KERNEL MASTERS

Online/offline Training on

Debugging Embedded Linux Kernel & Drivers

Embedded Linux Troubleshooting

By Kishore Kumar Boddu

Hardware used

- The practical exercises will be run on a Beagle Bone Black (BBB) with a Cortex ARM.
- All exercises will be applicable to any other type of board supported by Linux.
- Online practical demonstration for Linux porting on BBB. Later on you can buy and practise, support will be provided. Material will be provided with step by step procedure for lab guidance.

"Debugging Embedded Linux Kernel & Drivers" Syllabus Summary:

Embedded Linux Troubleshooting

Session 1: @ User Space

User Space	Purpose
printf	Useful for monitoring
gdb	Source level debugger
strace	strace tool that shows all the system calls issued by a user-space program
valgrind	Memory checker like segmentation fault ERRORS.
debugfs	Debug FS useful for custom debug messages.
DDT	Device Driver Test cases like gpio, uart, i2c etc.
LTP	Linux Test Project useful for Tracing of events in the kernel and also checks kernel performance.
mmap()	User directly communicating with hardware for register programming.

Session 2: @ Kernel Space

Kernel Space	Purpose
printk & dmesg	Useful for monitoring
Module test cases	Useful for particular module test
/proc and /sys	Proc file system useful for overall system information. Sys file system useful for device hierarchy.
Probe Status	Useful for to test device working condition.
KDB	KDB is In built Debugger to test whenever kernel goes panic.
KGDB	KGDB is Kernel GDB useful for source level debugging.
Crash dump	Whenever kernel crashes, it create crash dump file for crash dump analysis.
ftrace	Ftrace is the ability to see what is happening inside the kernel

Session 3: @ u-boot (Boot loader)

Boot Loader	Purpose
u-boot commands	gpio, i2c, sspi commands.
Add commands	Useful for create custom command for debugging.
Read/write Device Registers	Using md, mm commands read/write device specific registers.