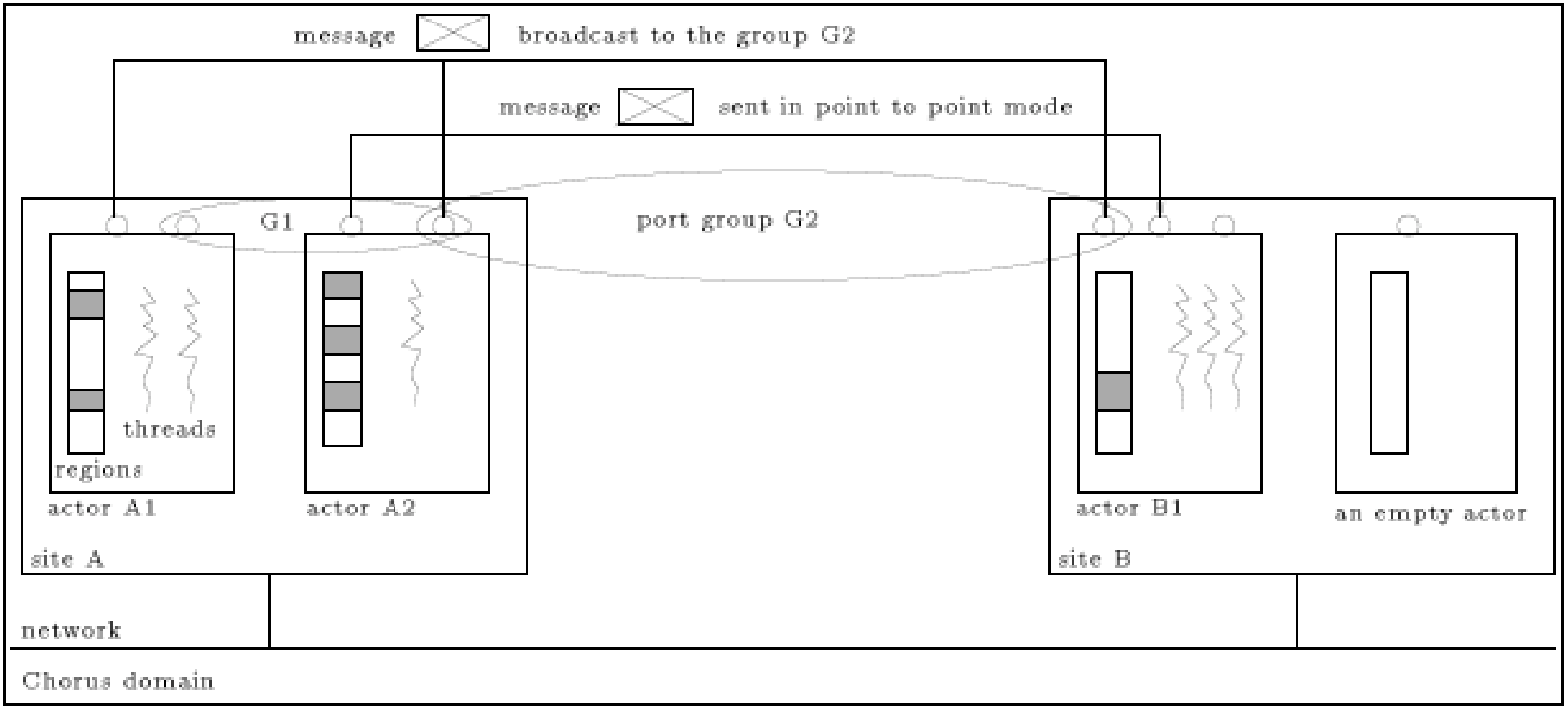
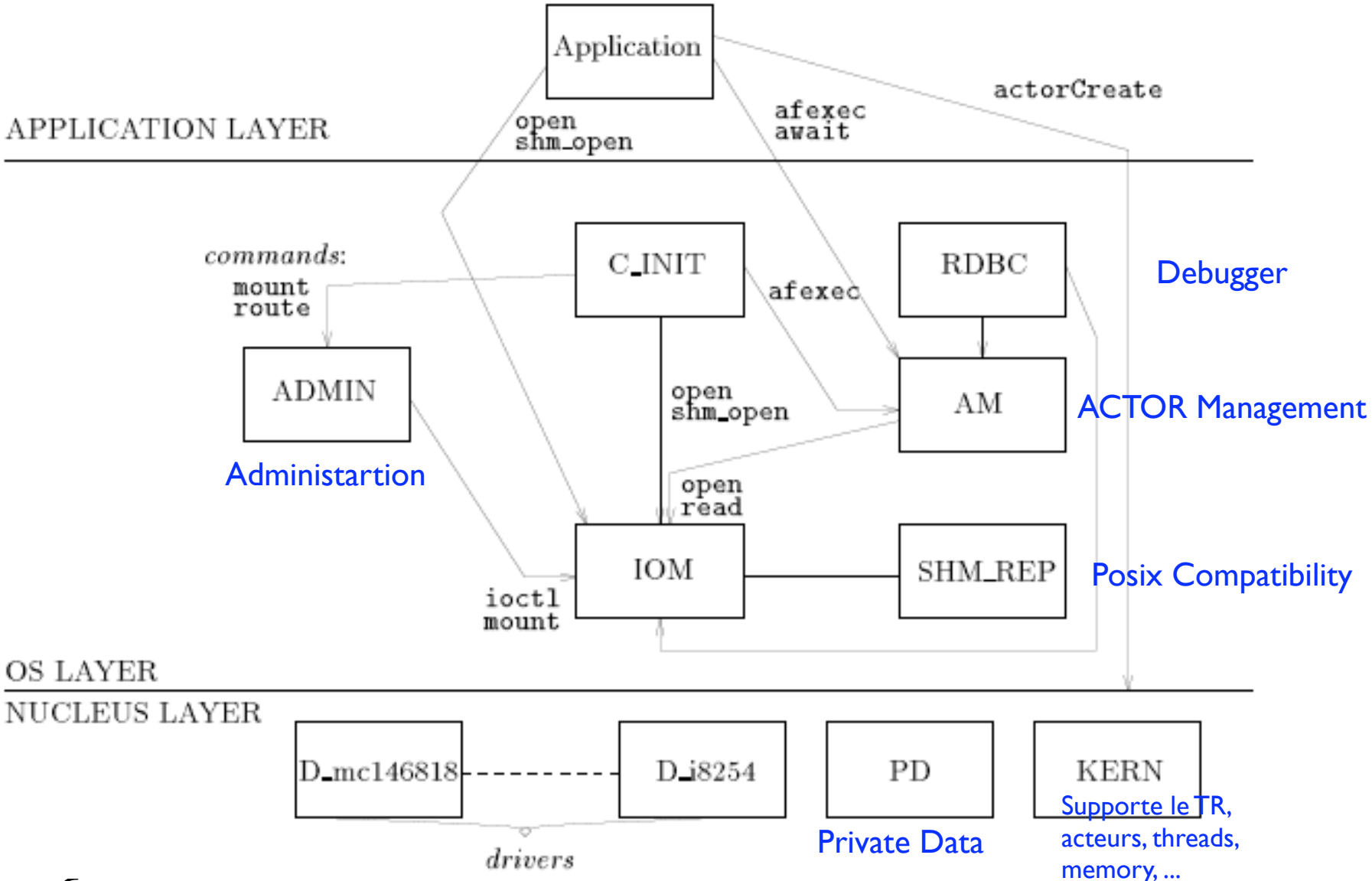


Schéma général suite



Chorus est RT

- Ordonnancement est basée sur la priorité fixe d'un composant d'une application.
  - Le threads à plus haute priorité est élu,
  - Il rend le processeur quand il a terminé ou quand un thread de plus haute priorité est prêt à s'exécuter.
- Les appels aux noyaux sont bornés en temps.
- Le temps de latence des interruptions est minimal.
- Il est possible d'attacher un gestionnaire d'interruptions à une application.