

A travers l'étude approfondie des principaux éléments du développement système sous Linux, les participants seront capables d'aborder des projets d'envergure sur l'ensemble des systèmes Unix. Les points essentiels de leur interface de programmation standard (Posix - SUSv3) sont étudiés tant de manière théorique que dans le cadre de travaux pratiques détaillés.

Organisation

Durée : 4 jours (28 heures).

Pré-requis : connaissance de Linux (niveau utilisateur) et du langage C.

Conseil cursus : pour prolonger cette formation, nous vous proposons la formation « Programmation réseau sous Linux » ou notre filière de formations « Linux Industriel ».

Thèmes abordés

- **Outils et méthodes de développement** : environnements et outils de compilation et mise au point.
- **Déroulement des processus** : création, exécution, attente...
- **Programmation multi-threads** : création et déroulement, synchronisation et communications.
- **Gestion de la mémoire** : mémoire virtuelle, allocation et libération, débogage.
- **Communications entre processus** : mémoire partagée, files de messages, signaux...
- **Aperçu de programmation réseau** : client et serveur TCP/IP.

Plan détaillé

Outils et méthodes de développement

Environnement Linux

Noyaux et distributions, composants d'un système Linux, rôle de la bibliothèque C.

Outils de développement

Editeurs et environnement Eclipse, compilateurs et bibliothèques, débogage.

Travaux pratiques

Compilations et débogages ; débogage post-mortem ; statistiques d'exécution ; tests de couverture.

Déroulement des processus

Processus Unix

Concepts, identifiants, filiations.

Débuts et fins d'un processus

Création, fins normales et anormales, fin d'un processus fils.

Exécution d'un nouveau programme

Appels-système `exec()`, `system()`, `popen()`, problèmes de sécurité.

Travaux pratiques

Créations de processus ; examen de l'état Zombie ; attente de la fin d'un fils ; exécutions de programmes ; appels de sous-programmes ; exécutions parallèles.

.../...

Programmer avec les threads Posix.1c

Threads Posix.1c

Exécution, terminaison et attributs des threads.

Synchronisation

Mutex, verrous rwlock, conditions.

Travaux pratiques

Création de threads, synchronisation sur des mutex, partage de données.
Comparaison des temps de création avec les processus.

Gestion de la mémoire

Principes de la mémoire virtuelle

Espace d'adressage et mémoire physique, segmentation, pagination.

Allocation de la mémoire

Mécanisme d'allocation et de libération, fiabilité.

Projections en mémoire

Principes, projections de fichiers ou de périphérique.

Débogage

Configuration des pages, indication d'usage, détection des fuites et débordements mémoire.

Travaux pratiques

Mémoire virtuelle ; exploration ; variables dynamiques ; projection de fichier ; configuration avec malloc() ; utilisation de mmap() ; réussite des allocations et saturation de la mémoire ; débordements de chaîne ; protection.

Communications entre processus

Mécanismes IPC Posix

Principes, files de messages, segments de mémoire partagée, sémaphores.

Signaux Unix

Principes, émission, réception, gestionnaire, blocage, signaux temps-réel.

Tubes de communication

Principes, création et utilisation, duplication, tubes nommés.

Travaux pratiques

Émission et réception de signaux ; blocage et déblocage ; signaux BSD et Système V ; signaux temps-réel ; files de message ; segments de mémoire partagée ; synchronisation par un sémaphore ; tube simple ; redirection de commandes ; tubes nommés.

Aperçu de la programmation réseau

Famille TCP/IP

protocoles en couches, routage, services, résolution de noms.

Client TCP/IP

socket, connexion, lecture et écriture, fermeture.

Serveur TCP/IP

multi-processus et multi-threads.

Travaux pratiques

Résolution d'adresse et services, client et serveurs TCP/IP.

Conclusion

Discussions libres sur l'ensemble des thèmes abordés.

Travaux pratiques

Expérimentations libres suivant les demandes des participants.