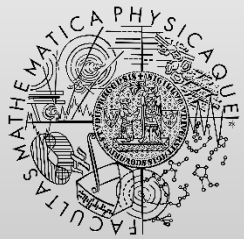
Advanced Tools for Software Development and Monitoring

http://d3s.mff.cuni.cz

Pavel Parízek

parizek@d3s.mff.cuni.cz



FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

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: 20-25 m
    - describe tool with demo and large examples
  - Short: 10-12 m
    - your own practical experience with the tool
- Alternative
  - Presentation of your own tool (extension, plugin)

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, 31.5
- Motivation: some fun, practical experience
- Private laptops allowed (Windows, Mac)

Important: changes for online teaching

- Regular seminars online via Zoom
- Supported modes for presentations
  - Live talk via Zoom in the scheduled time
    - Recorded and shared with all enrolled students
  - Create video (screencast) and publish it
    - How: upload to YouTube or share via Google Disk
    - Distributed to all students via email (just URL)
- Likely no challenges (or in a very different form)

Grading

- Presentations: **2**
  - one longer (20-25m), one shorter (10-12 m)
- Attendance: **60%**
  - Submitted questionnaires
- Challenge: **1**

Topics 1

- Software building
  - Ivy, 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)
  - Automation: Gauge
  - Testing web applications (HtmlUnit, Selenium, Jasmine, WatiN)
  - Mock testing for web (WireMock)
  - Testing mobile applications (Espresso)
  - Acceptance testing (cucumber)
- Debugging
  - Advanced features of GDB (remote, multi-threaded, etc)
  - JPDA: Java Platform Debugger Architecture (JVM TI, JDI)
  - Firebug (web development)
- Bug detectors: webhint
- Programming support
  - Tools for API design and evolution: Apiary.io, swagger, GraphQL

Topics 2

- Runtime monitoring
  - Java Management Extensions (JMX), Microsoft Azure Application Insights
  - Kamon, Prometheus, Grafana, Concurrency Visualizer, Micrometer, Sentry
- Performance testing (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
- Container platforms (Docker, Kubernetes, Titus)
- Cross development (with GCC)
- Hardware emulators (QEMU)
- Virtualization: hypervisors (Xen)
- Configuration management and deployment (Ansible)
- Infrastructure as code (HashiCorp Terraform)
- Backup tools (Duplicacy)

Topics 3

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

- 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
  - test automation: Katalon

- 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 complex distributed and cloud applications
  - SimianArmy and Hystrix (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

Choose your topic

Contact

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

Department of Distributed and Dependable Systems