FOLLOW ALONG WITH THE EXAMPLES…

```
$ git clone https://gitlab.com/jtfrey/unix-software-dev.git
( or "git pull" if you cloned at last session…)

$ git checkout tags/session2

$ ls -l
total 8
-rw-r--r--   1 frey  staff   670B Apr 21 15:23 README.md
drwxr-xr-x  15 frey  staff   510B Apr 21 15:23 src-1
drwxr-xr-x   8 frey  staff   272B Apr 21 15:23 src-2
drwxr-xr-x   8 frey  staff   272B Apr 21 15:23 src-3
drwxr-xr-x   7 frey  staff   238B Apr 21 15:23 src-4
drwxr-xr-x   6 frey  staff   204B Apr 21 15:23 src-5
```

https://gitlab.com/jtfrey/unix-software-dev.git

UNIX SOFTWARE DEVELOPMENT BASICS

AUTOTOOLS & CMAKE

AUTOMATED BUILDS

- We saw a few examples of how to structure a project and author one or more files for the sake of using *make*
- Many aspects of those files were formulaic – after figuring out how to express an action in *make*, future projects can reuse it…
- …so while *make* is great, all that repetitive authoring of Makefiles gets annoying after a few projects.

AUTOMATED BUILDS

- Whenever a process is encountered that is:
  - Repetitive
  - Formulaic
- …you write programs to do that work for you.
- So we desire a tool that takes a simplified description of a code project and produces the files needed to build and deploy its products.

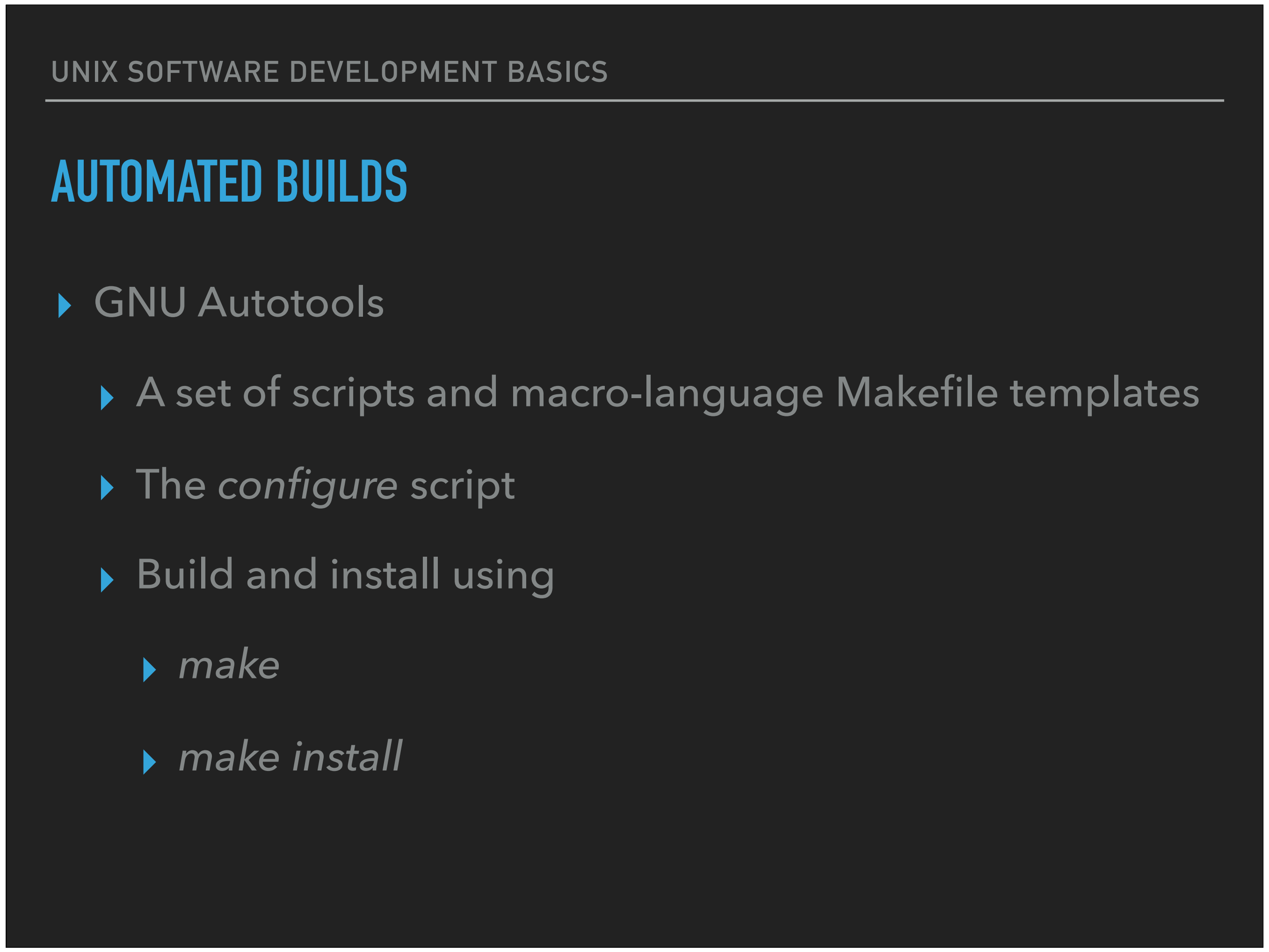
AUTOMATED BUILDS

- GNU Autotools
    - A set of scripts and macro-language Makefile templates
        - Developer describes the project, dependencies and features necessary to build it
        - Uses *auto** tools to create template Makefile(s), support scripts, and *configure* script(s)

https://www.lrde.epita.fr/~adl/dl/autotools.pdf

AUTOMATED BUILDS

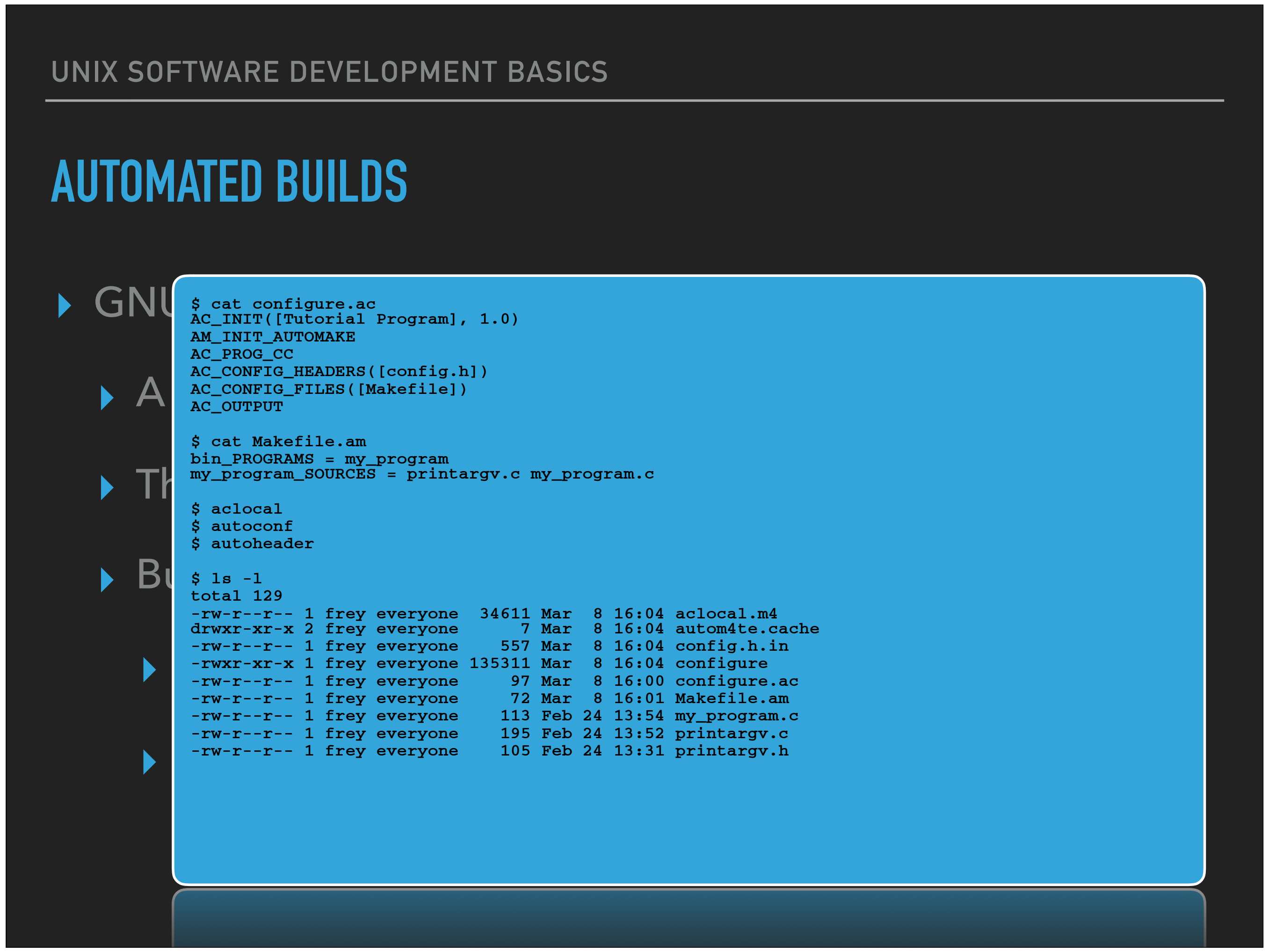
- GNU Autotools
  - A set of scripts and macro-language Makefile templates
  - The *configure* script
    - Analyzes the system on which it is run to satisfy dependencies and features needed
    - Instantiates Makefile(s) from the templates using that information



