Autotools and GNU Build system

YAKA *2008*

EPITA

12/12/2006

Licence

Outline

Deal with portability issues

Consider C functions...

- that do not exist everywhere (e.g., `setenv`)
- that have different names (e.g., `strchr()` vs. `index()`)
- that have varying prototypes
  (e.g., `int setpgrp(void);` vs. `int setpgrp(int, int);`)
- that can behave differently (e.g., `malloc(0);`, `free(NULL);`)
- that might require other libraries
  (is `pow()` in *libm.so* or in *libc.so*?)
- that can be defined in different headers
  (*string.h* vs. *strings.h* vs. *memory.h*)

How to handle that?

Standard build systems

People are used to a standard installation procedure:

```
~ % tar zxf amhello-1.0.tar.gz
~ % cd amhello-1.0
~/amhello-1.0 % ./configure
...
~/amhello-1.0 % make
...
~/amhello-1.0 % make check
...
~/amhello-1.0 % su
Password:
~/amhello-1.0 # make install
...
```

Standard build systems

People are used to a standard installation procedure:

```
~ % tar zxf amhello-1.0.tar.gz
~ % cd amhello-1.0
~/amhello-1.0 % ./configure
...
~/amhello-1.0 % make
...
~/amhello-1.0 % make check
...
~/amhello-1.0 % su
Password:
~/amhello-1.0 # make install
...
```

Standard build systems

People are used to a standard installation procedure:

```
~ % tar zxf amhello-1.0.tar.gz
~ % cd amhello-1.0
~/amhello-1.0 % ./configure
...
~/amhello-1.0 % make
...
~/amhello-1.0 % make check
...
~/amhello-1.0 % su
Password:
~/amhello-1.0 # make install
...
```

Standard build systems

People are used to a standard installation procedure:

```
~ % tar zxf amhello-1.0.tar.gz
~ % cd amhello-1.0
~/amhello-1.0 % ./configure
...
~/amhello-1.0 % make
...
~/amhello-1.0 % make check
...
~/amhello-1.0 % su
Password:
~/amhello-1.0 # make install
...
```

Standard build systems

People are used to a standard installation procedure:

```
~ % tar zxf amhello-1.0.tar.gz
~ % cd amhello-1.0
~/amhello-1.0 % ./configure
...
~/amhello-1.0 % make
...
~/amhello-1.0 % make check
...
~/amhello-1.0 % su
Password:
~/amhello-1.0 # make install
...
```

Standard build systems

People are used to a standard installation procedure:

```
~ % tar zxf amhello-1.0.tar.gz
~ % cd amhello-1.0
~/amhello-1.0 % ./configure
...
~/amhello-1.0 % make
...
~/amhello-1.0 % make check
...
~/amhello-1.0 % su
Password:
~/amhello-1.0 # make install
...
```

Standard build systems

People are used to a standard installation procedure:

```
~ % tar zxf amhello-1.0.tar.gz
~ % cd amhello-1.0
~/amhello-1.0 % ./configure
...
~/amhello-1.0 % make
...
~/amhello-1.0 % make check
...
~/amhello-1.0 % su
Password:
~/amhello-1.0 # make install
...
```

Standard targets

You don't want to bother to create all these rules:

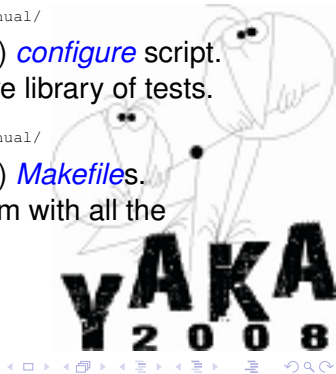|	
`make install`	Install what needs to be installed.
`make uninstall`	The opposite of `make install`.
`make clean`	Erase what has been built (the opposite of `make all`).
`make distclean`	Additionally erase anything `./configure` created.
`make check`	Run the test suite, if any.
`make distcheck`	Check the tarball is complete and that VPATH builds work.
`make dist`	Create *PACKAGE-VERSION.tar.gz*.

Generate your build system

The autotools help you generate a complete build system.
'autoconf' and 'automake' are the central tools.

'autoconf' http://www.gnu.org/software/autoconf/manual/

- Generates your (portable) *configure* script.
- Provides a comprehensive library of tests.

