



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.

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

Autoconf & Automake

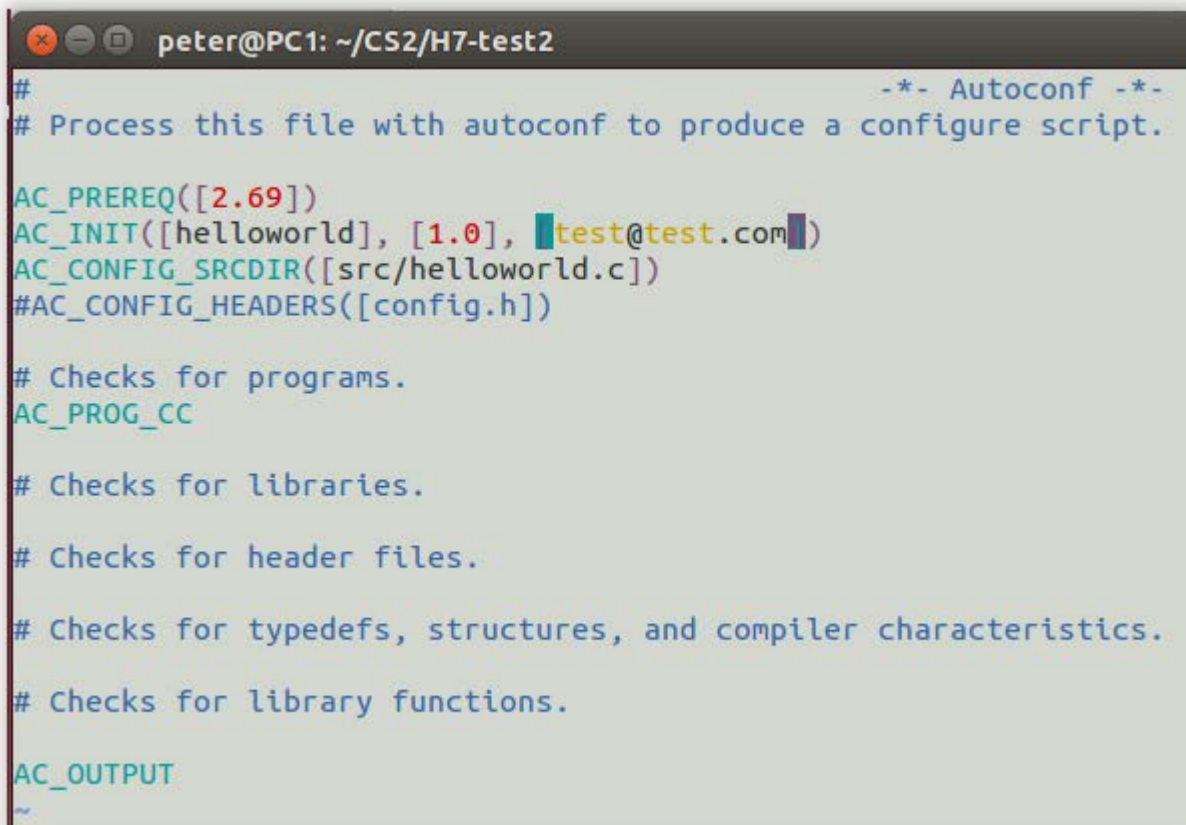
- Autoscan

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

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


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
```

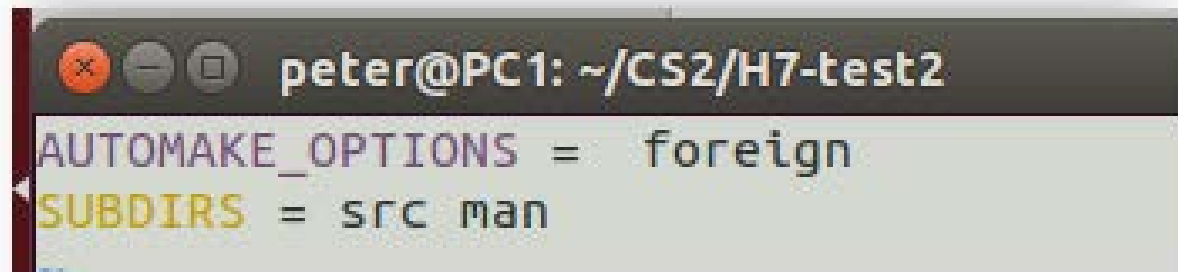
This produces two files: autom4te.cache and configure.

- 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

A screenshot of a terminal window with a dark background. The title bar shows window control buttons and the text 'peter@PC1: ~/CS2/H7-test2'. The terminal displays two lines of text: 'AUTOMAKE_OPTIONS = foreign' and 'SUBDIRS = src man'. The word 'SUBDIRS' is highlighted in yellow, and 'foreign' is in purple.

