

# Formation Programmation Avancée Multithread C/C++23

S'inscrire

Durée

3 jours ( 21 heures )

## Présentation

Cette formation sur la programmation parallèle en général, et l'approche multithread en particulier, vous permettra d'approfondir vos connaissances sur les architectures multicores modernes, leur support dans Linux (2.6+), les outils de développements afin de développer des applications multithread plus simples et plus performantes.

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++23](#).

## Objectifs

- Connaître les nouveautés de C++17 à C++23.
- Maîtriser les concepts de programmation avancés en contexte multithread

## Public visé

- Développeurs
- Architectes

## Pré-requis

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

Programme de la formation Multithread C/C++ version 23

## 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

## Le threading avec C++17

### 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 de 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 aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée.

## 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 multiple permet de vérifier l'acquisition correcte des compétences.

## Sanction

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