L03 Build Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

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

```
#!/bin/sh
#
#␣@author␣Klemens
#␣@brief␣Calls␣all␣other␣reformat␣scripts
#␣@date␣29.03.2019
#␣@tags␣reformat

DEV_SCRIPTS_DIR=$(dirname␣"$0")
.␣"${DEV_SCRIPTS_DIR}/include-common"

cd␣"$SOURCE"

reformat()␣{
␣␣␣␣reformat=$1
␣␣␣␣shift
```

Reformatting (2)

```
␣␣␣␣echo␣"starting␣$reformat␣..."
␣␣␣␣"$reformat"␣"$@"
␣␣␣␣echo␣"finished␣$reformat"
}

IFS='
'
for␣reformat␣in␣$(ls␣"$DEV_SCRIPTS_DIR"/reformat-*);␣do
␣␣␣␣[␣"$(basename␣"$reformat")"␣=␣"reformat-all"␣]␣&&␣continue
␣␣␣␣reformat␣"$reformat"␣"$@"
done
```

L03 Build Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

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-*`

```
cmake␣\
␣␣␣␣-DPLUGINS="ALL"␣\
␣␣␣␣-DTOOLS="ALL"␣\
␣␣␣␣-DENABLE_DEBUG="OFF"␣\
␣␣␣␣-DENABLE_LOGGER="OFF"␣\
␣␣␣␣.
```

Example: Elektra Plugins

- only `add_plugin`[1] 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

[1] implemented and documented in scripts/cmake/Modules/LibAddPlugin.cmake

L03 Build Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

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:

```
# common configure script
export SCRIPTS_DIR="${ELEKTRA_DIR}/scripts/dev"
. "${SCRIPTS_DIR}/include-common"

export PATH="$BUILD/bin:$SCRIPTS_DIR:$PATH"
export KDB_EXEC_PATH="$COMMON_PATH:$KDB_EXEC_PATH"
export LD_LIBRARY_PATH="$BUILD/lib:$LD_LIBRARY_PATH"
export MANPATH="$SOURCE/doc/man:$MANPATH"
export CLASSPATH="$CLASSPATH:$BUILD/lib/libelektra.jar"

export PS1="[DEV] $PS1"
```

L03 Build Tools

Markus Raab

Institute of Information Systems Engineering, TU Wien

Meeting

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

Recapitulation.

-

Feedback

- Feedback Talk

L04 Continuous Integration