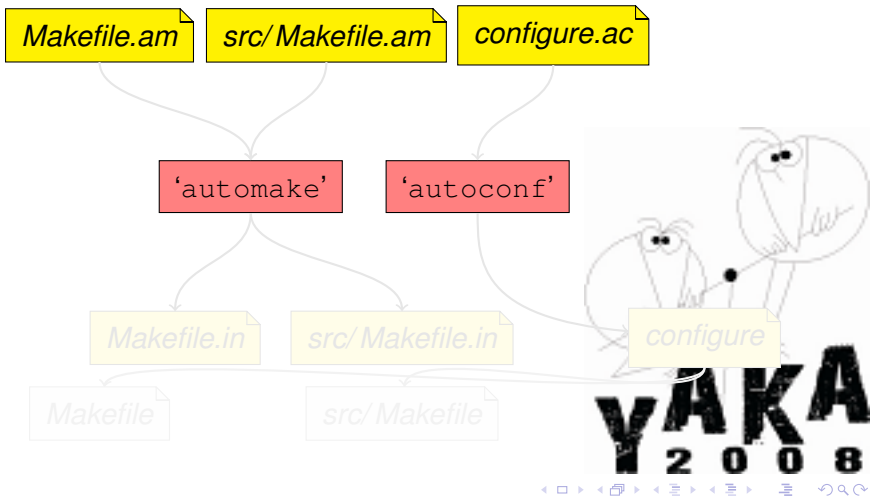**'automake'** http://www.gnu.org/software/automake/manual/

- Generates your (portable) *Makefile*s.
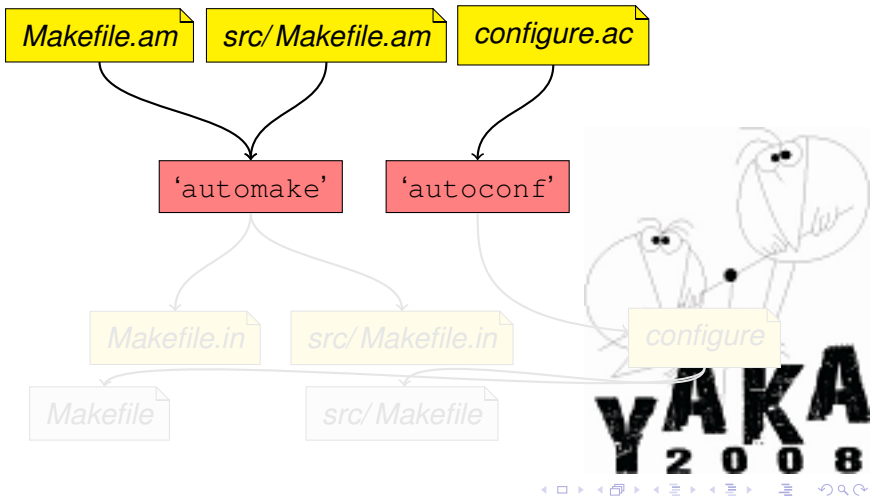- Provides your build system with all the standard rules.

