

Programmation répartie

Domaine	Ingénierie et Architecture
Filière	Informatique
Orientation	Logiciel (IL)
Mode de formation	Plein temps

Informations générales

Nom:	:	Programmation répartie
Identifiant:	:	PRR
Années académiques	:	2019-2020, 2020-2021
Responsable:	:	Patrick Lachaize
Charge de travail:	:	120 heures d'études
Périodes encadrées:	:	64 (= 48 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6
Cours								32	
Laboratoire								32	

Connaissances préalables recommandées

L'étudiant doit connaître et savoir utiliser les notions suivantes : rudiments de téléinformatique et du modèle OSI, programmation concurrente faisant intervenir des processus communiquant par mémoire partagée, bases de la synchronisation de processus et pratique du langage C et d'un langage orienté objet issu du C comme C++, Java ou C#.

Les unités d'enseignements POO1 (programmation orientée objet), PCO (programmation concurrente) et TIB (téléinformatique de base) permettent d'acquérir ces connaissances. Conditions pour la programmation automatique de cette unité : L'étudiant-e doit avoir obtenu une note supérieure ou égale à la limite de compensation dans les unités : PCO. L'étudiant-e doit avoir suivi ou suivre en parallèle les unités : TIB, POO1.

Objectifs

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de : expliquer la différence entre une application répartie et des applications où interviennent des appels à des services réseaux; décrire les critères de conception des systèmes répartis; expliquer comment mettre en oeuvre des communications réalisant une classe de fiabilité donnée; expliquer le fonctionnement des algorithmes d'exclusions mutuelles de Lamport, de Ricart et Agrawala, et de Raymond; expliquer le fonctionnement et le but des algorithmes d'élection; expliquer le fonctionnement d'applications réparties faisant intervenir les paradigmes de programmation ondulatoire, de sondes et échos, de jetons, de parallélisme adaptatif, et de diffusion; choisir le paradigme le plus adapté pour réaliser un problème simple. A l'issue des travaux pratiques en laboratoire, principalement destinés à l'assimilation des connaissances et à l'acquisition d'expérience dans le développement d'applications réparties, l'étudiant-e sera en outre capable de : implémenter des processus communiquant par messages en TCP, UDP et par diffusion; implémenter des processus communiquant par appels de procédures ou de méthodes éloignées; tester et vérifier des programmes répartis; analyser des problèmes répartis.

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 32 périodes

- Introduction aux systèmes répartis : survol architectural, définition d'un système réparti, communication par messages, modèle OSI et client-serveur, adressage, communication par RPC, fiabilité et reprise à la suite d'une panne. 4
- Présentation du langage Go (Golang) adapté à la programmation concurrente et répartie. Utilisation de canaux de communications en programmation concurrente. Utilisation des primitives de communications sur TCP-IP via des sockets. 4
- Exclusion mutuelle : mise en oeuvre des algorithmes, datation des événements, algorithmes de Lamport, de Ricard et Agrawala, et de Raymond. 8
- Algorithmes d'élection : énoncé du problème et applications, élections sans panne et avec pannes des sites, élection en anneau sans jeton, élection en anneau avec pannes. 8
- Paradigmes de la programmation répartie : programmation ondulatoire, par sondes et échos, par jetons, par parallélisme adaptatif, par diffusion, par estampilles. 8

Laboratoire: 32 périodes

- Mise en oeuvre en Go d'un algorithme simple pour mettre en pratique la programmation concurrente, les canaux de communication et les sockets . 10
- Mise en oeuvre en Go d'un algorithme d'exclusion mutuelle en programmation répartie 10
- Mise en oeuvre en Go d'un algorithme d'élection en programmation répartie 6
- Mise en oeuvre en Go d'un paradigme de la programmation répartie 6

Bibliographie

Distributed Systems: Principles and Paradigms, par Andrew S.Tanenbaum et Maarten Van Steen, Prentice Hall, 2007.

Contrôle de connaissances

Cours:

l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 2 périodes.

Laboratoire:

ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

Examen:

L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 60 minutes.

Matériel autorisé:

- Information communiquée directement par l'enseignant.

Calcul de la note finale

Note finale = moyenne cours x 0.3 + moyenne laboratoire x 0.2 + moyenne examen x 0.5

Fiche validée le 18.09.2019 par Donini Pier