Course Agenda

Crash Dump Analysis 2014/2015

CHARLES UNIVERSITY IN PRAGUE
faculty of mathematics and physics

Department of Distributed and Dependable Systems

ORACLE
SUSE
redhat
Motivation

- **Goal**
  - Use the right tool when an application or the kernel crashes

- **Observation**
  - Using `printf()` is usually not the right tool
More observations

- System can crash even in production
  - We cannot alter the binary and run it again
  - We have to investigate post mortem
    - Using the record of the memory layout in the time of the crash (crash dump)

- It is not wise to reinvent the wheel
  - We will see some well-proven tools and best practices
  - Some degree of low-level programming knowledge is required
Course Outline

- **Basic low-level programming**
  - Processor architectures, assembler, stack, ABI
  - IA-32, AMD64, SPARC V9

- **Basic system debugging tools**
  - Solaris, Linux, Windows
  - mdb, gdb, crash, WinDbg

- **Core files, crash dumps**
  - How to analyze them
Course Outline (2)

- **Common causes of crashes**
  - Memory corruption
  - Deadlock
  - Lockup

- **Dynamic tracing tools**
  - DTrace, SystemTap

- **System diagnostics tools**
Expected Knowledge

- **Basic C language**
  - *Programming in C++* course should be more than sufficient

- **Basic low-level programming**
  - *Principles of Computers* course should be more than sufficient

- **User-level UNIX environment**
  - *Introduction to UNIX* course should be more than sufficient

- **Basic technical English**
  - For the slides, literature, tools and manuals
Practical

- **Lectures**
  - Thursdays 12:20 – 13:50 in room **S9**

- **Tutorials / Labs**
  - Thursdays 14:00 – 15:30 in lab **SU1**
    - Physical capacity is limited to about 15 students
    - Make sure your u-lab account is working
    - You can bring your own laptops
      - Make sure you have Internet connection
Practical (2)

- **Current lecturers**
  - Martin Děcký – martin.decky@d3s.mff.cuni.cz
  - Jiří Svoboda – jiri.svoboda@oracle.com
  - Tomáš Jedlička – tomas.jedlicka@oracle.com
  - Vlastimil Babka – vbabka@suse.cz
  - Michal Hocko – mhocko@suse.cz
  - Petr Muller – muller@redhat.com

- **Past contributors**
  - Jakub Jermář (Oracle/Avast)
  - Vítězslav Bátrla (Oracle)
  - Vineeth Pillai (Oracle)
Practical (3)

- **Web**
  - Slides, practical information
  - [http://d3s.mff.cuni.cz/cda](http://d3s.mff.cuni.cz/cda)

- **Mailing list**
  - Important announcements, discussion
    - Make sure you are enlisted
  - cda@d3s.mff.cuni.cz
  - [https://d3s.mff.cuni.cz/mailman/listinfo/cda](https://d3s.mff.cuni.cz/mailman/listinfo/cda)
Grading

- **Labs credit**
  - No lab attendance required
  - But strongly recommended
  - Passing a **practical test**
    - Typical assignment: Identifying a root cause of a crash from a crash dump
    - At the end of the semester (3 tries)

- **Exam**
  - Passing a **written test**
    - Questions available on the web (3 terms)
Resources

- Lectures and labs
  - Most important hands-on experience
  - **Note**: The slides serve just as an outline

- Literature
  - **Frank Hofmann**: *The Solaris Operating System on x86 Platforms, Crashdump Analysis, Operating System Internals*
Resources (2)

- Literature (cont.)
  - Igor Ljubuncic: *Linux Kernel Crash Book*
    - Link
  - Chris Drake, Kimberley Brown: *PANIC! UNIX System Crash Dump Analysis Handbook*
    - Useful general reference
Literature (cont.)

- Richard McDougall, Jim Mauro, Brendan Gregg: *Solaris Performance and Tools: DTrace and MDB Techniques for Solaris 10 and OpenSolaris*
  - Dynamic tracing and core dump analysis using mdb
Resources (4)

References

- *Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 2*
  - IA-32 and AMD64 instruction set reference
  - Link

- *SPARC Assembly Language Reference Manual, Appendix E SPARC-V9 Instruction Set*
  - SPARC V9 instruction set reference
  - Link
Disclaimer

- Your mileage may vary
  - Different operating systems have different levels of support for crash dump analysis and observability
    - This course tries to explain the general principles
    - But sometimes we just need to demonstrate those principles in action
      - Therefore we primarily use Solaris and Fedora (on IA-32, AMD64 and SPARC V9)
      - It is up to you to translate the general principles and concrete examples to your favorite platform
      - We welcome any constructive suggestions