AUTOMATED BUILDS

- GNU Autotools
  - A set of scripts and macro-language Makefile templates
  - The *configure* script
  - Build and install using
    - *make*
    - *make install*

- The configure.ac file describes the project itself, what libraries or features it requires to build
- The Makefile.am describes the product(s) and the ingredients
- The aclocal command makes a copy of the autoconf tools inside the project
- The autoconf command processes the configure.ac to produce the configure script

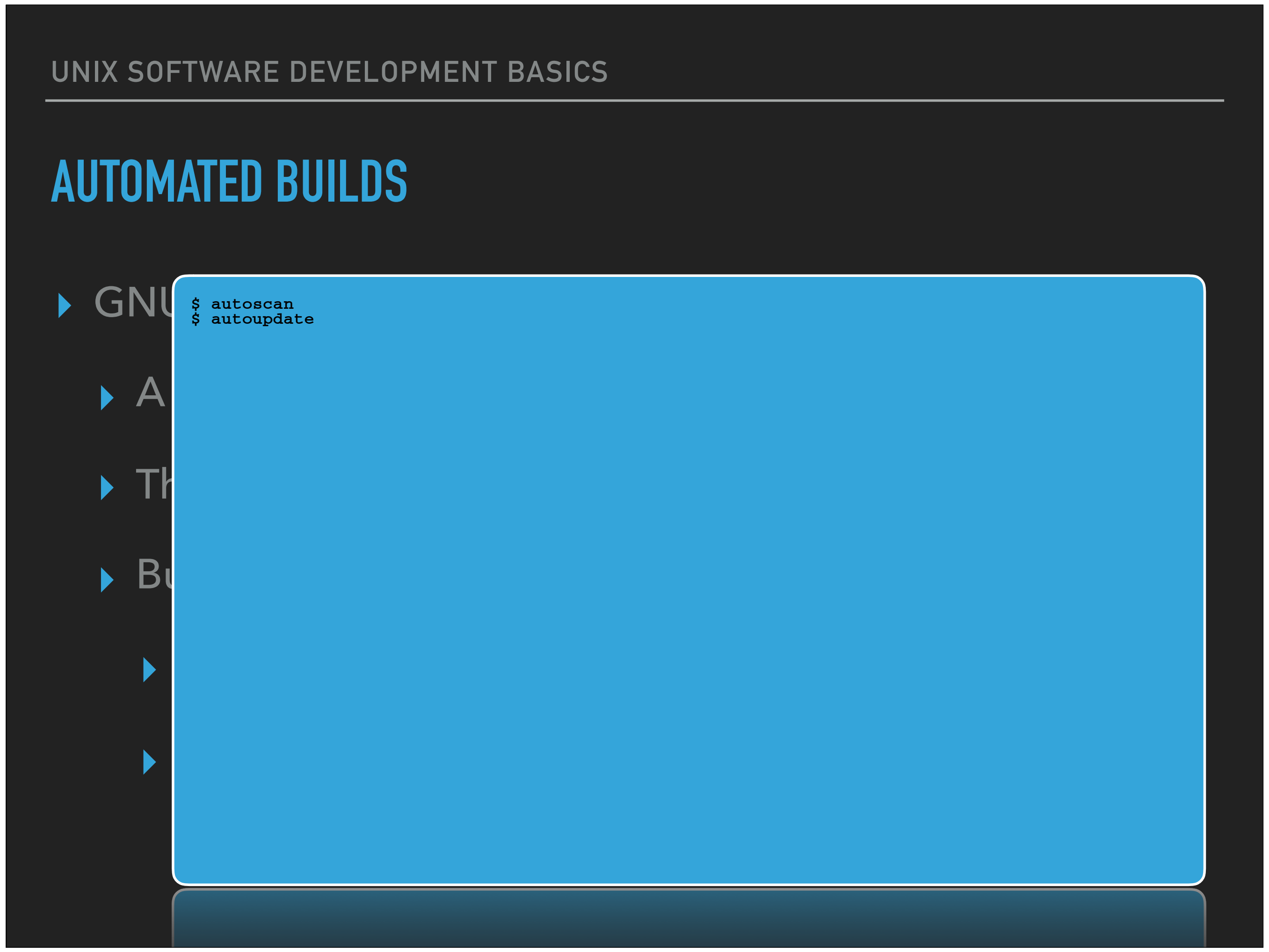


AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

```
$ cat configure.ac
AC_INIT([Tutorial Program], 1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_CONFIG_HEADERS([config.h])
AC_CONFIG_FILES([Makefile])
AC_OUTPUT

$ cat Makefile.am
bin_PROGRAMS = my_program
my_program_SOURCES = printargv.c my_program.c

$ aclocal
$ autoconf
$ autoheader

$ ls -l
total 129
-rw-r--r-- 1 frey everyone  34611 Mar  8 16:04 aclocal.m4
drwxr-xr-x 2 frey everyone      7 Mar  8 16:04 autom4te.cache
-rw-r--r-- 1 frey everyone    557 Mar  8 16:04 config.h.in
-rwxr-xr-x 1 frey everyone 135311 Mar  8 16:04 configure
-rw-r--r-- 1 frey everyone     97 Mar  8 16:00 configure.ac
-rw-r--r-- 1 frey everyone     72 Mar  8 16:01 Makefile.am
-rw-r--r-- 1 frey everyone    113 Feb 24 13:54 my_program.c
-rw-r--r-- 1 frey everyone    195 Feb 24 13:52 printargv.c
-rw-r--r-- 1 frey everyone    105 Feb 24 13:31 printargv.h
```

- The configure.ac file describes the project itself, what libraries or features it requires to build
- The Makefile.am describes the product(s) and the ingredients
- The aclocal command makes a copy of the autoconf tools inside the project
- The autoconf command processes the configure.ac to produce the configure script

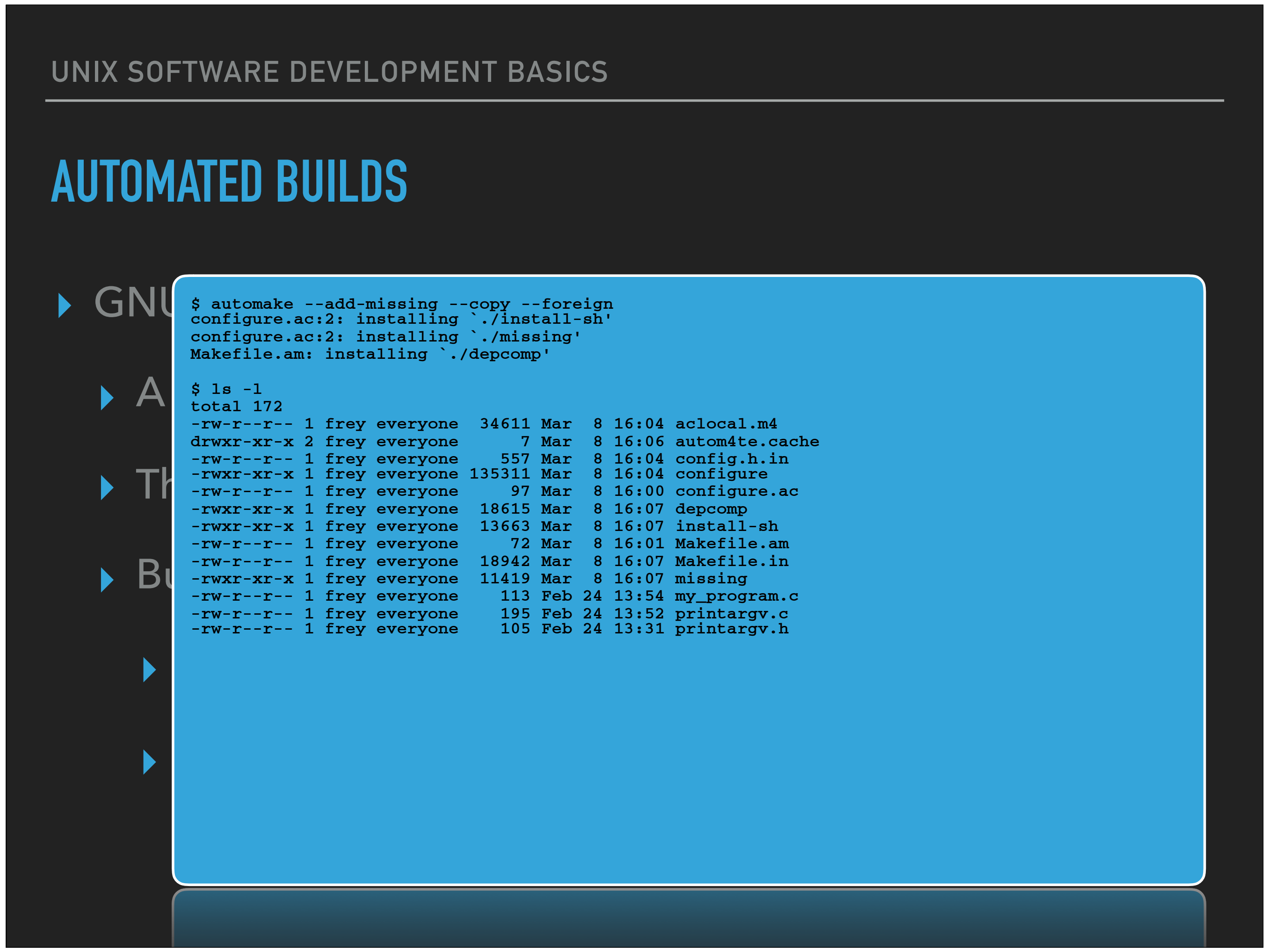


AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

```
$ autoscan
$ autoupdate
```

