

Description

Cette classe est destinée aux participants qui connaissent déjà les bases de la programmation C et veulent savoir comment tirer parti des services (appels système) offerts par un noyau Linux ou UNIX. La classe couvre également les fonctions de la bibliothèque « ANSI standard C » qui fournissent soit des fonctions couramment nécessaires et/ou une meilleure interface pour certains appels système.

Après que les participants aient terminé le cours, ils seront en mesure d'écrire du code portable qui peut faire usage de fichiers et faire des manipulations de répertoire, processus, signaux, communication interprocessus à l'intérieur d'une machine, et la communication réseau.

Objectifs

À la fin du cours, les participants pourront :

- Comprendre d'avantage les principes avancés de la programmation Unix
- Utiliser les concepts avancés de la programmation Unix

Contenu

Introduction

- Introductions des élèves et de l'instructeur
- Hypothèses sur vos antécédents
- Classe logistique, calendrier
- Textes de classe
- Conventions typographiques

Mémoire sans tampon I/O

- Présentation
- Les descripteurs de fichiers
- « open »
- « umask »

La bibliothèque standard I/O

- Présentation
- Ouverture et fermeture de fichiers
- « Buffering »
- D'autres fonctions « stdio » non-I/O
- Fonctions I/O

Diverses fonctions, partie 1

- Fonctions et valeurs temps
- Déterminer les limites du système lors de l'exécution
- Fonctions de fichiers groupe et mot de passe

Fichiers et répertoires, partie 1

- Les fonctions de statistiques
- st_mode: type de fichier et l'autorisation des bits
- « chmod » et « fchmod »
- Access: fichier accessibilité à cocher
- Propriétaire de nouveaux fichiers
- Changement de propriétaire
- Taille du fichier
- « File Times »

Fichiers et répertoires, partie 2

- Nombre « Inode »
- Renommez
- Nombre de liens
- Lier fichiers
- Délier des fichiers
- Les liens symboliques
- Fichier système

Fichiers et répertoires, partie 3

- Création de répertoires
- Suppression de répertoires
- Changer le répertoire courant
- Chemin de répertoire courant
- La lecture de répertoires

Autres sujets I/O

- Ajout atomique
- Partage de fichiers
- « Synchronous » I/O
- Mémoire mappée I / O

L'environnement d'un processus

- Arguments de ligne de commande
- Traitement des arguments de ligne de commande
- Les variables d'environnement
- Arrêt de processus
- L'allocation dynamique de la mémoire

Travailler avec les processus, partie 1

- Identificateurs de processus
- Les groupes de processus
- Sessions
- Terminal de contrôle
- « Fork »
- « Waiting for children »



Travailler avec les processus, partie 2

- La famille de fonctions « exec »
- Définir l'identité de l'utilisateur et des groupes pour les fichiers

Traitement du signal simplifié

- Concepts du signal
- Certains signaux importants
- Générer des signaux
- « Kill »
- Alarme
- Manipulation de signaux
- Exemple

Manipulation fiable des signaux

- Signaux fiables
- Terminologie de signal fiable
- Ensembles de signaux et les fonctions qui travaillent avec eux
- Manipulation de signaux
- Appels système interrompus
- Fonctions rentrants
- « Fork », « exec », et les signaux

Diverses fonctions, partie 2

- L'utilisation des ressources
- Fixer des limites des ressources
- « Gotos » non locales

Processus de « Daemon »

- Qu'est-ce qu'un « Daemon »?
- Devenir « Daemon »
- Journalisation des erreurs de « Daemon »

I/O avancé, partie 1

- Redirection des I/O
- « fcntl »
- Le verrouillage des fichiers

Communication interprocessus

- « Pipes »
- Mémoire partagée
- Accès aux lignes de commande des structures IPC

I/O avancé, partie 2

- Présentation
- Sélectionner
- Sélectionner et le « non-blocking » I/O
- Un exemple de mise en réseau

Communication réseau, partie 1

- IPv4 structure d'adresse
- L'ordre des octets
- Nom et adresse de cartographie
- Fonctions utilitaires
- Ports

Communication réseau, partie 2 (UDP)

- Une introduction à « UDP »
- Fonction d'appelle utilisé avec « UDP »
- Un exemple de serveur
- Un exemple de client

Communication réseau, partie 3 (TCP)

- Une introduction à « TCP »
- Fonction d'appelle utilisée avec « TCP »
- Un exemple de serveur
- Un exemple de client

Communication réseau, partie 4

- Considérations de programmation « UDP »
- Considérations de programmation « TCP »
- Choix d'architecture du serveur
- Obtenir un signal lorsqu'I/O est disponible
- « Socket options »

Auditoire

- Développeurs

Préalable

- Très bonne maîtrise du langage C
- Être très familier avec la manipulation des fichiers Unix
- Avoir une très bonne expertise dans l'utilisation d'Unix Éditeur