Computersystemen 2 Labo

AutoConf & Automake

Autoconf & Automake

- User gets application
- User types:
  - ./configure
  - make
  - make install

Structuur

- Structuur:
  - Src
  - Doc
  - Man
  - Scripts
  - Examples

Autoconf & Automake

- sources in src/
- documentation in doc/
- man pages in man/
- scripts in scripts/
  - Things to be installed but not compiled
- examples in examples/

Autoconf & Automake

- sources in src/
- documentation in doc/
- mar
- scri
  - Th
- exa

```
peter@PC1:~/CS2/H7-test2$ ls -R
.:
doc  examples  man  scripts  src

./doc:

./examples:

./man:
helloworld.1

./scripts:

./src:
helloworld.c
```

Autoconf & Automake

- sources in src/
- documentation in doc/
- mar
- scri
  - Th
- exa

```
peter@PC1:~/CS2/H7-test2$ ls -R
.:
doc  examples  man  scripts  src

./doc:

./examples:

./man:
helloworld.1

./scripts:

./src:
helloworld.c
```

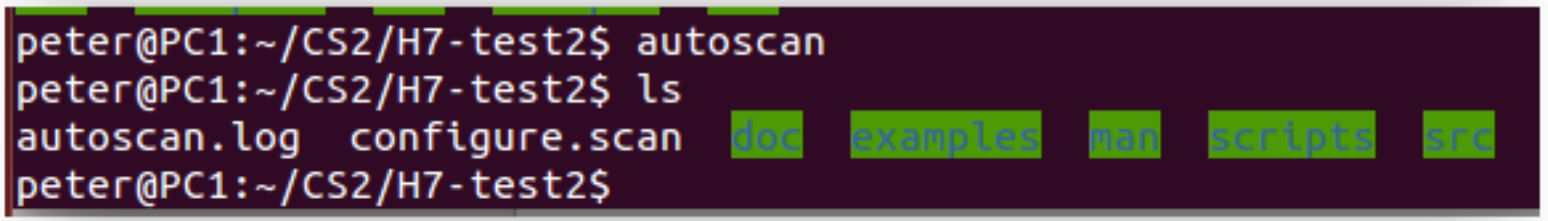
peter@PC1: ~/CS2/H7-test2/src

```
#include <stdio.h>

int main()
{
        printf("Hello world!\n");
        return 0;
}
```

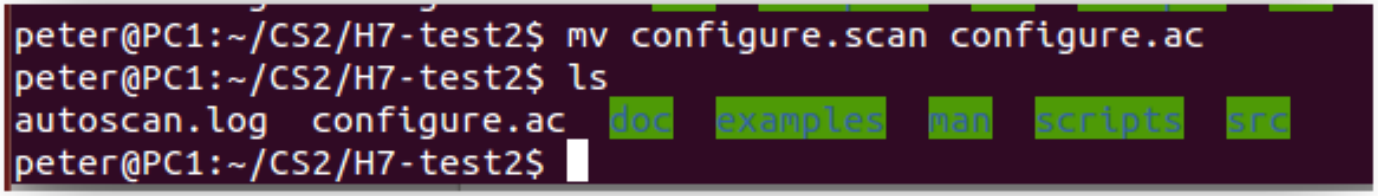
Autoconf & Automake

- Autoscan
  - autoscan tries to produce a suitable configure.ac file (autoconf's driver) by performing simple analyses on the files in the package. This is enough for the moment (many people are just happy with it as permanent). Autoscan actually produces a configure.scan file.



Autoconf & Automake

- Autoscan
  - Autoscan actually produces a configure.scan file, so we rename manually to configure.ac = file needed by autoconf.



Autoconf & Automake

- Manually adapt configure.ac
  - Change AC_INIT to reflect our program.

peter@PC1: ~/CS2/H7-test2

```
#                                               -*- Autoconf -*-
# Process this file with autoconf to produce a configure script.

AC_PREREQ([2.69])
AC_INIT([helloworld], [1.0], [test@test.com])
AC_CONFIG_SRCDIR([src/helloworld.c])
#AC_CONFIG_HEADERS([config.h])

# Checks for programs.
AC_PROG_CC

# Checks for libraries.

# Checks for header files.

# Checks for typedefs, structures, and compiler characteristics.

# Checks for library functions.

AC_OUTPUT
~
```

Autoconf & Automake

- Let autoconf produce first configure script:

