

HMIN105M : Principes de la programmation concurrente et répartie

Responsable : Hinde Bouziane (bouziane@lirmm.fr)
Intervenants : H. Bouziane et J. Cailler

UM - LIRMM

1 Introduction

Organisation

- Cours 13,5h, TD 13,5h, TP 15h
- Les supports de cours/TD-TP sont disponibles sur Moodle
- Contenu global : principes et outils pour la programmation concurrente (multi-threads et multi-processus) et la programmation d'applications distribuées
- Contrôle de connaissances : Contrôle Continu Intégral (CCI)
 - 3 évaluations
 - les détails sont sur Moodle
 - Clé de la réussite : comprendre et bien travailler les TP, être à jour.
- Pré-requis : cours système et réseaux de la licence et langage C

Questions

- Que signifie le terme concurrence ?
- A votre avis, pourquoi apprendre la programmation concurrente ?
- Donnez une définition d'une application distribuée.
- Quels sont les moyens de communication que vous connaissez dans le cadre des applications distribuées ?

Quelques définitions

- Concurrence : potentiel pour l'exécution en même temps (en parallèle) de plusieurs codes sur une même machine
 - plusieurs exécutions simultanées d'un même programme
 - exécution en parallèle de plusieurs programmes/processus
 - indépendants ou non (même application)
 - Exécution d'un programme engendrant plusieurs processus lourds (fork()) ou légers (threads)
- Programme : entité statique (fichier(s) avec un point d'entrée pour l'exécution (main))
- Processus : programme en cours d'exécution (entité dynamique)
- Remarque : une application ou un logiciel peut être composé de plusieurs programmes (exemple ?)

Contenu du cours

- Concurrency et communications intra-système
 - ① Activités dans les processus (*processus légers*, ou *threads*), communication et synchronisation entre activités
 - ② Outils de communication et de synchronisation inter-processus (IPC) : Files de messages, mémoire partagée et ensembles de sémaphores
- Programmation distribuée : communications distantes (Réseaux)
 - ① Mises en oeuvre d'applications client-serveur avec gestion simultanée de plusieurs clients et utilisation des sockets
 - Remarque : mise en oeuvre différente de celle vue en L3.
 - ② Mécanismes de communication de plus haut niveau : RPC (Remote Procedure Call)

A retenir

- Application des concepts et outils étudiés en langage C
 - ne signifie pas que ces concepts n'existent pas dans d'autres langages
 - faire la différence entre les propriétés d'un concept et leur implémentation
- Il est primordial de passer par une étape de réflexion et raisonnement avant l'étape d'implémentation
 - définir des algorithmes avant de coder
- les illustrations dans le cours sont des exemples d'utilisation des concepts présentés
 - savoir s'adapter à d'autres contextes

Bibliographie

J.M. Rifflet, J.B. Yunès *Unix, Programmation et communication*,
Dunod, 2003

Joëlle Delacroix *Linux, Programmation système et réseau*, 4th edition,
Dunod, 2012

Andrew Tanenbaum, *Réseaux*, 5^{ème} édition, Pearson Éducation, 2011
Computer Networks, 5th edition.