

By: ~~Roberto~~ Popa, Linux Editor

[LinuxCertified Announces the Linux Device Driver Development Course](#)

Covering interesting modules related to device drivers

LinuxCertified is a provider of Open-Source Products, Training and Services. They sell pre-installed Linux Laptops. They also offer training in Linux and other Open-Source software. Recently, LinuxCertified has announced the Linux Device Driver Development Course class to be held in South Bay, CA from July 30th to August 1st. It seems that the course would deal with some issues related to the development and testing of the Linux device drivers. Among many others, the course would show attendees how device drivers work with the Linux kernel and how to compile and load drivers, but also how to debug drivers. For more practice development, the attendees will have to develop a complete, simple driver, which shall get them used to the entire process of creating a Linux device driver. The course will be also available as an on-site class throughout North America. **Course Objectives:-** to provide an understanding of the essentials of Linux device drivers.- to give you practical experience in developing Linux device drivers- the steps necessary to add devices to a Linux system - how to determine what hardware is present on a Linux system - the purpose and functionality of device drivers - compiling and linking device drivers - trade-offs between loadable modules and drivers compiledAs it is said on [LinuxCertified's website](#), the course addresses the software engineers who are new to Linux device drivers. However those interested in attending should have in mind that experience is almost mandatory with C and they should also be able to perform basic Unix commands, and have some experience with GNU tools of gcc, gdb. Among the more interesting modules covered by the class I could count: *User Space Driver APIs, How Loadable Modules Work, Tracing and Debugging, Accessing Hardware, Handling Interrupts, Block Device Drivers or Adding a Driver to the Kernel Tree.*