AVANCÉ | ATELIER | RÉALISABLE EN ANGLAIS

ARCHITECTURE LINUX AVEC SYSTEMD ET D-BUS

FFCNCERCERXIO26 | PRIX : 2 580 € | DURÉE : 3 JOURS

Pauses et déjeuners offerts

PRÉSENTATION

Les scripts d'initialisation System V ont progressivement laissé place à l'infrastructure systemd. Cette évolution conduit les professionnels de Linux à faire évoluer leurs pratiques, car systemd introduit de nombreuses fonctionnalités et modernise l'administration du système.
Cette formation reprend les fondamentaux de Linux pour permettre aux professionnels de retrouver rapidement leurs repères et d'exploiter pleinement systemd. Elle donne également aux débutants les bases nécessaires pour comprendre le fonctionnement interne du système.

OBJECTIFS

- Expliquer le fonctionnement de systemd
- Intégrer un nouveau daemon dans le démarrage du Système
- Déboguer des problèmes de dépendances dans systemd
- Optimiser le temps de boot grâce à systemd
- Mettre en place des dépendances complexes entre daemons
- Utiliser systemd pour démarrer des daemons à la demande
- Exploiter systemd pour la sécurisation des daemons et des systèmes

PROGRAMME

Introduction

Systemd : concepts basiques
- Écrire un simple daemon : introduction à systemd, systemd units, systemd unit files, Les services systemd
- Vie et mort d'un service : initialisation, supervision, arrêt
- Dépendance entre unités : scheduler, configuration, target units, process de boot, autorisations, dépendances et désactivation, dépendances implicites, early-boot units
- Journald : gestion des log, entrées, journald, journalctl, fonctionnalités avancées

Travaux pratiques
- Introduction à systemctl, mise en place d'un « test service » et service fonctionnel
- Phase d'initialisation, supervision et redémarrage des services
- Etude de dépendances, cible de boot
- Installation, écriture de service « early boot »
- Etudions journald

Systemd : utilisation avancée
- Organisation dynamique des unités : gestion du fichier de configuration, generators, templates, unités de type Transient
- Monitorer des évènements systemd : paths, timers, sockets, devices, exemple complet
- Debugger avec systemd : documentation, debugger dépendances, analyse des erreurs de boot, autres outils de debug
- Gestion des points de montage : hiérarchie des fichiers système, file system management units, /etc/fstab et systemd

Travaux pratiques
- Utilisation des Drop-ins, éxprimentation de generators, utilisation de templates
- Ecriture d'un simple path, création d'un Timer, écriture d'un serveur echo
- Analyse du temps de Boot, utilisation de of /etc/fstab

Sécurité avec systemd sous Linux
- Isolation et contrôle des services : namespaces, gestion des ressources et restrictions d'exécution
- Gestion des privilèges : utilisateurs, UID/GID, droits d'accès et capabilities
- Sécurisation des services : protection du système de fichiers, filtrage des appels système, gestion des credentials et paramètres réseau

Travaux pratiques
- Utilisation de « slices », des namespaces, utilisateurs et droits, systemd et capabilities, accès aux fichiers et restrictions, systemd et syscalls

D-Bus, interprocess communication
- D-Bus architecture, D-Bus connections, noms des Clients sur le bus, remote function call, signals et stetard interfaces, D-Bus et systemd, sécurité de D-Bus, outils de Debug

Travaux pratiques
- Utilisation de busctl, monitoring d'une application simple

Synthèse et conclusion

PUBLIC/PRÉREQUIS

Professionnels ayant une première expérience d'intégration ou d'architecture de système Linux désirant acquérir une maîtrise complète.
Quel que soit le domaine d'application, si votre rôle est de comprendre ou concevoir un système Linux complexe avec plusieurs applications interagissant entre elles, cette application apportera les bases nécessaires pour maîtriser cet aspect du poste.

Une bonne connaissance en intégration ou architecture de système Linux est nécessaire pour suivre cette formation.

RESPONSABLE(S)

Samuel TARDIEU
Enseignant-chercheur à Télécom Paris. Spécialisé dans le développement sur mesure de solutions matérielles/logicielles pour les systèmes embarqués et la robotique, incluant toute la chaîne d'outils de développement. Proactif dans le monde du logiciel libre, ses expertises sont : systèmes embarqués/distribués, robotique, langages, compilation et conception de compilateurs, processeurs & microcontrôleurs.

Jérémy ROSEN
CTO de la branche embarquée de Smile et responsable des experts. Il a défini des architectures basées sur systemd dans de nombreux projet. Il a également effectué de nombreuses expertises sur les subtilités des interactions de daemons dans les systèmes Linux.

MODALITÉS PÉDAGOGIQUES

Formation animée par des professionnels utilisant systemd au quotidien dans les projets et basée sur des problématiques réelles.

De nombreux exemple et illustrations, ainsi que des mises en pratique permettent d'ancrer les informations théoriques dans le réel.