

# Formation Développement système sous Linux ou Unix

## Description de la formation linux développement :

Le Développement d'application système sous Linux ou Unix exige de la rigueur et une connaissance structurée des API. Cette formation vous permettra de maîtriser toutes les facettes de la programmation système : Gestion des processus et threads, gestion fine du système de fichiers et de la mémoire, et bien entendu l'ensemble des outils de communication inter-processus. Pour être vraiment complet ce cours présente également les éléments d'administration système essentiel à tous développeurs.

## Objectifs de la formation

À la fin de la session, vous serez apte à :

- Utiliser les outils et les méthodes de développement Linux
- Comprendre le déroulement des processus
- Gérer de la mémoire
- S'accommoder les signaux et la communication entre processus
- Apprendre à programmer avec les threads
- Synchroniser et implémenter des threads
- Communiquer sur le réseau établi



ITgate

Training

Your Gateway to Excellence

## À qui s'adresse cette formation ?

### Public :

### Prérequis :

Les participants doivent disposer d'une bonne connaissance d'Unix ou de Linux et d'une bonne maîtrise de la programmation en C.

## Contenu du cours XML

### Outils et méthodes de développement

Environnement Linux, logiciels libres, licence GPL

Outils de développement disponibles : éditeurs, compilateurs et constructeurs, profileurs et débogueurs, bibliothèques et environnements intégrés.

Création de bibliothèques statiques et dynamiques.

### Déroulement des processus

Exécution des processus Concept de processus, identifications. Création de processus et exécution de nouveaux programmes. Terminaison d'un processus. Problèmes de sécurité Ordonnancement des processus Multitâche et priorités. Configuration classique de l'ordonnancement. Utilisation avancée des ordonnancements temps-réel Posix.1b.

### Gestion de la mémoire

Principes de la mémoire virtuelle Espace d'adressage et mémoire physique, segmentation, pagination, fautes de page. Allocation de la mémoire Fonctions classiques, exploration de l'espace d'adressage, récapitulatif. Réussites et échecs des allocations, surréservation. Utilisation de la mémoire Projection, configuration des pages, indication d'usage, détection des fuites et débordements mémoire, problèmes de sécurité.

### Signaux et communication entre processus

Gestion des signaux classiques : émission et réception d'un signal. Blocages et attentes.

Signaux temps-réel : principes, émission et réception. Communication entre processus PC

---

**Capital Social:** 50000 DT **MF:** 1425253/M/A/M/000 **RC:** B91211472015

**Tél. / Fax.:** +216 73362 100 **Email:** [contact@itgate-training.com](mailto:contact@itgate-training.com) **Web:** [www.itgate-training.com](http://www.itgate-training.com)

**Adresse :** 12 Rue Abdelkadeur Daghrrir - Hammam Sousse 4011 – Tunisie

Systeme V.

Tubes et tubes nommés.

Entrées-sorties classiques et avancées

Descripteurs et flux : concepts, utilisation, paramétrage

Entrées-sorties non bloquantes et multiplexées

Entrées-sorties asynchrones.

Les threads Posix.1c

Programmation avec les threads Principe des threads Posix.1c. Exécution et terminaison des threads. Attributs d'exécution.

Données privées des threads. Synchronisation des threads Utilisation des mutex et des conditions Posix.1c. Sémaphores Posix.1b. Implémentations des threads Modèles d'implémentation des threads. Organisation de l'espace mémoire.

Ordonnement des threads.

Gestion des signaux.

Communiquer sur le réseau

Les appels système fondamentaux. Résolution de noms, de services, création de sockets.

TCP/IP et sockets STREAM : serveurs multiprocessus et multithreads, clients TCP.

UDP/IP et sockets DGRAM : communication unicast, multicast et broadcast.

Options des sockets : paramétrer et affiner les communications

Travaux Pratiques

Les nombreux exercices et études de cas progressifs seront réalisés sur un réseau de serveurs Linux, tous les programmes réalisés en TP existent sous forme de squelette que les participants complètent eux-mêmes.