- autoscan checks source code for portability issues, ensures configure.ac checks for them
- autoupdate checks configure.ac for proper syntax

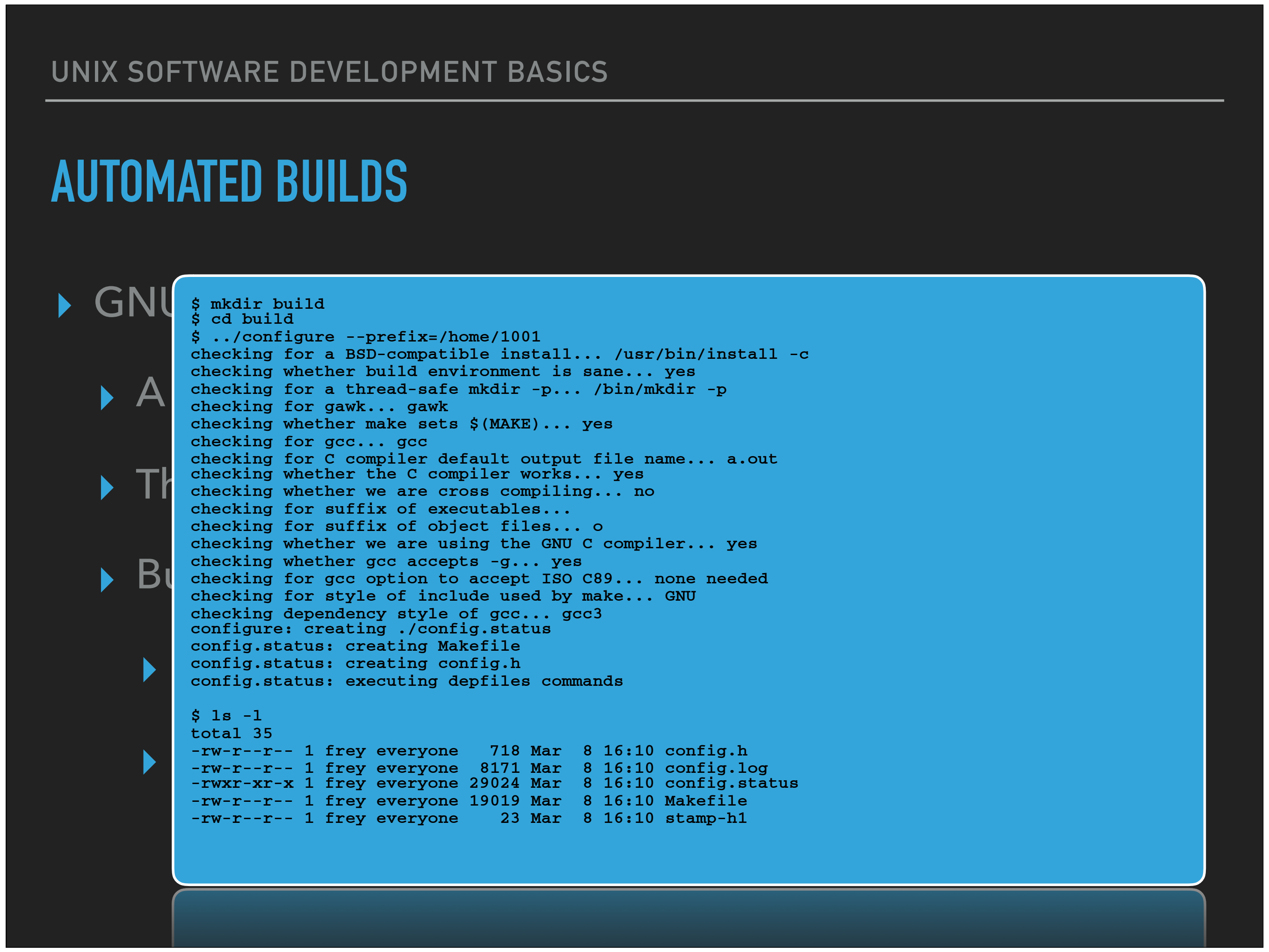


AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

```
$ automake --add-missing --copy --foreign
configure.ac:2: installing `./install-sh'
configure.ac:2: installing `./missing'
Makefile.am: installing `./depcomp'

$ ls -l
total 172
-rw-r--r-- 1 frey everyone  34611 Mar  8 16:04 aclocal.m4
drwxr-xr-x 2 frey everyone      7 Mar  8 16:06 autom4te.cache
-rw-r--r-- 1 frey everyone    557 Mar  8 16:04 config.h.in
-rwxr-xr-x 1 frey everyone 135311 Mar  8 16:04 configure
-rw-r--r-- 1 frey everyone     97 Mar  8 16:00 configure.ac
-rwxr-xr-x 1 frey everyone  18615 Mar  8 16:07 depcomp
-rwxr-xr-x 1 frey everyone  13663 Mar  8 16:07 install-sh
-rw-r--r-- 1 frey everyone     72 Mar  8 16:01 Makefile.am
-rw-r--r-- 1 frey everyone  18942 Mar  8 16:07 Makefile.in
-rwxr-xr-x 1 frey everyone  11419 Mar  8 16:07 missing
-rw-r--r-- 1 frey everyone    113 Feb 24 13:54 my_program.c
-rw-r--r-- 1 frey everyone    195 Feb 24 13:52 printargv.c
-rw-r--r-- 1 frey everyone    105 Feb 24 13:31 printargv.h
```

- The automake command creates the template Makefile(s)
- Also copies into the project any auto tools necessary during the build (e.g. install-sh and depcomp)

