

L03 Development Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
"Attribution-ShareAlike 4.0 International" license.



Scripts

- 1 Scripts
- 2 Build Tools
- 3 Development Tools
- 4 Meeting
 - Recapitulation
 - Assignments
 - Preview

Learning Outcomes

After successful completion of L03
students will be able to

- remember how development tools help
- make modifications in scripts

Scripts

FLOSS encourages to:

- combine
- modify
- share

Finding

Can be easily done using scripts.

Portable Shell Scripts

- defined by POSIX
- supported by most shells like bash, zsh, ...
- use `#!/bin/sh` as shebang

development tools:

- checkbashisms (in devscripts, `check_bashisms`)
- shfmt (`check_posix`)
- shellcheck

Reformatting (1)

Serial variant of scripts/dev/reformat-all

```
1 #!/bin/sh
2 #
3 #_@author_Klemens
4 #_@brief_Calls_all_other_reformat_scripts
5 #_@date_29.03.2019
6 #_@tags_reformat
7
8 DEV_SCRIPTS_DIR=$(dirname "$0")
9 ._ "${DEV_SCRIPTS_DIR}/include-common"
10
11 cd "$SOURCE" || exit
12
```

Reformatting (2)

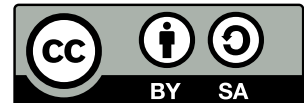
```
13 reformat() {
14 ____reformat=$1
15 ____shift
16 ____echo "starting $reformat ..."
17 ____"$reformat" "$@"
18 ____echo "finished $reformat"
19 }
20
21 IFS= '
22 '
23 for reformat in "$DEV_SCRIPTS_DIR"/reformat-*; do
24 ____["$(basename "$reformat")" = "reformat-all"] && continue
25 ____reformat "$reformat" "$@"
26 done
```

L03 Development Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“*Attribution-ShareAlike 4.0 International*” license.



Build Tools

- 1 Scripts
- 2 Build Tools**
- 3 Development Tools
- 4 Meeting
 - Recapitulation
 - Assignments
 - Preview

Goals

Build tools:

- (cross-)compile the software
- generate documentation and other files
- run scripts, shared with build server
- run tests (e.g., ctest)
- create packages (e.g., cpack)

Generation

Different ways of building software:

- e.g. automake generates `./configure` shell scripts, which generates Makefiles
- e.g. CMake generates Makefiles, Ninja or project files of various IDEs
- other build tools directly invoke the compiler

Task

Which build tool do you know?

How does it work?

Example: Elektra Plugins

README.md of plugins is used:

during configure time: include/exclude plugins based on infos/status

during build time: README.md gets included in the plugin's code

at run time: mounting plugins using infos/provides

Implementation see `scripts/cmake/Modules/LibAddPlugin.cmake`

CMake: Different Configurations

Common variants of how to run cmake are in scripts/dev/configure-*

```
1 cmake _\
2 -DPLUGINS="ALL;-EXPERIMENTAL" _\
3 -DTOOLS="ALL" _\
4 -DENABLE_DEBUG="ON" _\
5 -DENABLE_LOGGER="ON" _\
6 .
```

Environment

- 1 use Docker
- 2 `./scripts/dev/run_env` which basically does:

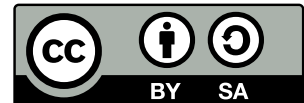
```
1 #_common_configure_script
2 export _SCRIPTS_DIR="${ELEKTRA_DIR}/scripts/dev"
3 ._ "${SCRIPTS_DIR}/include-common"
4
5 export _PATH="$BUILD/bin:${SCRIPTS_DIR}:${PATH}"
6 export _KDB_EXEC_PATH="$COMMON_PATH:${KDB_EXEC_PATH}"
7 export _LD_LIBRARY_PATH="$BUILD/lib:${LD_LIBRARY_PATH}"
8 export _MANPATH="$SOURCE/doc/man:${MANPATH}"
9 export _CLASSPATH="$CLASSPATH:$BUILD/lib/libelektra.jar"
10
11 export _PS1="[DEV]_ $PS1"
```

L03 Development Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“*Attribution-ShareAlike 4.0 International*” license.



Development Tools

- 1 Scripts
- 2 Build Tools
- 3 Development Tools**
- 4 Meeting
 - Recapitulation
 - Assignments
 - Preview

Use Tools or Libraries?

Example: Internationalization (i18n) and Localization (L10n)

- self-made key/value lookup
- GNU gettext model, plural, context

Finding

Quickly pays off to learn FLOSS tools.

Licensing Check Tools

If you copy FLOSS to Elektra's source, make sure:

- it is BSD-licensed (or less restricted)
- license information is added in `.reuse/dep5`

<https://reuse.software/>

Elektra's Test Tools

See `doc/TESTING.md`:

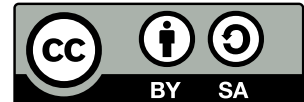
- valgrind
- ASAN
- AFL++
- Shell Recorder

L03 Development Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons
“*Attribution-ShareAlike 4.0 International*” license.



Meeting

- 1 Scripts
- 2 Build Tools
- 3 Development Tools
- 4 Meeting**
 - Recapitulation
 - Assignments
 - Preview

Learning Outcomes

After successful completion of L03
students will be able to

- remember how development tools help
- make modifications in scripts

Portable Shell Scripts

What is a shebang?

Which tools help developing portable shell scripts?

Portable Shell Scripts

- defined by POSIX
- supported by most shells like bash, zsh, ...
- use `#!/bin/sh` as shebang

development tools:

- checkbashisms (in devscripts, `check_bashisms`)
- shfmt (`check_posix`)
- shellcheck

Goals

What are the goals of build tools?

Goals

Build tools:

- (cross-)compile the software
- generate documentation and other files
- run scripts, shared with build server
- run tests (e.g., ctest)
- create packages (e.g., cpack)

Generation

Different ways of building software:

- e.g. automake generates `./configure` shell scripts, which generates Makefiles
- e.g. CMake generates Makefiles, Ninja or project files of various IDEs
- other build tools directly invoke the compiler

Task

Which build tool do you know?

How does it work?

Test Tools

- Presentation: Selenium and Cucumber
- Which other test tools do you know? Share your experiences.

Elektra's Test Tools

See doc/TESTING.md:

- valgrind
- ASAN
- AFL++
- Shell Recorder

Task

Break.

Docker

Reformatting&Test execution in Docker

Did the tutorials help you?

- A. Yes
- B. No: Didn't work
- C. No: Prefer different way
- D. Didn't try

development tools

Task

Elevator-Pitch for your favorite development tool. How did it help you?

Task

Break.

H2

Task

Problems on specific issues or PRs?

Review can also be on:

- closed H2 PRs.
- any other PRs.

Task

Add "[FLOSS H2 REVIEW]" in the beginning of the review posting.

P0

Some additional reviews were added, as there were problems in TUWEL random assignment.

For this time, however, one review is okay.

Task

Reviews finish today, next week upload corrections.

Feedback Talk

- Difficulties in P0/H2?
- ECTS breakdown realistic?
- Best/Worst Videos?



TU3 Testing

Your responsibility is that CI passes. We provide help on these topics:

- Jenkins
- Reformatting
- Tests

Task

Seize the opportunity, join the tutorial today 16:30!

M04 Continuous Integration

- Use CI
- Reproducibility
- Improve CI