

PC3R

Cours 00 - Présentation

Romain Demangeon

PC3R MU1IN507 - STL S2

15/01/2025

Outils distanciels

- ▶ [Moodle](#): Sujets de TD/TME, rendus de TME. ([site principal](#))
- ▶ Cours, TDs, TMEs : [présentiel](#)

Equipe Pédagogique et Emploi du temps

- ▶ [Cours](#): (mercredi 13h00-15h45) [R. D.](#)
- ▶ [TD-TME](#) (jeudi 08h30-12h45) [R. D.](#)
- ▶ [Contact](#): romain.demangeon@sorbonne-universite.fr

- ▶ **Programmation:** langages, techniques, méthodes, outils
- ▶ **Concurrente:** mémoire partagée, *threads*,
- ▶ **Réactive:** programmation synchrone,
- ▶ **Répartie:** passage de messages, client-serveur,
- ▶ **Réculaire:** Web.

Résumé du Cours

"La programmation concurrente, c'est difficile."

Objectifs:

- ▶ **Comprendre** les difficultés engendrer par la programmation de plusieurs agents indépendants,
- ▶ **Connaître** des méthodes qui permettent de garantir la qualité du code produit,
- ▶ **Explorer** (superficiellement) différents langages et leurs modèles concurrents.
- ▶ **Expérimenter** différents styles de programmation (synchrone, web, passage-de-message, ...)

- ▶ Avant PC3R
 - ▶ Cours de **Systèmes d'Exploitation** (mémoire partagée),
 - ▶ Cours de **Réseau** ou de **Web** (client-serveur).
- ▶ Après PC3R
 - ▶ **PPC**: théorie de la concurrence et programmation synchrone.
 - ▶ **CPS**: composants concurrents.
 - ▶ **DAAR**: algorithmique web.
- ▶ Prérequis:
 - ▶ connaissance de la programmation **impérative**, **objet** et **fonctionnelle**,
- ▶ **Langages** abordés en TD/TME:
 - ▶ **impératifs**: C, Go, Rust
 - ▶ **objets**: Java,
 - ▶ **fonctionnels**: OCaml,
 - ▶ **synchrone**: Esterel, Lustre
 - ▶ **spécification**: Promela.

- ▶ Troisième année de cette version du cours.
 - ▶ calibrage des TDs et des TMEs.
- ▶ Années précédentes:
 - ▶ beaucoup de travail en distanciel (projet)
- ▶ PC3R est une évolution de PC2R.
 - ▶ moins de threads,
 - ▶ pas d'internet,
 - ▶ pas de (gros) projet,
 - ▶ des langages différents.
- ▶ PC3R intègre une partie de DAR.
 - ▶ design des applications web,
 - ▶ projet converti en rendu de TME.
- ▶ PC3R intègre une petite partie de CPS-old.
 - ▶ vérification de modèles concurrents.

Plan (prévisionnel) du Cours

1. (Concurrente I) [Préemption](#), Modèles Concurrents.
2. (Concurrente II) [Coopération](#), Appel.
3. (Répartie I) [Passage de Messages](#), Canaux.
4. (Répartie II) [Passage de Messages Avancé](#), Futures, Appels distants.
5. (Répartie III) [Vérification](#), Types de Session, Modélisation.
6. (Réticulaire I) [Design et Communication](#), Services vs. Ressources.
7. (Réticulaire II) [Serveurs Web](#), Servlets.
8. (Réticulaire III) [Clients Web](#), Javascript.
9. (Réactive) [Instants et Signaux](#), Esterel.
10. (Répartie IV) [Ouverture](#), π -calcul.

(1-2) Programmation Concurrente

- ▶ Etude du modèle **préemptif**,
 - ▶ sémantique d'**entrelacement**,
 - ▶ **écueils** de la concurrence,
 - ▶ **atomicité**, **mutex** et **algorithmes**.
- ▶ **Modèles** de concurrence,
 - ▶ threads *POSIX*,
 - ▶ threads *Java*,
 - ▶ *ARC* de *Rust*,
 - ▶ *fair threads*,
 - ▶ *Lwt*
- ▶ **TME**: exercices "classiques" (modèles) multi-langage.

(3-4-5) Programmation Répartie

- ▶ Etude du modèle **passage de messages**,
 - ▶ opération sur les **canaux**,
 - ▶ **mobilité**.
- ▶ **Utilisations** des messages,
 - ▶ **futures**,
 - ▶ **appels distants**.
- ▶ **Vérification** des programmes répartis:
 - ▶ types de **session**,
 - ▶ **model-checker**.
- ▶ **TME**: exercices "classiques" (modèles).

(6-7-8) Programmation Réticulaire

- ▶ Design des appli webs (multi-tier)
- ▶ Communication client-serveur: HTTP et surcouches.
- ▶ Programmation Serveur: *Servlets*, ORM.
- ▶ Programmation Client: *Javascript*.
- ▶ TME: application web (sur 5 semaines).

(9) Programmation Réactive

- ▶ Présentation d'Esterel:
 - ▶ sémantique *synchrone*.
 - ▶ calcul par *instant*.
 - ▶ communication par *signaux*.
- ▶ Présentation de Lustre:
 - ▶ opérations sur les *flux*,
 - ▶ *représentation*,
 - ▶ *applications*.
- ▶ TME: Robots ?

- ▶ 1ère session:
 - ▶ Examen réparti 1 (20%) (sur les cours 1-5):
 - ▶ épreuve de 2h à mi-semestre.
 - ▶ Examen réparti 2 (40%) (sur tout le programme):
 - ▶ épreuve de 2h en fin de semestre.
 - ▶ Rendus de TME 1-5 (20%):
 - ▶ à rendre chaque semaine.
 - ▶ Projet 6-10 (20%):
 - ▶ à rendre en fin de semestre.
- ▶ Examen 2ème chance (100%) (sur tout le programme) :
 - ▶ épreuve de 2h.

Evaluation (II)

► Examens:

- **annales** possibles (Moodle).
- exercices **couvrant** les différents thèmes de l'UE.
- exercices **similaires** à ceux vus en **TD**.
- **multi-langage** (impératif, objet, fonctionnel, synchrone).
 - la **syntaxe** n'est pas fortement évaluée.
- **notes** manuscrites et impressions des **transparents** de cours autorisés pendant l'examen.

► Rendus de TME:

- **deux** rendus par semaine:
 - un "rendu **de fin de séance**" à la fin des 2h sur machine,
 - un "rendu **final**" une semaine après.
- soumission par Moodle.
- travail en **binome**.
- sanction du **plagiat**.

► Projet Web:

- soumission par GitLab.
- travail en **binôme**.