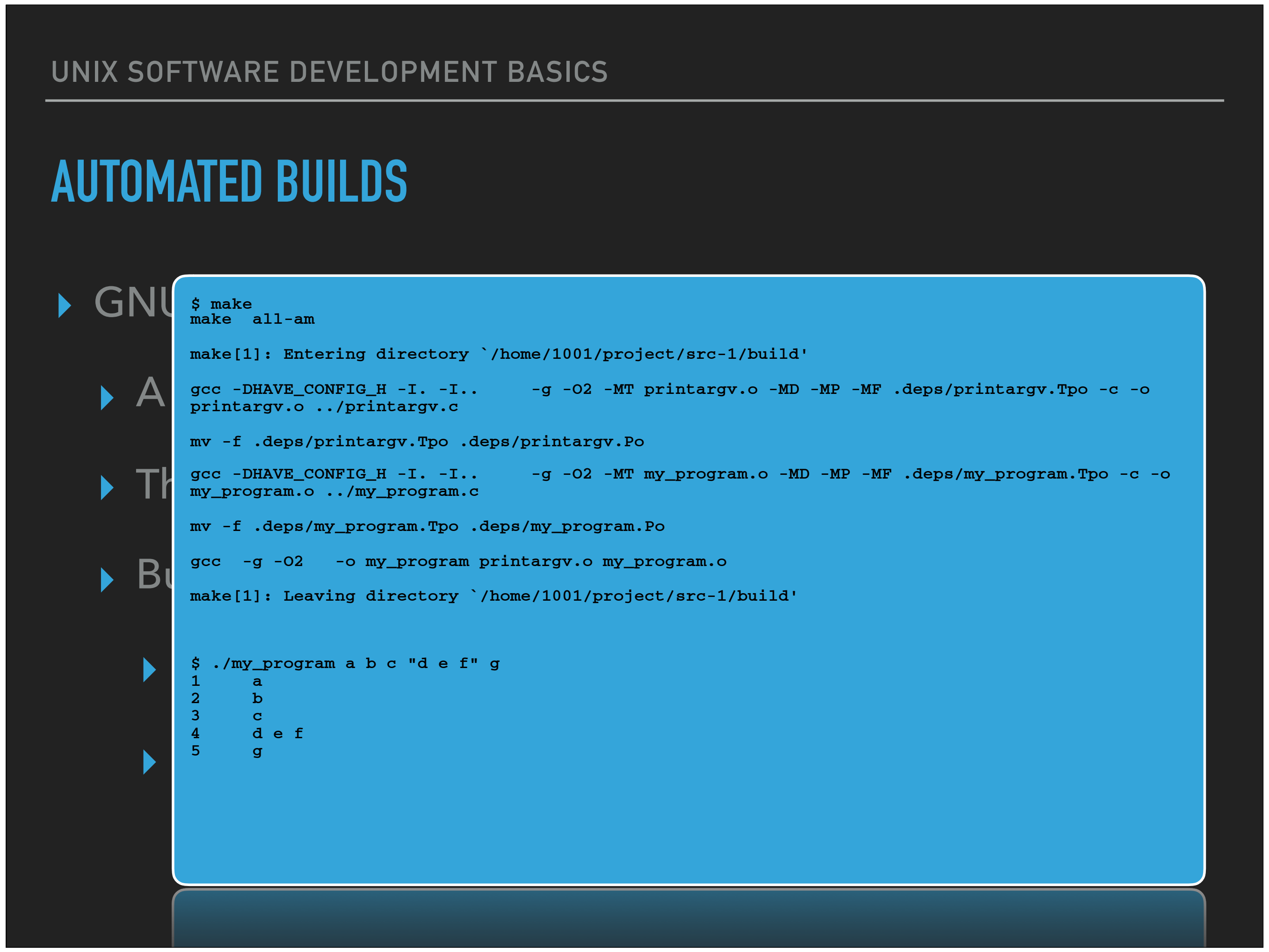


AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

```
$ mkdir build
$ cd build
$ ../configure --prefix=/home/1001
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: executing depfiles commands

$ ls -l
total 35
-rw-r--r-- 1 frey everyone   718 Mar  8 16:10 config.h
-rw-r--r-- 1 frey everyone  8171 Mar  8 16:10 config.log
-rwxr-xr-x 1 frey everyone 29024 Mar  8 16:10 config.status
-rw-r--r-- 1 frey everyone 19019 Mar  8 16:10 Makefile
-rw-r--r-- 1 frey everyone    23 Mar  8 16:10 stamp-h1
```

- Test the result by creating a "build" directory
- Autotools takes care of determining all the paths, so you don't need to build inside the source code itself
  - Our Makefile crafted by hand did the compiling and linking right in with the source code
  - To make an alternate build using e.g. increased optimization I would have to duplicate the entire source tree

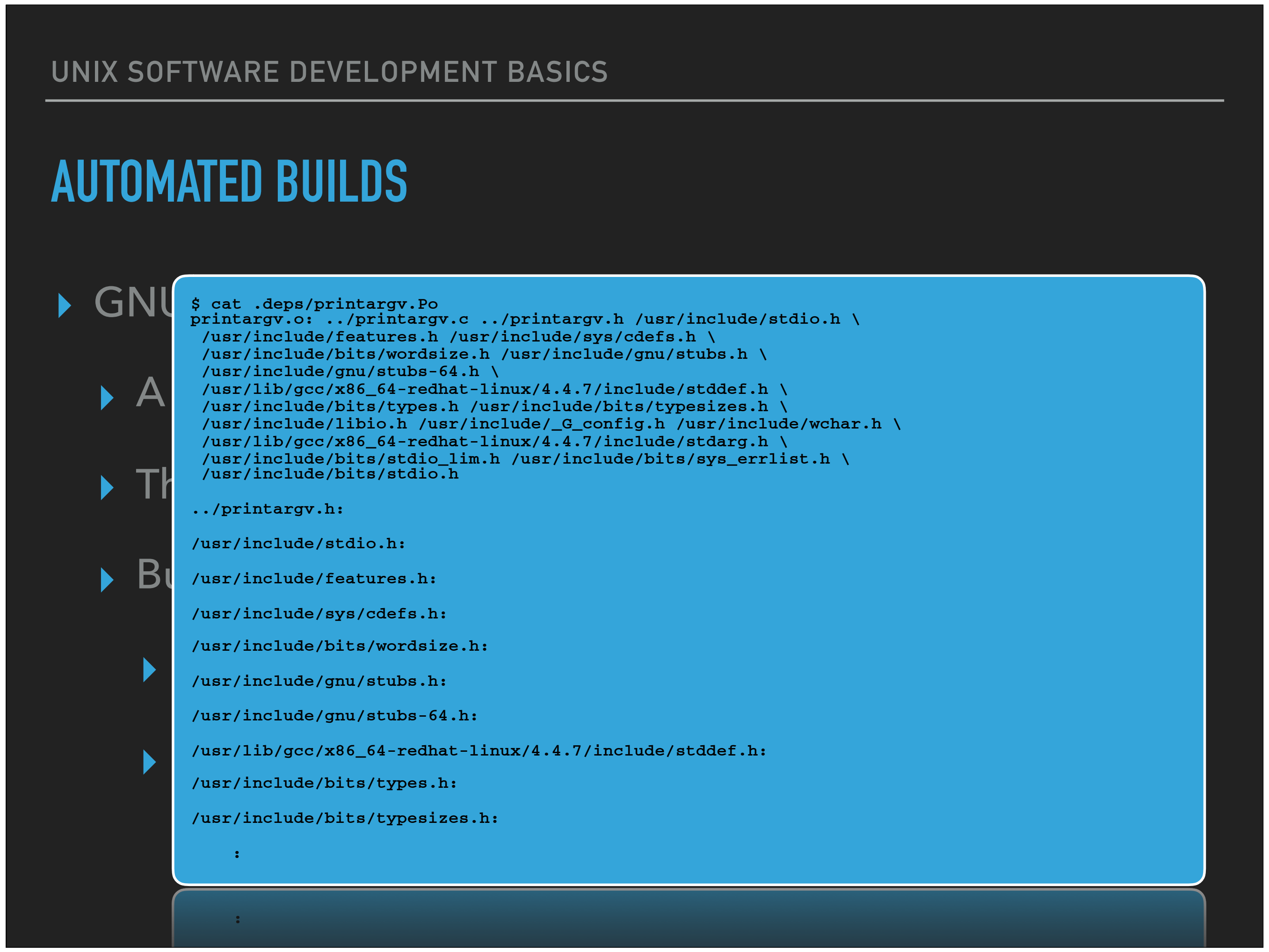


AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

```
$ make
make  all-am

make[1]: Entering directory `/home/1001/project/src-1/build'

gcc -DHAVE_CONFIG_H -I. -I..     -g -O2 -MT printargv.o -MD -MP -MF .deps/printargv.Tpo -c -o
printargv.o ../printargv.c

mv -f .deps/printargv.Tpo .deps/printargv.Po

gcc -DHAVE_CONFIG_H -I. -I..     -g -O2 -MT my_program.o -MD -MP -MF .deps/my_program.Tpo -c -o
my_program.o ../my_program.c

mv -f .deps/my_program.Tpo .deps/my_program.Po

gcc  -g -O2   -o my_program printargv.o my_program.o

make[1]: Leaving directory `/home/1001/project/src-1/build'


$ ./my_program a b c "d e f" g
1      a
2      b
3      c
4      d e f
5      g
```

- Notice the creation of files in ".deps"



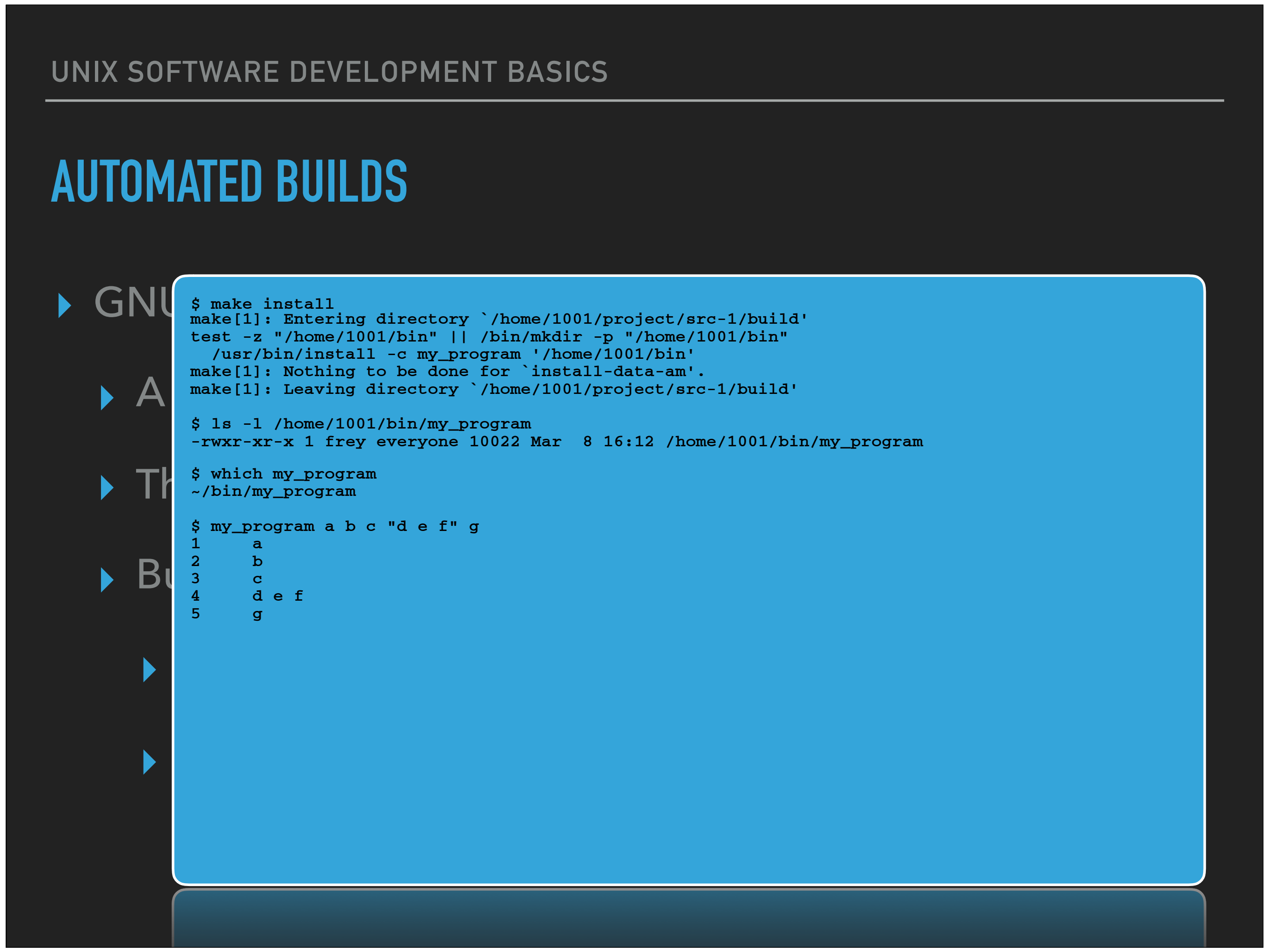
AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

```
$ cat .deps/printargv.Po
printargv.o: ../printargv.c ../printargv.h /usr/include/stdio.h \
 /usr/include/features.h /usr/include/sys/cdefs.h \
 /usr/include/bits/wordsize.h /usr/include/gnu/stubs.h \
 /usr/include/gnu/stubs-64.h \
 /usr/lib/gcc/x86_64-redhat-linux/4.4.7/include/stddef.h \
 /usr/include/bits/types.h /usr/include/bits/typesizes.h \
 /usr/include/libio.h /usr/include/_G_config.h /usr/include/wchar.h \
 /usr/lib/gcc/x86_64-redhat-linux/4.4.7/include/stdarg.h \
 /usr/include/bits/stdio_lim.h /usr/include/bits/sys_errlist.h \
 /usr/include/bits/stdio.h

../printargv.h:

/usr/include/stdio.h:

/usr/include/features.h:

/usr/include/sys/cdefs.h:

/usr/include/bits/wordsize.h:

/usr/include/gnu/stubs.h:

/usr/include/gnu/stubs-64.h:

/usr/lib/gcc/x86_64-redhat-linux/4.4.7/include/stddef.h:

/usr/include/bits/types.h:

/usr/include/bits/typesizes.h:

    :
```

```
    :
```

