

# ARCHITECTURE LINUX AVEC SYSTEMD ET D-BUS

FFCNCERCERXIO26

PRIX : 2 550 €

DURÉE : 3 JOURS

Pauses et déjeuners offerts

## PRÉSENTATION

Les System V sont remplacés par l'infrastructure systemd. Cette évolution force tous les professionnels de Linux à réapprendre leurs acquis, car systemd amène un grand nombre de nouvelles fonctionnalités et consolide le fonctionnement de Linux.

La formation réexplique les bases du système Linux et permet à un professionnel Linux de rapidement retrouver ses marques et d'exploiter toute la richesse de ces nouveaux outils.

Pour l'intégrateur débutant, elle permet de dépasser le stade utilisateur et de poser les bases de « Linux : comment ça marche ».

## 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
- Exploiter les possibilités de systemd pour la sécurisation des daemons, des systèmes et contrôler l'utilisation des ressources

## 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 d'un service, supervision de Process, stopper une unité, après que l'unité est stoppée
- La dépendance entre unités : systemd en tant que scheduler, configurer les dépendances, target units, le process de boot, autoriser une unité systemd, dépendances et désactivation, les dépendances implicites, early- boot units
- Journald : gestion des logs, gestion des log Unix, les entrées de journal journald, journalctl, fonctionnalités avancées

#### Travaux pratiques

- Introduction à systemctl, mettre en place un "test service", un service fonctionnel
- Peaufiner la phase d'initialisation, superviser et redémarrer des services
- Implémentation des dépendances, étude de dépendances, ajouter une nouvelle cible de boot, installation d'une unité, vérifier les dépendances implicites et celles par défaut, écrire un service « early boot »
- Étudions journald

#### Systemd, utilisation avancée

- Organisation dynamique des unités : gestion du fichier de configuration, generators, templates, unités de type Transient
- Monitorer des événements systemd : paths, timers, sockets, devices, exemple complet

- Debugger avec systemd : documentation systemd, debugger les unités, debugger les dépendances, analyser les erreurs de boot, les autres outils de debug
- Gestion des points de montage : hiérarchie des fichiers, file system management units, /etc/fstab et systemd

#### Travaux pratiques

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

#### Sécurité

- Gestion des ressources, namespaces, restrictions et droits d'accès, UID, GID et utilisateurs, capabilities, filesystem Protection et masking, syscall filtering, passing credentials, sécurité du réseau, autres paramètres liés à la sécurité

#### 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 systemd 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



AVANCÉ



ATELIER



RÉALISABLE EN ANGLAIS



INTERNET DES OBJETS, SYSTÈMES  
CONNECTÉS ET LEURS APPLICATIONS

## 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 projets. 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 exemples et illustrations, ainsi que des mises en pratique permettent d'ancrer les informations théoriques dans le réel.

FORMATIONS INTER-ENTREPRISES  
SYSTÈMES EMBARQUÉS