

Éléments pour l'examen final (Automne 2017)

Note: Les éléments en italiques/magenta ne sont pas matière à examen

- Programmation concurrente Ruby
 - * Classes Thread, Mutex, ConditionVariable
 - * Moniteur: Exclusion mutuelle => méthodes protégées par verrou
Synchronisation conditionnelle => variables de condition

- Programmation concurrente et parallèle Java
 - * Thread vs. Runnable vs. Callable vs. Lambda-expression
 - * Méthode synchronisée vs. verrou java.util.concurrent vs. variable atomique
 - * Interfaces et classes de java.util.concurrent:
Lock/ReentrantLock, Condition, CyclicBarrier/CountDownLatch, CompletableFuture, etc
 - * Pool de threads et futures: Fixed vs. Cached vs. ForkJoin
 - * Attribut volatile *et modèle de mémoire*

- Programmation parallèle OpenMP/C
 - * Régions parallèles et activation d'équipes de threads
 - * Distribution du travail entre threads (*sections*, boucles)
 - * Section critique vs. opération atomique vs. réduction parallèle
 - * Création dynamique de tâches (task/task_wait)

- Programmation concurrente Threads Posix/C
 - * Threads, verrous, variables de condition
 - * Passage d'arguments multiples, capture du contexte, retour de résultat
 - * Parallélisme récursif

- Programmation parallèle TBB/C++
 - * Approche orientée tâches plutôt que threads, vol de tâches
 - * Notion de blocked_range et approche diviser-pour-régner
 - * Parallélisme de données: parallel_for, parallel_reduce
 - * Parallélisme de contrôle: parallel_invoke, task_group/run/cancel

- *Programmation parallèle OpenCL/C pour GPU*

- Devoir, exemples, exercices et labos
 - * Variables synchronisées
 - * Polynomes
 - * Moniteurs: Tampon borné, barrière, verrou réentrant, réducteur, etc.
 - * Arbres n-aires et count (par. récursif), histogramme (par. de résultat), etc.
 - * Etc.