- Notice the creation of files in ".deps"
  - For each source file processed, a very complete list of dependencies
  - No recipe, just the dependencies
  - All header files and source files that were used, so a system patch that changes /usr/include/stdio.h would also see this being rebuilt on a subsequent "make"

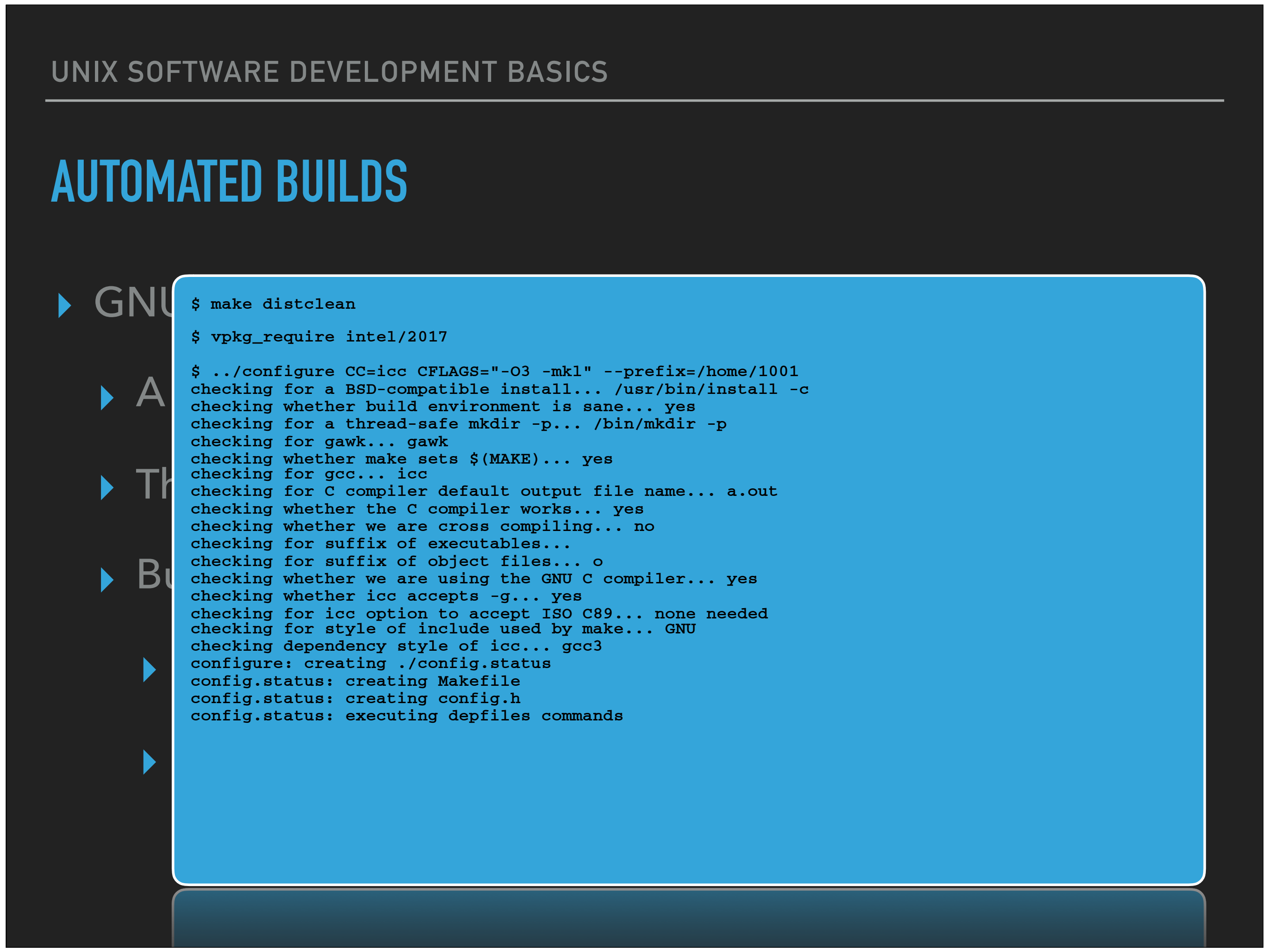


AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

```
$ make install
make[1]: Entering directory `/home/1001/project/src-1/build'
test -z "/home/1001/bin" || /bin/mkdir -p "/home/1001/bin"
  /usr/bin/install -c my_program '/home/1001/bin'
make[1]: Nothing to be done for `install-data-am'.
make[1]: Leaving directory `/home/1001/project/src-1/build'

$ ls -l /home/1001/bin/my_program
-rwxr-xr-x 1 frey everyone 10022 Mar  8 16:12 /home/1001/bin/my_program

$ which my_program
~/bin/my_program

$ my_program a b c "d e f" g
1      a
2      b
3      c
4      d e f
5      g
```

