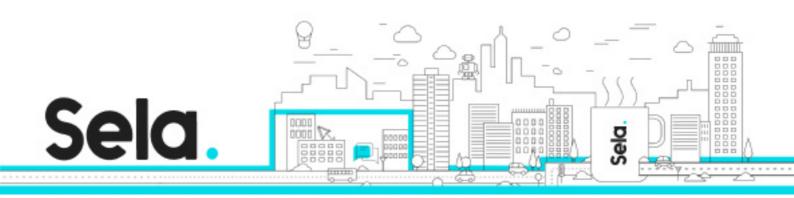LinuxDebuggWS

Linux Debugging Workshop for C and C++ Developers

Linux Debugging Workshop for C and C++ Developers

LinuxDebuggWS - Version: 1

🕐 3 days Course

Description:

What will you learn and practice
- How to effectively use gdb.
- Extending gdb to make a better developer experience.
- How to debug memory problems with gdb and valgrind.
- Debug multithreaded programs.
- Debug with Time Travel.
- Debug with GUI/TUI tools.
- Create and debug core dump files
- Remote debugging
- Using strace, ltrace and ftrace, procdump and procmon tools
- Use gdb extensions and customizations

Intended audience:

Software engineers with hands-on experience developing C or C++ applications who need to up their debugging game. Some basic knowledge of Linux, bash, make/cmake is required.

Prerequisites:

Objectives:

Topics:

gdb basics

- Preparing the executable
- Running and attaching
- Break points
- Watch points
- Catch points
- Displaying data
- Stack frames
- Printf style debugging with gdb
- Signals
- gdb and C++ exceptions

gdb UI

- TUI
- GUI front ends

Debug symbols

- Striping symbols
- Debug fission
- gdb index
- Debug symbol server

Remote debug

- gdbserver
- debug remote target

Advanced gdb

- Altering execution
- Chek points
- Reverse execution
- Trace points
- Recording and replaying

Debugging memory problems

- How to debug memory problems with gdb
- Extending gdb to display heap information
- Combining valgrind and gdb
- Valgrind vs sanitizers

Multithreaded Problems

- All stop mode
- None stop mode
- Background execution
- Debugging multithreaded code
- Analyzing with race conditions with helgrind

Customizing Gdb

- Configuring gdb with gdbinit and python
- gdb dashboard, GEF and other customizations

Complementary tools

- Bin Utils
- Linux ProcDump
- Linux ProcMon
- Other debuggers