"Weekend Project shop with Linux Drivers" by **Anil Pugalia**

Day 1

+ *Goal 1: Understanding the Project & its Development Environment*

- W's of this project? – Goals & Expectations
- Setting up the Project Development Environment
- First level debugging Techniques: syslog & Oops
- Fundamentals of 'Kernel C' programming: Concurrency, Delays, Work Queues

+ *Goal 2: Understanding the Block Drivers*

- Request Queue Ecosystem
- Kernel C APIs & Data Structures
- Creating a RAM Block Device
- Partitioning a Block Device

+ *Goal 3: LDDK as a USB Memory Device*

- Understanding the USB framework
- Integrating the USB Horizontal with Block Vertical
- Programming the Control Endpoint Zero
- Exchanging the Interrupt Messages

Day 2

+ *Goal 4: File System Design & Implementation*

- Virtual File System & its Role
- File System Design & Challenges
- Hardware File System & Formatting
- Kernel File System & the 5 Operation Sets

+ *Goal 5: File System in Action*

- Modifications, Enhancements, Feature Additions

+ *Wrap Up*

- Conclusion
- What Next?

Caution: All sessions are highly interactive & hands-on with hardware

Project Guidance Details

+ Provided with an individual PC
+ Feel free, if you want to bring & setup your laptop
+ Decide yourself to do the project individually, or as a team
+ Initiate yourself with design discussions
+ Indulge yourself in Template based Coding
+ Guidance available for:
    - Setting up the Development Environment
    - Kernel C APIs & Data Structures
    - Debugging Techniques
    - Project Flow & Design

Hands-On Details (Goal-wise)

+ ***Goal 1: Project Development Environment***
    - Setting up the project development environment

+ ***Goal 2: Understanding the Block Drivers***
    - Experiments with a RAM-based Block driver
    - Creating Partitions and Formating them

+ ***Goal 3: LDDK as a USB Memory Device***
    - Auto-probing & detection of a USB device
    - Control operations for memory & LEDs of LDDK
    - USB data transfer through interrupt endpoints
    - Integrating the LDDK memory into block vertical

+ ***Goal 4: File System Design & Implementation***
    - Designing your custom File System
    - Application to Format your File System
    - Coding for a hardware-less File System
    - Mounting the File System over LDDK

+ ***Goal 5: File System in Action***
    - Experiments with basic File System operations
    - Enhance the File System to Support bigger file sizes
    - Add the feature of (efficient) renaming of files