- An "install" target is present in the Makefile



AUTOMATED BUILDS

- GNU
  - A
  - Th
  - Bu

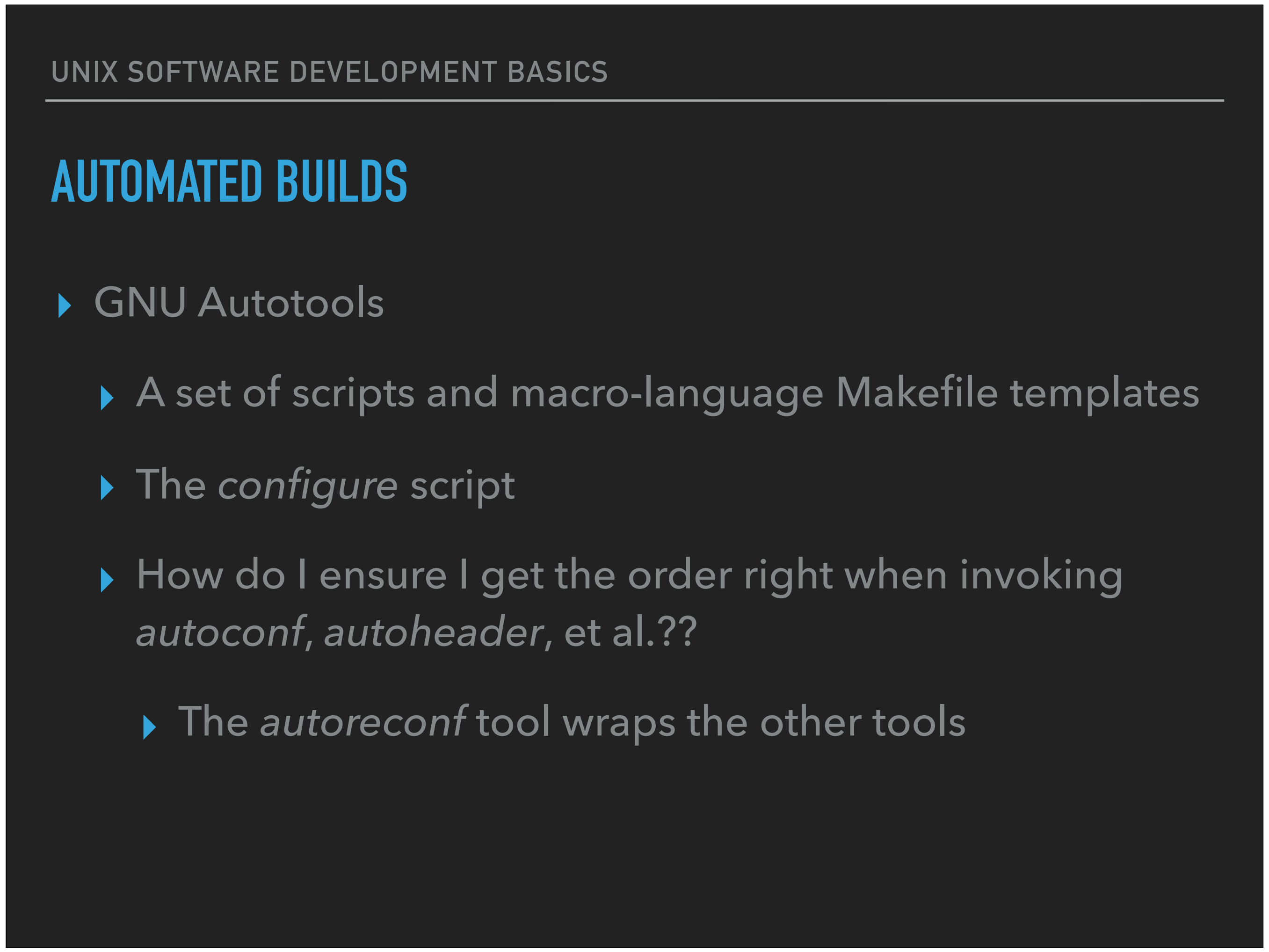
```
$ make distclean

$ vpkg_require intel/2017

$ ../configure CC=icc CFLAGS="-O3 -mkl" --prefix=/home/1001
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... icc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether icc accepts -g... yes
checking for icc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of icc... gcc3
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: executing depfiles commands
```

- A "clean" target is present, as is "distclean" to remove generated Makefile(s), etc.
- Alter the setup:
  - Intel C compiler, use optimization level 3 on all C code and include the MKL libraries

AUTOMATED BUILDS

- GNU
  - A
  - Th
  - B

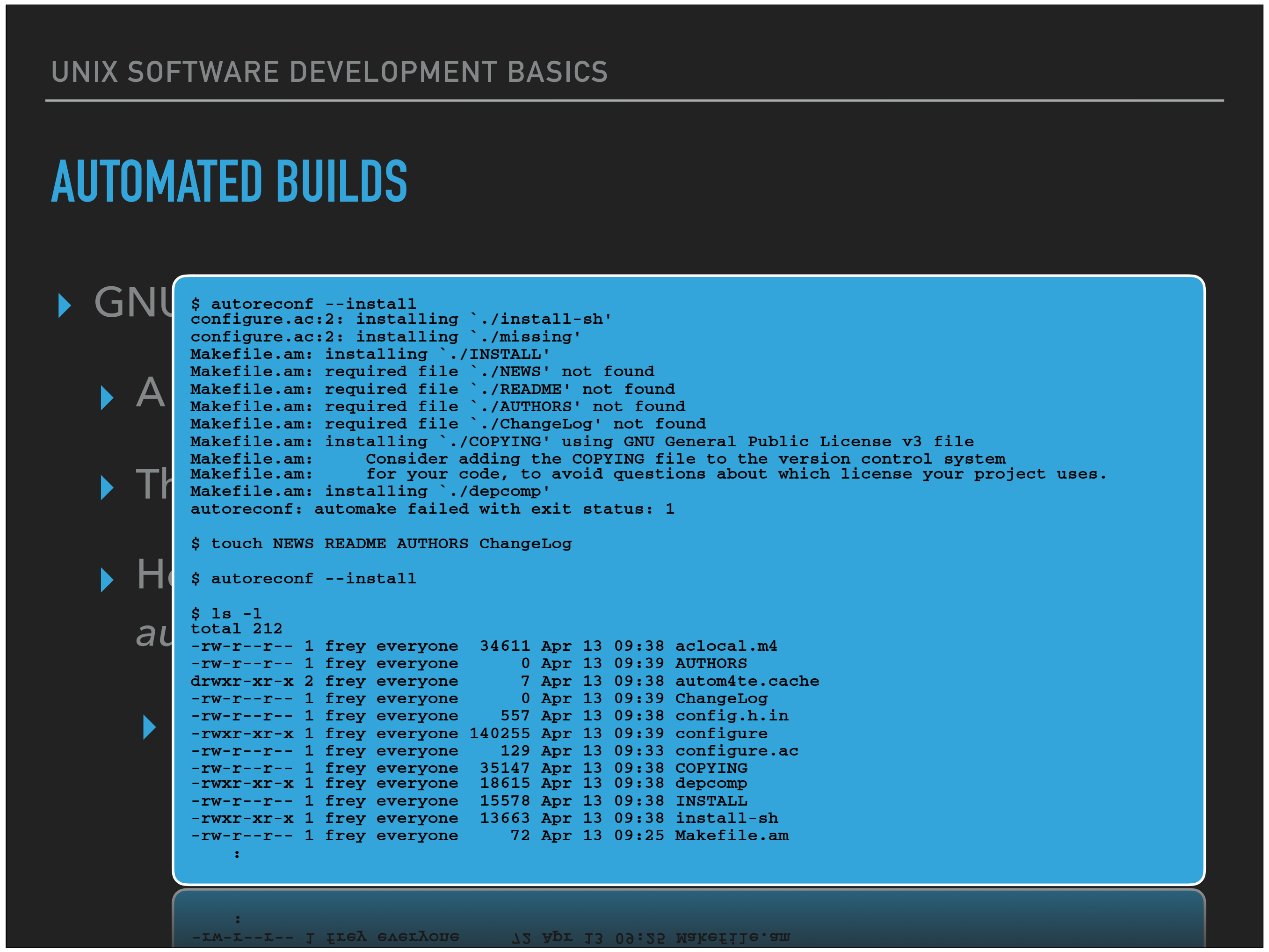
```
$ make
make  all-am

make[1]: Entering directory `/home/1001/project/src-1/build'

icc -DHAVE_CONFIG_H -I. -I..     -O3 -mkl -MT printargv.o -MD -MP -MF .deps/printargv.Tpo -c -o
printargv.o ../printargv.c

mv -f .deps/printargv.Tpo .deps/printargv.Po

icc -DHAVE_CONFIG_H -I. -I..     -O3 -mkl -MT my_program.o -MD -MP -MF .deps/my_program.Tpo -c -
o my_program.o ../my_program.c

mv -f .deps/my_program.Tpo .deps/my_program.Po

icc  -O3 -mkl  -o my_program printargv.o my_program.o

make[1]: Leaving directory `/home/1001/project/src-1/build'

$ ldd my_program
      linux-vdso.so.1 =>  (0x00007ffd80371000)
      libmkl_intel_lp64.so => /opt/shared/intel/2017/compilers_and_libraries_2017.1.132/linux/
mkl/lib/intel64_lin/libmkl_intel_lp64.so (0x00007f19231ab000)
      libmkl_intel_thread.so => /opt/shared/intel/2017/compilers_and_libraries_2017.1.132/linux/
mkl/lib/intel64_lin/libmkl_intel_thread.so (0x00007f192179d000)
      libmkl_core.so => /opt/shared/intel/2017/compilers_and_libraries_2017.1.132/linux/mkl/lib/
intel64_lin/libmkl_core.so (0x00007f191fcf6000)
      libiomp5.so => /opt/shared/intel/2017/compilers_and_libraries_2017.0.098/linux/compiler/
lib/intel64/libiomp5.so (0x00007f191f9ae000)
      libm.so.6 => /lib64/libm.so.6 (0x00000031fa800000)
      libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x0000003202800000)
      libpthread.so.0 => /lib64/libpthread.so.0 (0x00000031fb000000)
            :
```



AUTOMATED BUILDS

- GNU Autotools
  - A set of scripts and macro-language Makefile templates
  - The *configure* script
  - How do I ensure I get the order right when invoking *autoconf*, *autoheader*, et al.??
    - The *autoreconf* tool wraps the other tools

- Assume I'm working with a fresh copy of the src-3 directory…
- autotools tends to produce a LOT of extra files and directories inside your project`