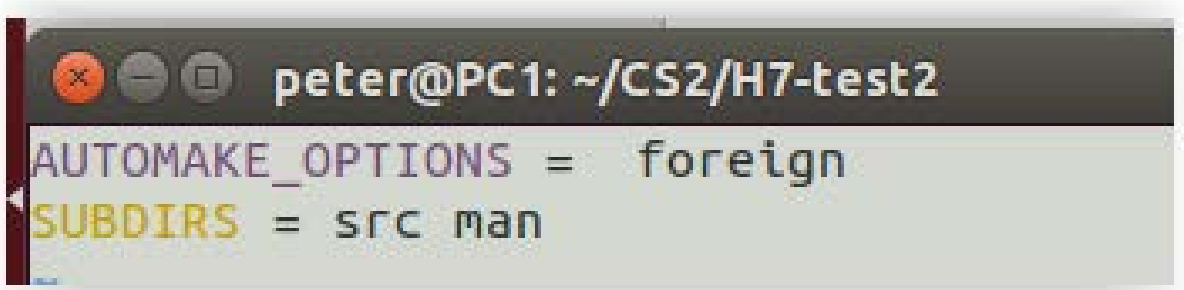
```
peter@PC1:~/CS2/H7-test2$ autoconf
peter@PC1:~/CS2/H7-test2$ ls
autom4te.cache  configure     doc       man       src
autoscan.log    configure.ac  examples  scripts
```

This produces two files: autom4te.cache and configure.

- The first one is a directory used for speeding up the job of autoshell tools, and may be removed when releasing the package.
- The latter is the shell script called by final users. In this status, what the configure script does is just checking for requirements as suggested by autoscan, so nothing very conclusive yet.

Autoconf & Automake

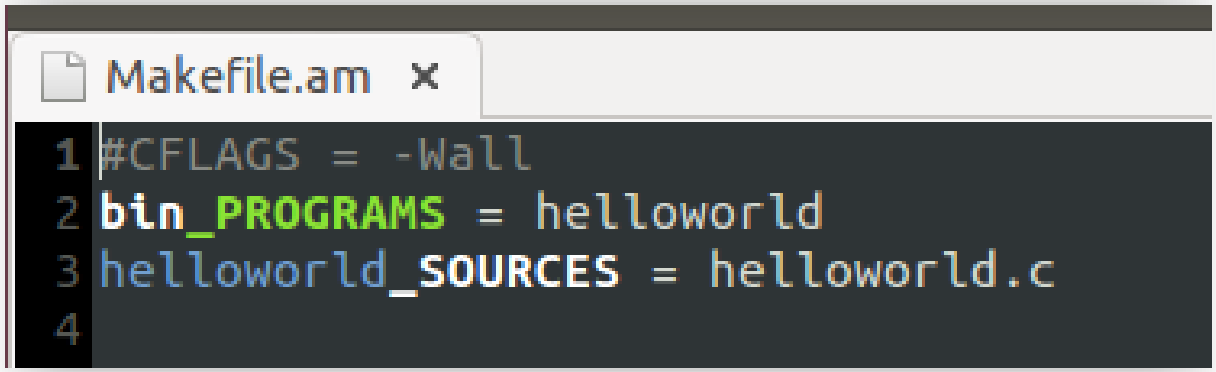
- Generate Makefile:
  - Vim Makefile.am



the first line sets the mode automake will behave like. "foreign" means not GNU, and is common for avoiding boring messages about files organized differently from what gnu expects.
The second line shows a list of subdirectories to descend for further work. The first one has stuff to compile, while the rest just needs installing, but we don't care in this file. We now prepare the Makefile.am file for each of these directories. Automake will step into each of them and produce the corresponding Makefile.in file. Those .in files will be used by autoconf scripts to produce the final Makefiles.

