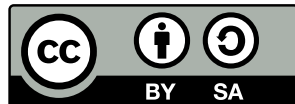


## L03 Build Tools

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 Scripts
- 3 Test Tools
- 4 Meeting
  - Recapitulation
  - Assignments
  - Preview

# Learning Outcomes

After successful completion of L03  
students will be able to

- make small modifications in (build) scripts
- remember basics of testing in FLOSS

# Portable Shell Scripts

- can use `#!/bin/sh` as shebang
- use another shell as shebang if they are not portable
- use a minimal subset of bash, zsh, ...
- can be checked with `make run_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"
12
13 reformat() {
14     reformat=$1
15     shift
```

## Reformatting (2)

```
1  echo "starting_$reformat..."
2  "$reformat" "$@"
3  echo "finished_$reformat"
4  }
5
6  IFS= '
7  '
8  for reformat in $(ls "$DEV_SCRIPTS_DIR"/reformat-*); do
9      [ "$(basename "$reformat")" = "reformat-all" ] && continue
10     reformat "$reformat" "$@"
11 done
```

## L03 Build 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 Scripts

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



# Goals

Build tools:

- (cross-)compile the software
- generate documentation or other files
- run tests or build server scripts
- create packages

# Generation

How build tools typically work:

- 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

# CMake

Elektra used automake and later switched to CMake.

## CMake

- cross-platform
- supports various programming languages
- has extensive modules for finding many tools and libraries
- see `scripts/cmake` of Elektra's repo
- `ctest` as test runner
- `cpack` for creating packages

## Different Configurations

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

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

## Example: Elektra Plugins

- only `add_plugin`<sup>1</sup> needed in `CMakeLists.txt`
- `README.md` contains instructions when to add a plugin by looking at `infos/provides` and also `infos/status`
- `cmake -DPLUGINS="ALL;-EXPERIMENTAL"` excludes plugins which have `EXPERIMENTAL` in `infos/status`
- `README.md` also gets included in the plugin's code

---

<sup>1</sup>implemented and documented in `scripts/cmake/Modules/LibAddPlugin.cmake`

## L03 Build Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

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



# Test Tools

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

# Test Runner

using ctest:

- valgrind
- ASAN
- AFL



# Environment

Run Elektra from the build folder via `scripts/dev/run_env`. 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 Build 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 Scripts
- 3 Test Tools
- 4 **Meeting**
  - Recapitulation
  - Assignments
  - Preview

# Recapitulation.



# Feedback

- Feedback Talk



# L04 Continuous Integration