AUTOMATED BUILDS

- GNU
  - A
  - Th
  - H
    au

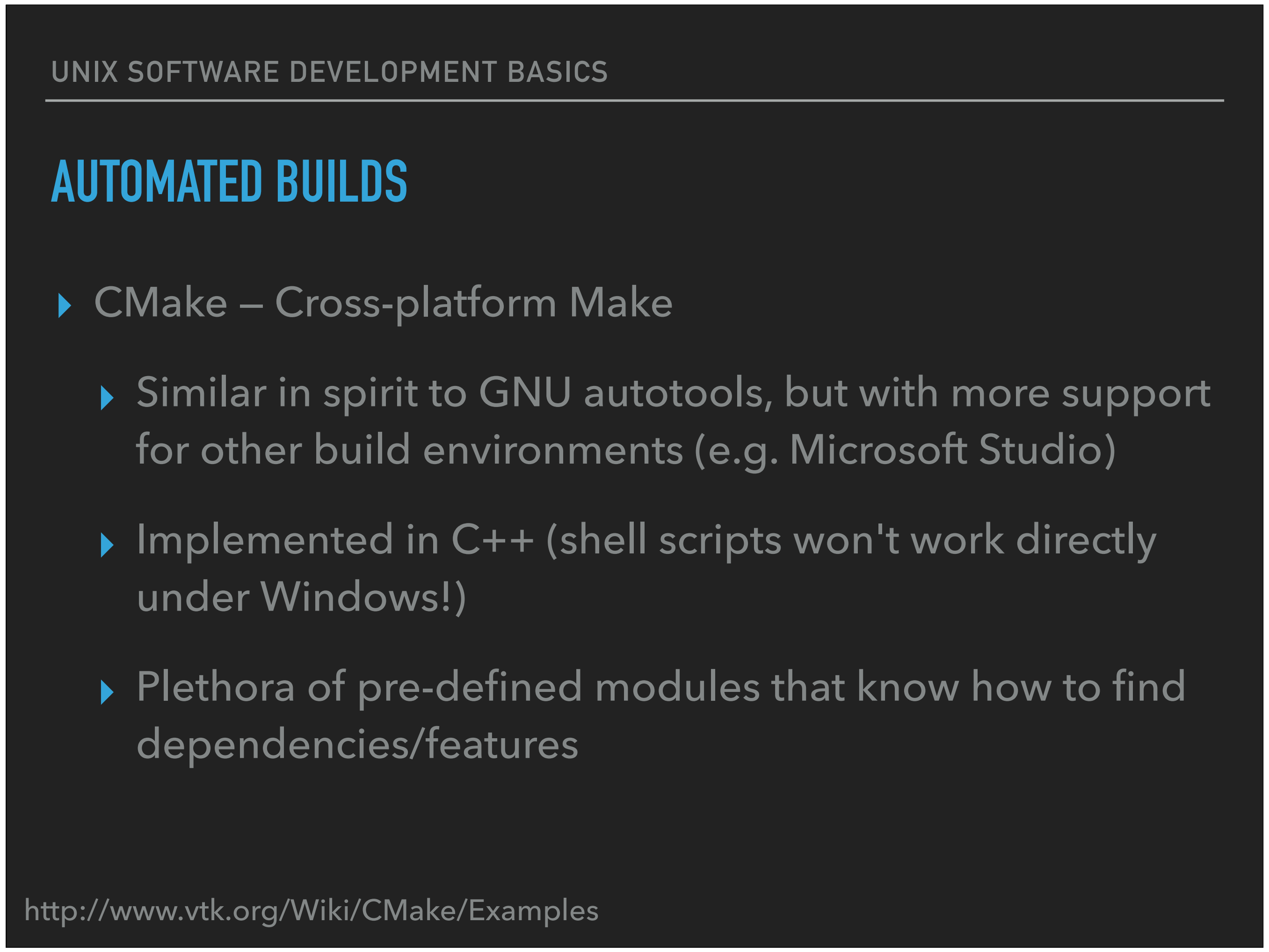
```
$ autoreconf --install
configure.ac:2: installing `./install-sh'
configure.ac:2: installing `./missing'
Makefile.am: installing `./INSTALL'
Makefile.am: required file `./NEWS' not found
Makefile.am: required file `./README' not found
Makefile.am: required file `./AUTHORS' not found
Makefile.am: required file `./ChangeLog' not found
Makefile.am: installing `./COPYING' using GNU General Public License v3 file
Makefile.am:     Consider adding the COPYING file to the version control system
Makefile.am:     for your code, to avoid questions about which license your project uses.
Makefile.am: installing `./depcomp'
autoreconf: automake failed with exit status: 1

$ touch NEWS README AUTHORS ChangeLog

$ autoreconf --install

$ ls -l
total 212
-rw-r--r-- 1 frey everyone  34611 Apr 13 09:38 aclocal.m4
-rw-r--r-- 1 frey everyone      0 Apr 13 09:39 AUTHORS
drwxr-xr-x 2 frey everyone      7 Apr 13 09:38 autom4te.cache
-rw-r--r-- 1 frey everyone      0 Apr 13 09:39 ChangeLog
-rw-r--r-- 1 frey everyone    557 Apr 13 09:38 config.h.in
-rwxr-xr-x 1 frey everyone 140255 Apr 13 09:39 configure
-rw-r--r-- 1 frey everyone    129 Apr 13 09:33 configure.ac
-rw-r--r-- 1 frey everyone  35147 Apr 13 09:38 COPYING
-rwxr-xr-x 1 frey everyone  18615 Apr 13 09:38 depcomp
-rw-r--r-- 1 frey everyone  15578 Apr 13 09:38 INSTALL
-rwxr-xr-x 1 frey everyone  13663 Apr 13 09:38 install-sh
-rw-r--r-- 1 frey everyone     72 Apr 13 09:25 Makefile.am
     :
```

- Assume I'm working with a fresh copy of the src-3 directory…
- autotools tends to produce a LOT of extra files and directories inside your project`

AUTOMATED BUILDS

- GNU Autotools
  - Generates a Unix-like build environment (*make*, et al.)
  - What about e.g. Windows?
    - Just install CYGWIN…
    - Doesn't leverage Windows-native build tools, though
  - So *make* is great for us Unix folks, but we need something more *cross platform* to cover more environments

FOR AN EXAMPLE, SEE `src-3/`

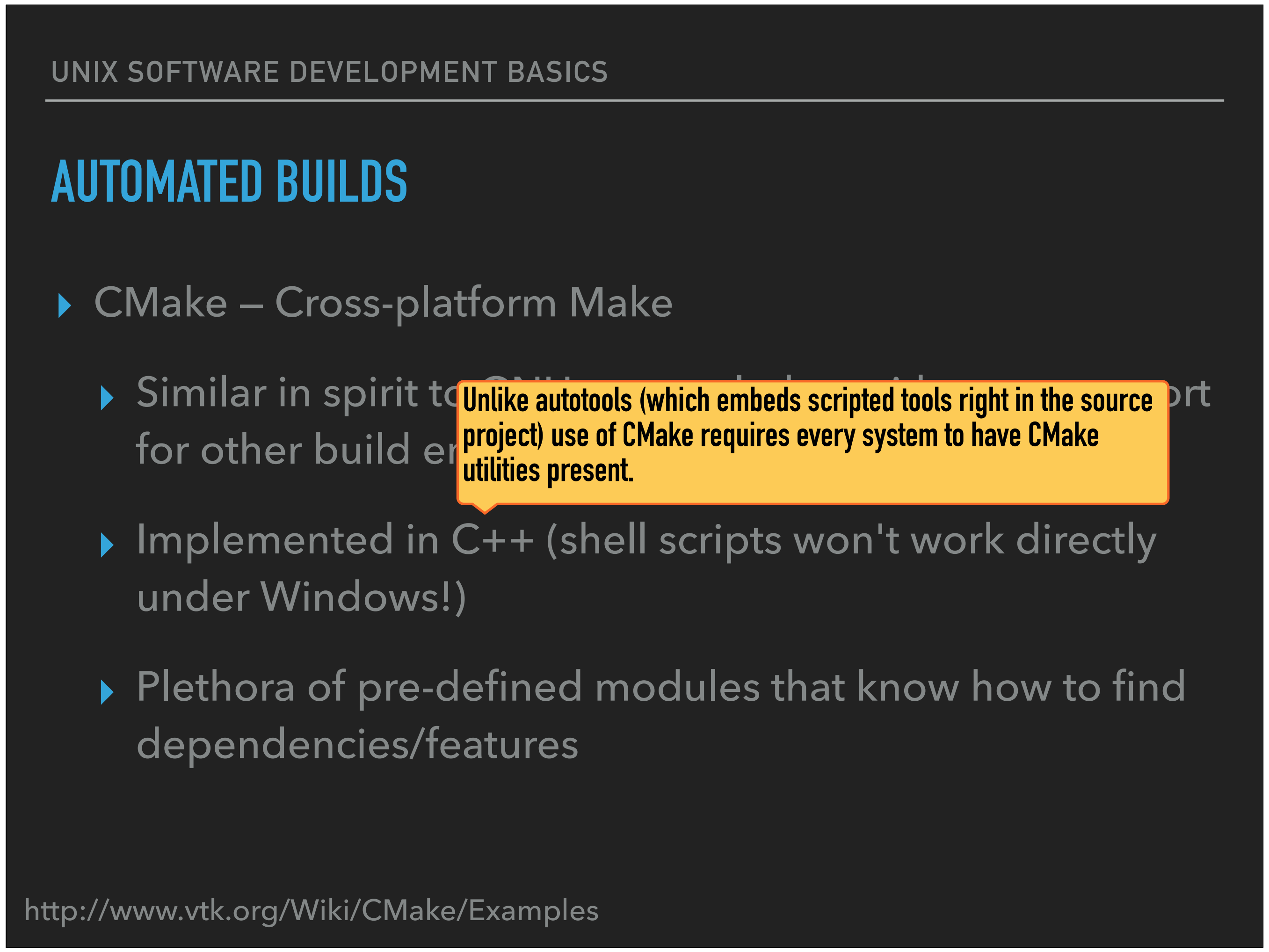


AUTOMATED BUILDS

- CMake – Cross-platform Make
  - Similar in spirit to GNU autotools, but with more support for other build environments (e.g. Microsoft Studio)
  - Implemented in C++ (shell scripts won't work directly under Windows!)
  - Plethora of pre-defined modules that know how to find dependencies/features

