

L-5U Linux Debugging Techniques : User-mode

(c) *kaiwanTECH*. Click [here](#) to contact us.

<i>Duration: 3 days</i>	
<i>Pre-requisites</i>	
<i>Mandatory</i>	<i>Preferable</i>
Course L-0 “Linux CLI and Shell Scripting” <i>plus</i>	Experience developing Linux apps with 'C', exposure to OS concepts
Course C-1 “Programming in C” <i>plus</i>	
Course level L-1 “Linux Systems Programming”	
<i>Below is the Outline TOC (Table Of Contents) document: it presents the (approximate) Day-wise Coverage.</i>	

Part I – Introduction

Day 1

Module 1 : Introduction

Debugging ^[1]

- Origin
- Tools
- Basic Steps
- Steps to Reduce Debugging
- Actual Cases

^[1]“Debugging” from Wikipedia.org

Part II – Essential Application-Space Debugging

Module 2 : Useful Tools and Skills

- Browsing Source
 - By Hand with find & grep
 - With ctags
 - With cscope
- Coding Style
 - Vim plugins for coding style
 - Using the indent utility
 - Makefile target
- Resources
- Static Code Analysis with [sp]Lint
 - Assignments
- Userspace Runtime Tracing Tools
 - strace
 - ltrace

Creating and Applying Patches
Lab Assignments

Day 2**Module 3 : Memory**

Internals: An Introduction to Virtual Memory Concepts
A Peek at the Internals Side
View of Process Image layout as seen by the Kernel

Common Memory Issues

Memory Leaks
malloc() and fork()
Memory overcommit

Resource Limits

Manipulating resource limits with system calls
Using ulimit

Module 4 : Using the GNU Debugger gdb

Instrumentation with [f]printf
The MSG macro

gdb Commands

Compiling a Program That Is to Be Debugged with gdb
A Typical gdb Session

Useful gdb commands

Alternatives

gdb in TUI mode / ddd / Eclipse CDT

Core Dump Analysis

What it is
Generation and Nomenclature
Procs settings
Analysing a Core Dump

Lab Assignments: Debugging with gdb

Stack Analysis

x86 (IA-32) Stack Layout in detail
Looking up the stack with gdb
[Optional] ARM-32 stack layout
[Optional] x86_64 stack layout

Attempting to Analyze a Core Dump generated without symbolic information
Techniques 1 and 2

Lab Assignments: Debugging with gdb

Day 3

Module 5 : Bug Prevention Guidelines

Some Important Guidelines for Robust API usage

General Issues

Signal Issues

Sleeping correctly

“The Ten Commandments for 'C' Programmers” – Henry Spencer

“The Top 10 Ways to get *crewe* by the 'C' Programming Language”

– D Dyer

Module 6 : System Information via procs and Performance Monitoring

What is the proc filesystem ?

A Map of /proc

Shell scripts to monitor procs – system and process

Performance Monitoring

vmstat, dstat

perf

top, iotop, iftop

More tools (blog article)

Module 7 : Memory Management Debugging

Available opensource tools

Valgrind

Introduction

Valgrind Quick Start

Installation

Debugging session

Comparison

Lab Assignments

Wrap Up
