Welcome to LKH08

D. P. Bovet M. Cesati

System Programming Research Group
DISP, University of Rome “Tor Vergata”

Rome, 26 March 2008

The System Programming Research Group at Tor Vergata

- The System Programming Research Group at the University of Rome "Tor Vergata" is coordinated by prof. Daniel Pierre Bovet and is active in the areas of operating systems, computer networks, real-time and embedded systems, and reverse engineering.
- Some recent achievements:
  - The ASMP project
  - Porting of Linux on Apple's Quad Xeon 64-bit Xserve
  - The Nettick/CAOS project
- Forthcoming: porting of Linux on the MicroBlaze processor

The ULK project

- 1997: first course on Linux at “Tor Vergata”
- January 2001: first edition of “Understanding the Linux Kernel”, covers Linux 2.2.14, O’Reilly, 702 pages
- December 2002: second edition, covers Linux 2.4.18, 766 pages, Linux Journal’s 2003 Editors’ Choice Award
- November 2005: third edition, covers Linux 2.6.11, 928 pages
- July 2005, Ottawa Linux Symposium: stable versions are out

The LKH courses

- 2001-2002: first LKH course, emphasis on kernel debugging and simple I/O drivers
- 2003: second LKH course, emphasis on PCI I/O drivers
- 2006: third LKH course, a bit of everything, first lecture on real-time systems
- 2008: less kernel hacking, description of work performed by the SPRG

Studying Linux

- Background: good knowledge of Unix and C
- Background: good knowledge of the hardware platform
- Poor level of documentation: no comprehensive uptodate book, no successful open source documentation projects
- Make tests: while studying, perform experiments to check whether your conjectures are correct (don't bother if the kernel you tampered with crashes)
- Don't start from scratch, try first to improve an existing project

Linux is really GNU-Linux

- Linux wouldn't exist without the GNU software
- The power of the GNU C compiler, its non standard extensions, and the score of compilation options are fully exploited by the Linux code
- The flexibility of the GNU linker is crucial both to improve performances and to make programs simpler

The importance of the ELF format

- The ELF format used by the GNU linker allows programmers to create special-purpose sections
- Examples of use by Linux of special-purpose sections:
  - Fix-up code: if an API contains an invalid address, the kernel detects it on the fly and kills the process
  - Init functions: the RAM used by initialization functions can be released after the kernel has been set up
  - Alternative code: replace instructions with better alternatives for the running CPU

The role of Assembly language

- Linux uses assembly code for different purposes:
  - to address machine registers (task switching)
  - to execute privileged instructions (interrupt disabling)
  - to make Read-Modify-Write instructions atomic (lock prefix byte)
  - to code in a more efficient way (e.g. memcpy())