

## “Weekend Workshop on Linux Kernel & Drivers Debugging” by Anil Pugalia

### Day 1

#### + **Session 1: Debugging by Printing & Querying**

- printk & dmesg
- syslogd, klogd
- Kernel Windows: proc & sys
- I/O Control

#### + **Session 2: Crash Dumping & Analysis**

- W's of Oops
- Oops Capturing & Analysis

#### + **Session 3: Kernel Hacking Options**

- Dynamic Debugging
- Stack Overflows, Deadlocks, ...
- Magic SysRq

### Day 2

#### + **Session 4: Kernel Debuggers**

- gdb Overview
- kdb Debugging
- Remote Debugging w/ kgdb

#### + **Session 5: Live Bug Fixing**

- kprobe
- jprobe & kretprobe

#### + **Session 6: Tracing**

- System Call Tracing
- Kernel Function Tracing

#### + **Wrap Up**

- Conclusion
- What Next?

**Caution: All sessions are highly interactive & filled with live bug experiences**

## Hands-On Details

- + **Debugging by Printing & Querying**
  - Examples involving printk() w/ dmesg
  - Querying various /proc windows
  - Querying various /sys windows
  
- + **Crash Dumping & Analysis**
  - Examples generating Oops
  - Capturing Oops
  - Analysing Oops to debug
  
- + **Kernel Hacking Options**
  - Dynamically Enabling/Disabling Debug Logs
  - Debugging Stack Overflow
  - Debugging Lockups & Deadlocks
  
- + **Kernel Debuggers**
  - Playing w/ gdb
  - Debugging using kdb
  - Debugging using kgdb & gdb
  
- + **Tracing**
  - stracing applications
  - Experimenting w/ Function Tracer
  
- + **Live Bug Fixing**
  - Bug fixing using kprobe / jprobe / kretprobe