```
peter@PC1: ~/CS2/H7-test2
AUTOMAKE_OPTIONS = foreign
SUBDIRS = src man
```

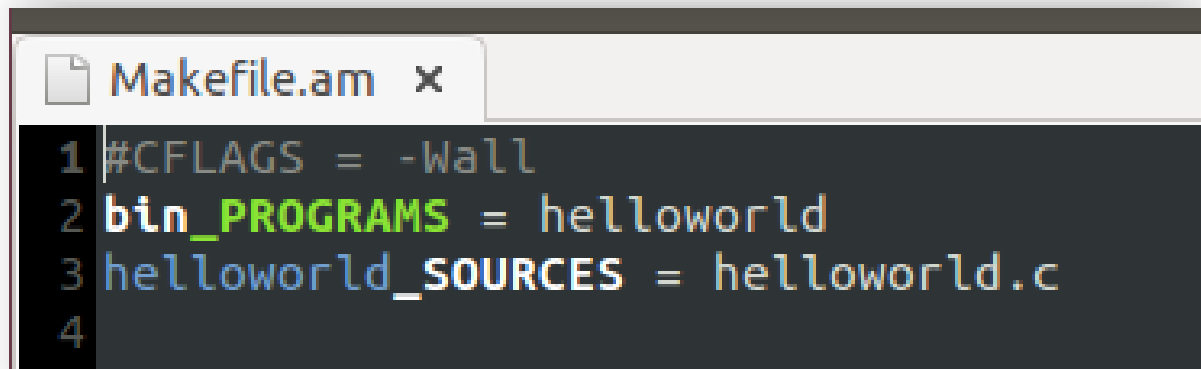
the first line sets the mode automake will behave like. "foreign" means not GNU, and is common for avoiding boring messages about files organized differently from what gnu expects.

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



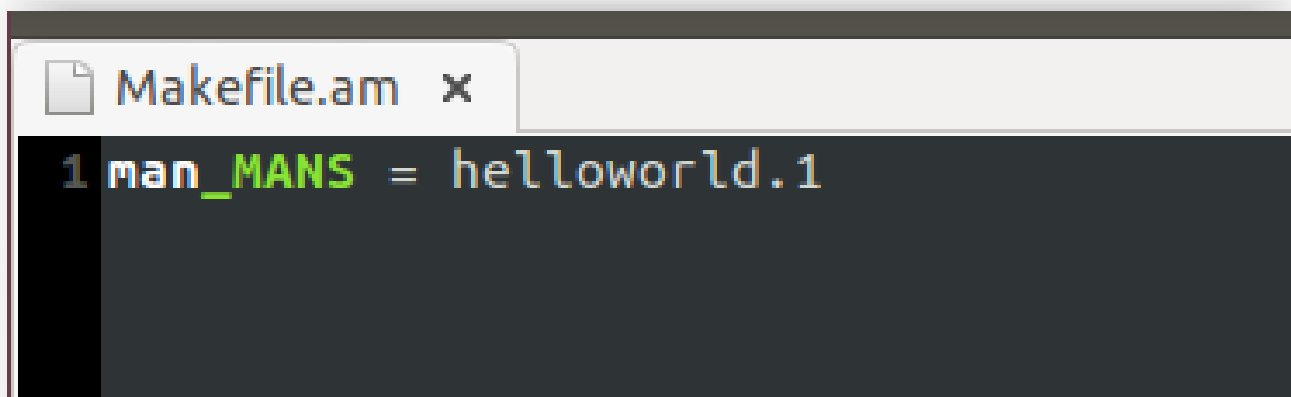
A screenshot of a Vim editor window. The title bar at the top shows a file icon, the name 'Makefile.am', and a close button 'x'. The editor area has a dark background with light-colored text. The text is as follows:

```
1 #CFLAGS = -Wall
2 bin_PROGRAMS = helloworld
3 helloworld_SOURCES = helloworld.c
4
```

Autoconf & Automake

- Generate Makefile:

- Vim man/Makefile.am



A screenshot of a Vim editor window. The title bar at the top shows a file icon, the name 'Makefile.am', and a close button 'x'. The editor area has a dark background and displays the text '1 man_MANS = helloworld.1' on the first line. The line number '1' is in light blue, 'man_MANS' is in green, and '= helloworld.1' is in light blue.

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

- If you are using Autoconf and Automake, you can use the following code in your `configure.ac` file to generate a `configure` script.
 - A snippet of `configure.ac` code:
- ```
-*- 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-missing

```
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/automake.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.

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

# 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$
```