LINUX

Debug Kernel

Plan de cours N° : 1217

Durée : 2 jours (14h)

PARTICIPANTS / PRE-REQUIS

Toute personne souhaitant acquérir les connaissances sur le fonctionnement du noyau Linux et les outils de debug Kernel.

Bonne maîtrise de l'environnement Linux.

OBJECTIFS PEDAGOGIQUES

Connaître les sources d'information relatives au fonctionnement du noyau Linux. Savoir collecter de manière exhaustive les informations liées à un dysfonctionnement du noyau. Savoir analyser les informations recueillies.

MOYENS PEDAGOGIQUES

Réflexion de groupe et apports théoriques du formateur
Travail d'échange avec les participants sous forme de réunion-discussion
Utilisation de cas concrets issus de l'expérience professionnelle
Validation des acquis par des exercices de synthèse
Alternance entre apports théoriques et exercices pratiques (en moyenne 30 et 70%)
Remise d'un support de cours.
Assistance post-formation d'une durée de 1 an sur le contenu de la formation via notre adresse mail dédiée formateurs@atp-formation.com

MOYENS PERMETTANT LE SUIVI DE L'EXECUTION ET DES RESULTATS

Feuille de présence signée en demi-journée,
Evaluation des acquis tout au long de la formation,
Questionnaire de satisfaction,
Attestation de stage à chaque apprenant,
Positionnement préalable oral ou écrit,
Evaluation formative tout au long de la formation,
Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles.

MOYENS TECHNIQUES EN PRESENTIEL

Accueil des stagiaires dans une salle dédiée à la formation, équipée d'ordinateurs, d'un vidéo projecteur d'un tableau blanc.

MOYENS TECHNIQUES DES CLASSES A DISTANCE

A l'aide d'un logiciel comme Teams, Zoom etc... un micro et éventuellement une caméra pour l'apprenant, suivez une formation en temps réel et entièrement à distance. Lors de la classe en ligne, les apprenants interagissent et communiquent entre eux et avec le formateur.
Les formations en distanciel sont organisées en Inter-Entreprise comme en Intra-Entreprise. L'accès à l'environnement d'apprentissage (support de cours, labs) ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.
Les participants recevront une convocation avec lien de connexion
Pour toute question avant et pendant le parcours, une assistance technique et pédagogique est à disposition par mail et par téléphone auprès de notre équipe par téléphone au 04.76.41.14.20 ou par mail à contact@atp-formation.com

ORGANISATION

Les cours ont lieu de 9h00-12h30 13h30-17h00.

PROFIL FORMATEUR

Nos formateurs sont des experts dans leurs domaines d'intervention
Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.

ACCESSIBILITE

Les personnes atteintes de handicap souhaitant suivre cette formation sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.

MISE A JOUR

01/01/2021

31, av. du Granier
38240 Meylan
Tél. 04 76 41 14 20
Fax 04 76 41 38 45

SARL au capital de 120 000€ - 390 032 050 RCS GRENOBLE
SIRET 390 032 050 00010 - APE 8559A - TVA Intracom. FR03390032050

LINUX

Debug Kernel

Plan de cours N° : 1217

Durée : 2 jours (14h)

Systèmes de fichiers et debug

- Système de fichiers virutel procfs
- Système de fichiers virutels sysfs
- Collecter des informations de debug avec debugfs
- Stocker des informations de manière persistente avec pstore

Debug user space

- Récupérer un core dump
- Utiliser gdb
- Détection de head corruption avec heap / alloc

Erreurs kernel et dialogue avec le noyau

- cktrace
- warn
- Kernel tainted - liste des flags
- oops
- panic
- bug

Configurer son kernel pour améliorer le debug

- debug info
- kdump / kexec
- Configuration de spin lock, mutex
  Utilisation de locks
- printk

Les outils de debug kernel

- System.map
- Mettre en place une console série
- Spécificités de l'utilisation d'une console série sous Xen
- Mise en place d'une netconsole
- Utiliser qemu pour debugger
- kgbd (port série)
- crash / kdump
- De l'importance de l'appareil photo
- Tracing / ftrace
- Quelques paramètres kernel utiles
  panic=oops
  vga=
  earlyprintk=
  ignore_loglevel
  initcall=debug
  log_buf_len

Analyser les informations recueillies

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

31, av. du Granier
38240 Meylan
Tél. 04 76 41 14 20
Fax 04 76 41 38 45

SARL au capital de 120 000€ - 390 032 050 RCS GRENOBLE
SIRET 390 032 050 00010 - APE 8559A - TVA Intracom. FR03390032050