1 **Why use autotools?**

2 **How does it work?**
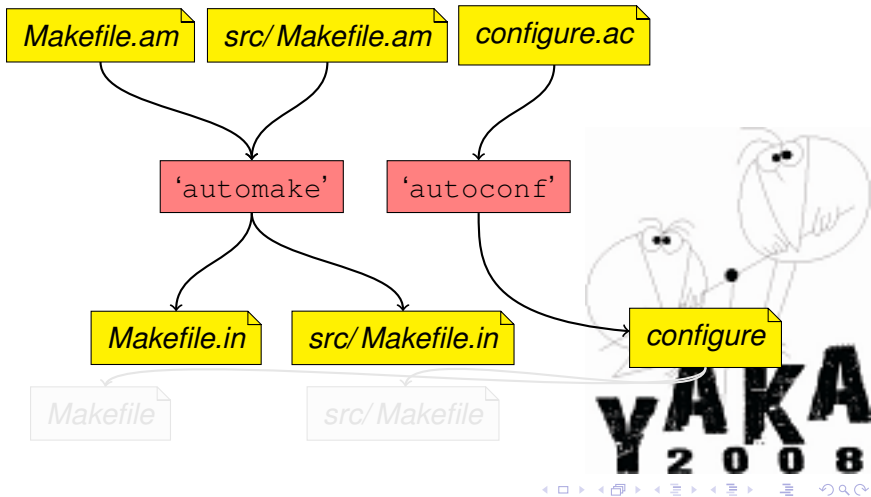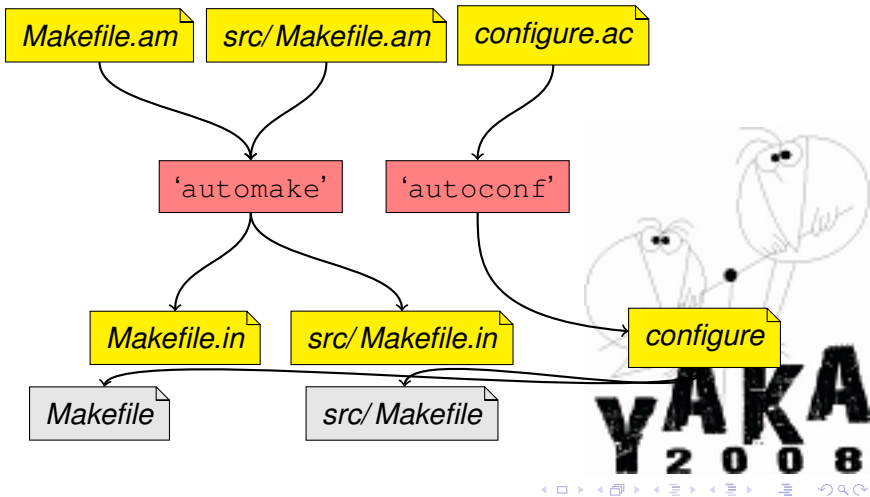
Hello world.

Hello world.

Hello world.

Hello world.

Autotools Inputs

configure.ac

```
AC_INIT([amhello], [1.0],
        [bug-report@address])
AC_CONFIG_AUX_DIR([build-aux])
AM_INIT_AUTOMAKE([
  -Wall -Werror foreign])
AC_PROG_CC
AC_CONFIG_FILES([
  Makefile
  src/Makefile
])
AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
EXTRA_DIST = AUTHORS
```

src/Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = main.c
                hello.h
```

bootstrap

```
#! /bin/sh
autoreconf -f -v -i
```

Autotools Inputs

configure.ac

```
AC_INIT([amhello], [1.0],
        [bug-report@address])
AC_CONFIG_AUX_DIR([build-aux])
AM_INIT_AUTOMAKE([
  -Wall -Werror foreign])
AC_PROG_CC
AC_CONFIG_FILES([
  Makefile
  src/Makefile
])
AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
EXTRA_DIST = AUTHORS
```

src/Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = main.c
                hello.h
```

bootstrap

```
#! /bin/sh
autoreconf -f -v -i
```

Autotools Inputs

configure.ac

```
AC_INIT([amhello], [1.0],
        [bug-report@address])
AC_CONFIG_AUX_DIR([build-aux])
AM_INIT_AUTOMAKE([
  -Wall -Werror foreign])
AC_PROG_CC
AC_CONFIG_FILES([
  Makefile
  src/Makefile
])
AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
EXTRA_DIST = AUTHORS
```

src/Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = main.c \
                hello.h
```

bootstrap

```
#! /bin/sh
autoreconf -f -v -i
```

Autotools Inputs

configure.ac

```
AC_INIT([amhello], [1.0],
        [bug-report@address])
AC_CONFIG_AUX_DIR([build-aux])
AM_INIT_AUTOMAKE([
  -Wall -Werror foreign])
AC_PROG_CC
AC_CONFIG_FILES([
  Makefile
  src/Makefile
])
AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
EXTRA_DIST = AUTHORS
```

src/Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = main.c \
                hello.h
```

bootstrap

```
#! /bin/sh
autoreconf -f -v -i
```

Links

- THE ultimate tutorial:
  `http://www-src.lip6.fr/homepages/Alexandre.`
  `Duret-Lutz/autotools.html`
- `http://www.gnu.org/software/autoconf/`
- `http://www.gnu.org/software/automake/`
- `http://www.gnu.org/software/libtool/`