Mis à jour le 17/05/2024

S'inscrire

Formation Programmation Avancée Multithread C/C++20

3 jours (21 heures)

Présentation

Notre formation en C++17 et 20 vous permettra d'améliorer la qualité de votre code et de faciliter la gestion de vos projets. Vous pourrez facilement implémenter des fonctionnalités avancées de C++ pour garantir la qualité de votre code.

Dans cette formation, destinée aux développeurs, vous apprendrez à utiliser les principales fonctionnalités de C++17 et 20 telles que les modules, les concepts, les lambdas génériques, les instructions `if constexpr` et la bibliothèque standard étendue.

Vous aborderez les concepts, procédés de modélisation et design patterns des applications multithread ainsi qu'un large choix d'outil facilitant leurs conceptions et développements. Un regard approfondi sera porté sur les primitives fournies par le matériel et le système, ainsi que leurs impacts en termes de performance.

De mêmes, certaines approches issues de la programmation temps-réel, pertinentes dans les applications multithread classiques et leur optimisation, seront étudiées.

Comme toutes nos formations, celle-ci vous présentera la dernière version stable en date et les nouveautés de [C++26](#).

Pour ce qui est de C++26, nous attendons que la version soit stable pour l'utiliser durant cette formation, mais les nouveautés seront évoquées dans notre programme.

Objectifs

- Comprendre les concepts fondamentaux de la programmation en C++
- Maîtriser les dernières fonctionnalités de C++17 et 20
- Utiliser les bibliothèques standard de C++

- Maîtriser les bonnes pratiques de programmation en C++

Public visé

- Développeurs
- Architectes

Pré-requis

- Connaissance de l'environnement de développement Linux
- Connaissance des langages C et C++

Pré-requis logiciel

Un environnement de développement C++ opérationnel avec un compilateur récent.

Programme de la formation Multithread C/C++ version 20

Concepts de base de la programmation multithread

- Modèle de gestion de tâches et de mémoire
- Principe du fonctionnement matériel

Les nouveautés de C++17

- Templates and Generic Code
- Lambda
- Attributes
- Syntax cleanup
- Cleaner multi-return and flow control
 - Library additions :
 - Data types
 - Invoke stuff
- File System TS v1
- Nouveaux algorithmes implémentés
- Threading : Parallelism TS v1
- Traits : swap, is_aggregate, has_unique_object_representations

Étude des APIs POSIX/Linux

- Threads
- Primitives de synchronisations

- Mémoire partagée & sémaphore SYSV, l'approche multiprocess
- Thread local storage

Parallélisation des algorithmes

- Partitionnement des données
- Modélisation des dépendances et interactions, définition du pipeline
- Correspondance avec l'architecture
- Réduction des données
- Exemple de patterns

Architecture matérielle

- Mono/multi processeur, mono/multicœur
- « Hyperthreading »
- Gestion des caches
- Gestion de la mémoire et NUMA

Ordonnancement

- Anatomie d'une tâche
- Changement de contexte et coût en performance
- Ordonnancement
- L'ordonnanceur Linux
- Gestion des priorités
- Les queues temps-réel
- Implémentation des primitives de synchronisation.
- Inversion de priorité

Gestion de la mémoire

- Mémoire virtuelle : MMU, pages et TLB
- Les optimisations en multiprocess
- Mémoire partagée, TLS
- Le support de NUMA

Un regard sur la latence et le temps-réel

- Anatomie d'un syscall
- Latence des syscalls
- Gestion de la mémoire et des I/O en userland

Étude et pratique des outils de développement

- Debug multithread sur gdb/lldb
- Kernel
 - perf
 - lockdep
 - stat
 - autre
- La suite valgrind, intrinsic
- Outils d'analyse et de contrôle de la gestion mémoire et des caches.
- Outils d'analyse et de contrôle du scheduler

Étude et pratique des outils de développement

- Thread safe containers
- Lockfree data structure
- Design patterns

Optimisation

- Cache fighting & false sharing
- Contention autour des verrous
- Hyperthreading

Découverte des nouveautés de C++20

- Les concepts
- Les ranges
- Les modules
- Les coroutines
- La réflexion
- Le réseau
- Les graphismes 2D (à la manière de Cairo)

Fonctionnalités C++ existantes supprimées

- Suppression des trigraphes
- Radiation des registres
- Suppression de l'opérateur déconseillé ++.
- Suppression des spécifications d'exception obsolètes
- Suppression de auto_ptr

Nouveautés du futur C++23

- Suffixes littéraux pour std::size_t
- basic_string et std::basic_string_view
- La librairie stacktrace
- Le header <stdatomic.h>

- Suppression de la liste de paramètres vides inutiles () des expressions lambda

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.