

# KAÏNA-COM

## CATALOGUE DE FORMATION

### Programmation noyau Linux et Écriture de drivers

**Ce cours pratique se concentre sur la programmation interne du noyau Linux, y compris les pilotes (drivers) de périphérique**



## KLI002 – Linux Kernel and Device Drivers

---

**Référence** KLI002

---

**Niveau**

- Débutant
- Intermédiaire
- Expert

---

**Nombre De Jours** Programme de Formation :

- 40 heures (4 heures/jour)

---

**Lieu De La Formation**

- I: i-learning, Formation individuelle (Formation en ligne)
- V: v-learning, classe virtuelle
- C: c-learning, cours présentiel

### **KAÏNA-COM**

LE CARRÉ HAUSSMANN II,  
6 Allée de la Connaissance  
77127 Lieusaint - France

---

**Prérequis** Les participants doivent disposer d'une bonne expérience de Linux et d'une bonne maîtrise de la programmation en ANSI C, avec les bibliothèques standard, y compris la programmation de sockets est essentielle. Connaissance de Linux (utilisateur).  
Un niveau d'anglais business moyen est requise car la formation sera dispensé en anglais.

---

**Public** Architectes logiciels, concepteurs, développeurs et analystes ayant une expérience Linux qui ont besoin d'apprendre et de programmer dans l'environnement du noyau, y compris les pilotes de périphériques.

---

*Ce sujet continue à la page suivante*



## KLI002 – Linux Kernel and Device Drivers, Suite

---

### Objectifs

Ce cours pratique se concentre sur la programmation interne du noyau Linux, y compris les pilotes de périphérique. Les participants découvriront l'architecture du noyau Linux, la programmation dans l'environnement du noyau, les considérations d'espace, les pilotes de périphériques réseau et les mécanismes de débogage. À l'issue de ce cours, les participants pourront développer des modules de noyau Linux et des pilotes de périphériques.

Des exemples sont en C.

Les exercices du cours comprennent l'implémentation d'un pilote de périphérique de caractère fonctionnel et d'un pilote de périphérique de réseau squelettique, utilisant le noyau 3.10 (RHEL 7.X).

---

*Ce sujet continue à la page suivante*



## KLI002 – Linux Kernel and Device Drivers, Suite

**Contenu du cours**      Contenu du cours :

Table 1: KLI002 - Contenu du cours

Chapter	Description
<b>Introduction</b>	<ul style="list-style-type: none"> <li>• The Linux Kernel</li> </ul>
<b>Kernel Architecture</b>	<ul style="list-style-type: none"> <li>• Linux kernel general properties</li> <li>• System calls</li> <li>• Task Scheduler – Details and evolution</li> <li>• I/O Schedulers               <ul style="list-style-type: none"> <li>– Elevators</li> <li>– CFQ</li> <li>– No op</li> </ul> </li> <li>• Kernel Preemption</li> <li>• Threads NPTL</li> </ul>
<b>The Kernel Perspective</b>	<ul style="list-style-type: none"> <li>• Files and FileSystems</li> <li>• Devices               <ul style="list-style-type: none"> <li>– SysFS</li> </ul> </li> <li>• Processes</li> <li>• Floating Point</li> </ul>
<b>Module Programming (+Exercises)</b>	<ul style="list-style-type: none"> <li>• Implementing Kernel modules</li> <li>• Module writing guidelines</li> <li>• Kernel structures</li> <li>• Printk</li> </ul>

*Ce sujet continue à la page suivante*



## KLI002 – Linux Kernel and Device Drivers, Suite

### Contenu du cours, Suite

Chapter	Description
<b>Character Device Drivers (+Exercises)</b>	<ul style="list-style-type: none"><li>• Device numbers</li><li>• Essential kernel structures<ul style="list-style-type: none"><li>– inode</li><li>– file</li><li>– file_operations</li><li>– cdev</li></ul></li><li>• Registering a character device</li></ul>
<b>Character Device Drivers (Continued)</b>	<ul style="list-style-type: none"><li>• Device System Calls</li><li>• open, close Working with User Space memory</li><li>• Implementing read, write and ioctl</li><li>• Virtual Memory Management – overview</li><li>• mmap</li><li>• devtmpfs</li><li>• udev</li></ul>
<b>Kernel Space Considerations (+Exercises)</b>	<ul style="list-style-type: none"><li>• Timing issues and kernel timers</li><li>• Synchronicity<ul style="list-style-type: none"><li>– semaphores</li><li>– spinlocks</li><li>– wait queues</li></ul></li><li>• read and write with support of both blocking and non blocking i/o</li><li>• poll</li><li>• Handling Interrupts</li><li>• Bottom Halves</li><li>• SoftIRQs, Work Queues, TaskLets and threaded irq's</li></ul>

*Ce sujet continue à la page suivante*



## KLI002 – Linux Kernel and Device Drivers, suite

### Contenu du cours, Suite

Chapter	Description
<b>Network Device Drivers (+ Exercises)</b>	<ul style="list-style-type: none"> <li>• The Linux Protocol Stack</li> <li>• Packet flow – from the interface to the application and back</li> <li>• Socket buffer operations</li> <li>• PF_PACKET</li> <li>• Hooking with NetFilter</li> <li>• Overriding network system calls</li> </ul>
<b>Debugging mechanisms</b>	<ul style="list-style-type: none"> <li>• Kernel debugging techniques in Linux <ul style="list-style-type: none"> <li>– strace</li> <li>– standard /proc and /sys entries</li> </ul> </li> <li>• Implementing entries in /proc</li> <li>• Handling Oops and Panics</li> <li>• debugfs</li> <li>• KProbes</li> <li>• Magic SYSRQ</li> <li>• KDB</li> </ul>
<b>The End</b>	<ul style="list-style-type: none"> <li>• Summary</li> <li>• Q&amp;A</li> <li>• Evaluation</li> </ul>