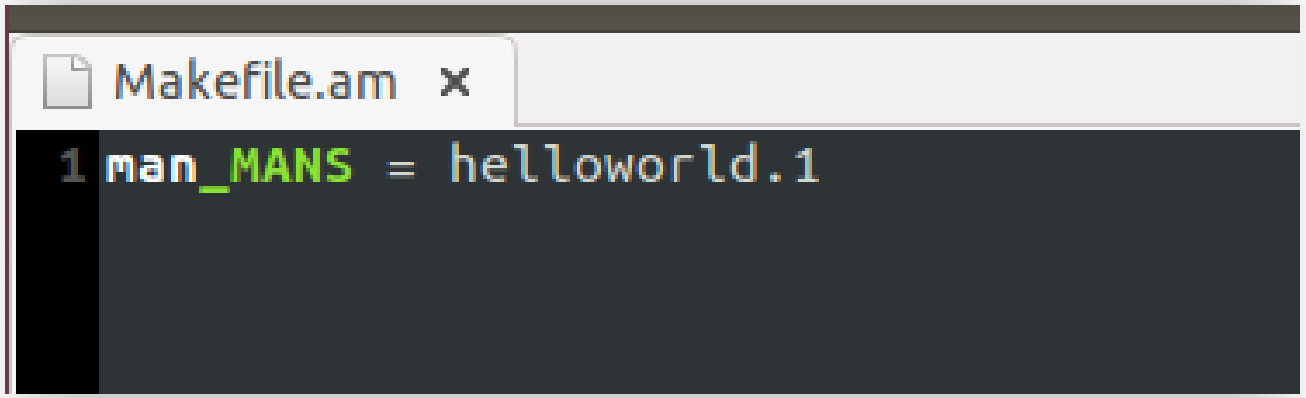
Autoconf & Automake

- Generate Makefile:
  - Vim src/Makefile.am



Autoconf & Automake

- Generate Makefile:
  - Vim man/Makefile.am



Autoconf & Automake

- Generate Makefile:
  - Other makefiles can be generated...
    - Scripts
    - Doc
    - Examples

Autoconf & Automake

- Inform autoconf about the Makefile files = edit configure.ac
- AM_INIT_AUTOMAKE
  - To initialize automake
- AC_OUTPUT
  - To output makefiles for all the directories above

Autoconf & Automake

- I
- A
  - –
- A
  - –

```
#                                               -*- Autoconf -*-
# Process this file with autoconf to produce a configure script.

AC_PREREQ([2.69])
AC_INIT([helloworld], [1.0], [test@test.com])
AM_INIT_AUTOMAKE(helloworld,1.0)
AC_OUTPUT(Makefile src/Makefile man/Makefile)
AC_CONFIG_SRCDIR([src/helloworld.c])
#AC_CONFIG_HEADERS([config.h])

# Checks for programs.
AC_PROG_CC

# Checks for libraries.

# Checks for header files.

# Checks for typedefs, structures, and compiler characteristics.

# Checks for library functions.

AC_OUTPUT
~
```

Autoconf & Automake

- Create the macros for automake.
- These will be in aclocal.m4
- Command to execute = aclocal
- Note: make sure you have all permissions on autom4te.cache...

```
peter@PC1:~/CS2/H7-test2$ aclocal
peter@PC1:~/CS2/H7-test2$ ls
aclocal.m4       configure          doc       man
autom4te.cache   configure.ac       examples  scripts
autoscan.log     configure.ac.first  Makefile.am  src
peter@PC1:~/CS2/H7-test2$
```

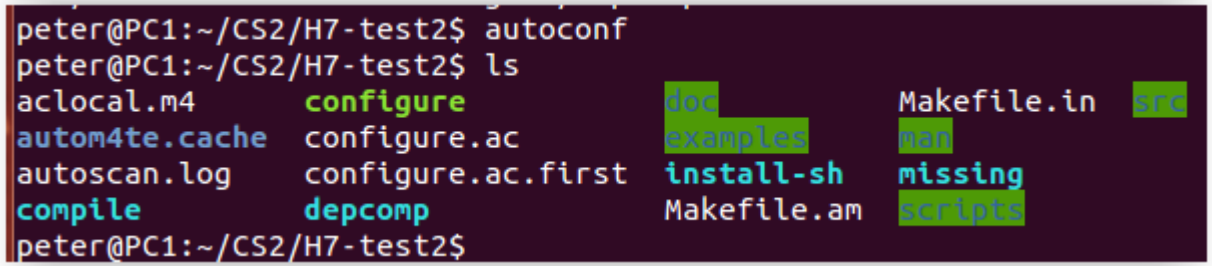
Autoconf & Automake

- Produce Makefile.in from Makefile.am and see what's missing…
- Automake –add-mising

```
peter@PC1:~/CS2/H7-test2$ automake --add-missing
configure.ac:6: warning: AM_INIT_AUTOMAKE: two- and three-argument
s forms are deprecated.  For more info, see:
configure.ac:6: http://www.gnu.org/software/automake/manual/automa
ke.html#Modernize-AM_005fINIT_005fAUTOMAKE-invocation
configure.ac:12: installing './compile'
configure.ac:6: installing './install-sh'
configure.ac:6: installing './missing'
src/Makefile.am: installing './depcomp'
peter@PC1:~/CS2/H7-test2$
```

Autoconf & Automake

- Run autoconf to create the full configuration script.



Autoconf & Automake

- Now you can run:
- ./configure
- make
- sudo make install
- … and the binary will be created in /src…

```
peter@PC1:~/CS2/H7-test2$ cd src
peter@PC1:~/CS2/H7-test2/src$ ls
helloworld    helloworld.o  Makefile.am
helloworld.c  Makefile      Makefile.in
peter@PC1:~/CS2/H7-test2/src$ helloworld
Hello world!
peter@PC1:~/CS2/H7-test2/src$
```