Advanced Tools for Software Development and Monitoring

Pavel Parízek
parizek@d3s.mff.cuni.cz
Goal of this course

- Show selected advanced tools (and features)
- Using tools to solve more complex problems
- Basic principles of their functioning (internals)
Expectations

- Basic knowledge of common platforms
  - UNIX/Linux, Windows, Java
- Advanced knowledge of main programming languages (C/C++, Java, C#)
- Experience with developing large systems
- Minimal experience with developing web applications
  - PHP, JavaScript, HTML, Servlets/JSP, ASP.NET
Plan for each lecture

- Brief introduction
- Presentations (3-4)
- Questionnaire
Presentations

• **Content**
  - General overview (purpose, features)
  - Live demo (how to use a given tool)
  - Technical details (implementation)
  - Your experience (opinion, limitations)
  - Practical exercises & small examples

• **Duration**
  - Long: 40-45 m
    - with demo and large examples
  - Short: 12-15 m
    - just basic concepts (no demo)
Questionnaire

- Feedback
  - Content (how useful it was)
  - Quality of the presentation
  - Your own comments
    - topic, tools, presentations
Challenges

- Goal: solve more complex practical task
  - in small teams (2-3 students)
  - using tools presented before

- Preliminary schedule: 12.4, 24.5

- Motivation: some fun, practical experience

- Private laptops allowed (Windows, Mac)
Grading

- Presentations: 2
  - one long (40-45 m), one short (12-15 m)

- Attendance: 60%
  - Submitted questionnaires

- Challenge: 1
Topics 1

- Software building
  - CMake, Ivy, Gradle, Bazel
  - Controlling the build process with GCC (writing linker scripts)

- Functional testing
  - Unit testing with mock objects (Mockito, Rhino Mocks, moq)
  - New libraries for unit testing (TestNG)
  - Test coverage & mutation testing
    - Jester, Jumble, NinjaTurtles, Cobertura, Clover
  - Automation: Gauge
  - Testing web applications (HtmlUnit, Selenium, Jasmine, WatiN)
  - Mock testing for web (WireMock)
  - Testing mobile applications (Espresso)

- Debugging
  - Advanced features of GDB (remote, multi-threaded, etc)
  - JPDA: Java Platform Debugger Architecture (JVM TI, JDI)
  - Firebug (web development)
Topics 2

• Runtime monitoring
  ▪ Java Management Extensions (JMX)

• Performance testing
  ▪ JMeter, Gatling, nmon, PerfCake

• Instrumentation (PIN, RoadRunner)

• Bytecode manipulation (ASM, Javassist)

• Code generation (Acceleo, AutoMapper)

• Software packaging and installation
  ▪ apt, rpm, portage, windows installers, msi files, flatpak
  ▪ Docker, Kubernetes

• Cross development (with GCC)
• Hardware emulators (QEMU)
• Virtualization: hypervisors (Xen)
Topics 3

- Continuous integration (Jenkins, TeamCity, Travis CI)
- Code review systems (Gerrit)
- Bug trackers (JIRA, Youtrack)
- Source code management (Fisheye, Perforce, Phabricator)

- Tools for dynamic programming languages
  - Short general overview (main specifics and distinct features)
  - Gem (Ruby), package (Python), NPM (for Node.js), Bundler (Ruby), Capistrano, spock (Groovy)

- Tools for developing modern applications
  - mobile: Xamarin Test Cloud, utilities for Android
  - distributed and cloud: SimianArmy, Hystrix

- Monitoring performance of distributed systems
  - Dapper, Zipkin, OpenTracing, Pinpoint, Dynatrace, AWS X-Ray
My recommendations

- **Interesting tools**
  - Apiary, Docker, Kubernetes, Selenium, Xamarin Test Cloud

- **Support for development of distributed and cloud applications**
  - SimianArmy and Hystrix ([https://github.com/Netflix/](https://github.com/Netflix/))
  - monitoring: Dapper, Zipkin, OpenTracing

- **Language-independent tools**
  - we do not want to see too many systems for Java or C#

- **Tools for debugging and testing (search for bugs)**
  - also related to security and privacy
Contact

- Web: [http://d3s.mff.cuni.cz/teaching/nswi126](http://d3s.mff.cuni.cz/teaching/nswi126)
- Email: parizek@d3s.mff.cuni.cz
- Room 202

- Office hours
  - Mon 10:30-12:00
  - Tue 10:00-12:00
  - Thu 13:30-15:00