

# Collecte et analyse d'informations liées à un dysfonctionnement du kernel

Cours Pratique de 2 jours - 14h

Réf : CZL - Prix 2024 : 1 660€ HT

Le noyau Linux génère différents messages permettant d'identifier un bug lié à un dysfonctionnement. Il est nécessaire de connaître les différentes sources et localisations de ces messages pour les extraire et les exploiter.

## OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Connaître les sources d'information relatives au fonctionnement du noyau Linux

Collecter de manière exhaustive les informations liées à un dysfonctionnement du noyau

Analyser les informations ainsi recueillies

## MÉTHODES PÉDAGOGIQUES

Ce stage, très interactif, s'appuie sur de nombreux exercices et études de cas progressifs.

## LE PROGRAMME

dernière mise à jour : 09/2021

### 1) Systèmes de fichiers et debug

- Système de fichiers virtuel procs.
- Système de fichiers virtuel sysfs.
- Collecter des informations de debug avec debugfs.
- Stocker des informations de manière persistente avec pstore.

*Travaux pratiques* : Récupérer un core dump. Utiliser gdb.

### 2) Erreurs kernel et dialogue avec le noyau

- Détection de head corruption avec heap / alloc.
- cktrace.
- Warn, Kernel tainted, liste des flags.
- Oops, panic, bug.

### 3) Configurer son kernel pour améliorer le debug

- Debug info.
- Kdump / kexec.

*Travaux pratiques* : Configuration de spin lock, mutex, utilisation de locks printk.

### 4) Les outils debug de kernel

- System.map.
- Mettre en place une console série.
- Spécificités de l'utilisation d'une console série sous Xen.
- kgbd (port série).
- Crash / kdump.
- Racing / ftrace.
- Paramètres kernel utiles : panic=oops, vga=, earlyprintk=, ignore\_loglevel, initcall\_debug, log\_buf\_len.

*Travaux pratiques* : Utilisations des outils. Mettre en place une netconsole. Utiliser qemu pour debugger.

## PARTICIPANTS

Développeurs Linux/Unix.

## PRÉREQUIS

Bonnes connaissances de Linux/Unix et de la programmation C.

## COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

## MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

## 5) Analyser les informations recueillies

- Identifier des adresses mémoire avec `addr2line`.
- Gdb, le couteau suisse du débogage.
- Outil d'analyse dédié au kernel : `crash`.
- Outil d'aide à l'analyse : `printk`.
- Définir un format de message avec `pr_*`.
- Extraire le device et son driver avec `dev_*printk` versus `dev_*` ?

*Travaux pratiques : Analyse de rapports de bugs.*

# LES DATES

---

Nous contacter