http://www.vtk.org/Wiki/CMake/Examples

- Excellent page containing often-used bits of CMake syntax



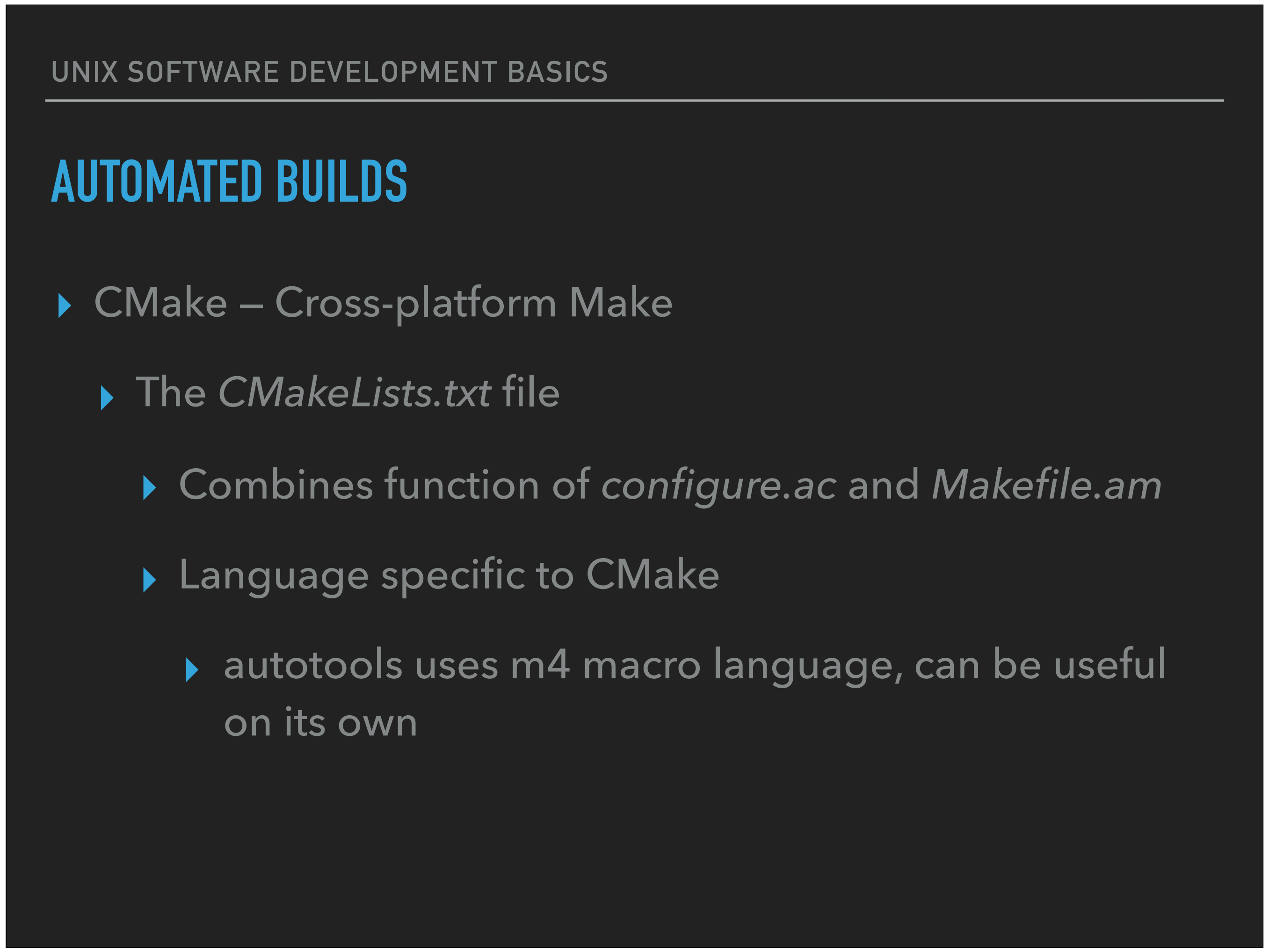
AUTOMATED BUILDS

- CMake – Cross-platform Make
  - Similar in spirit to ort
    for other build en
  - Implemented in C++ (shell scripts won't work directly under Windows!)
  - Plethora of pre-defined modules that know how to find dependencies/features

Unlike autotools (which embeds scripted tools right in the source project) use of CMake requires every system to have CMake utilities present.

http://www.vtk.org/Wiki/CMake/Examples

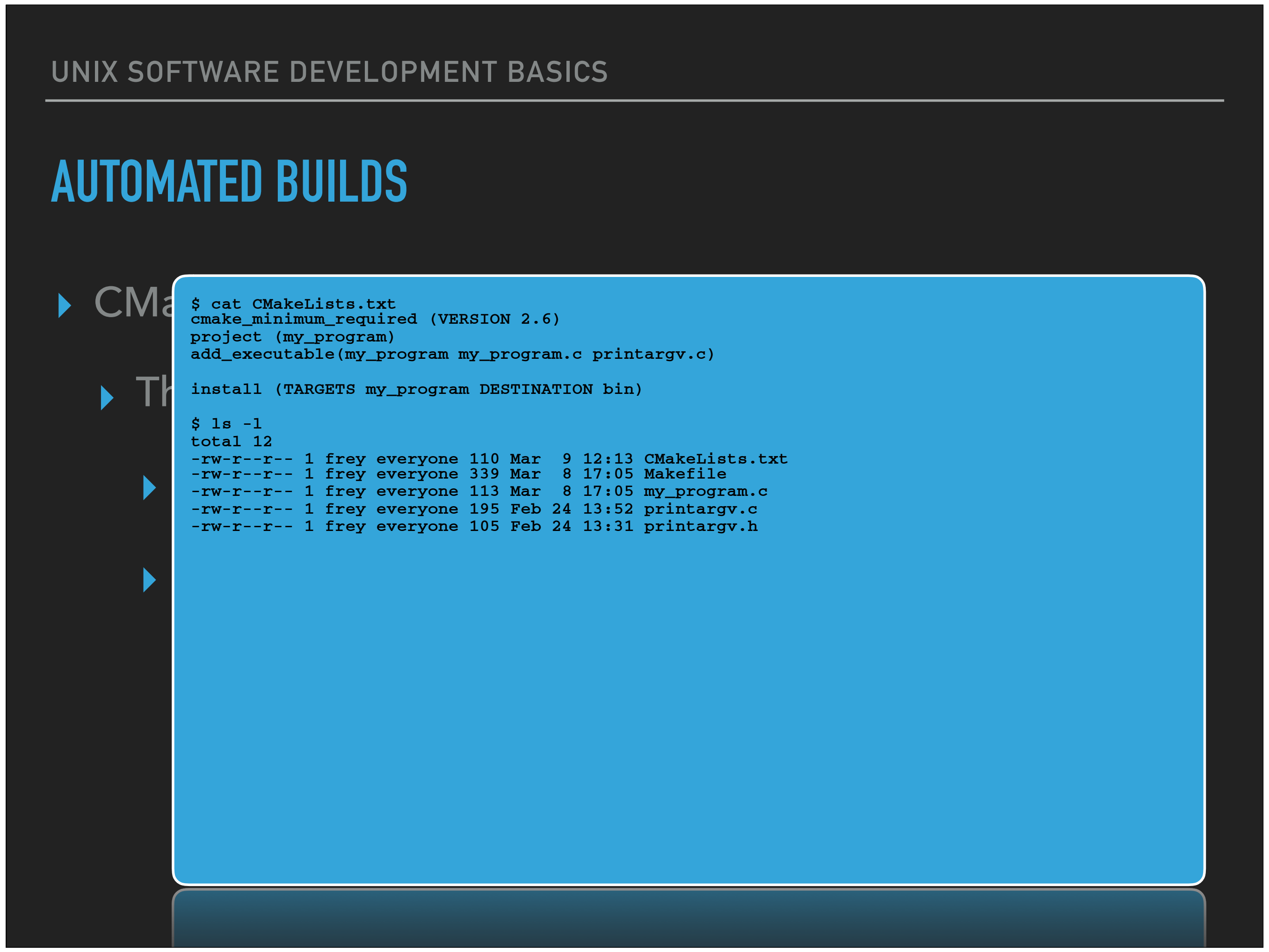
- Excellent page containing often-used bits of CMake syntax



AUTOMATED BUILDS

- CMake – Cross-platform Make
  - The *CMakeLists.txt* file
    - Combines function of *configure.ac* and *Makefile.am*
    - Language specific to CMake
      - autotools uses m4 macro language, can be useful on its own

- No additional files/tools are mandated in your source tree
- Syntax of CMake language is pretty straightforward, where m4 syntax may be harder for you

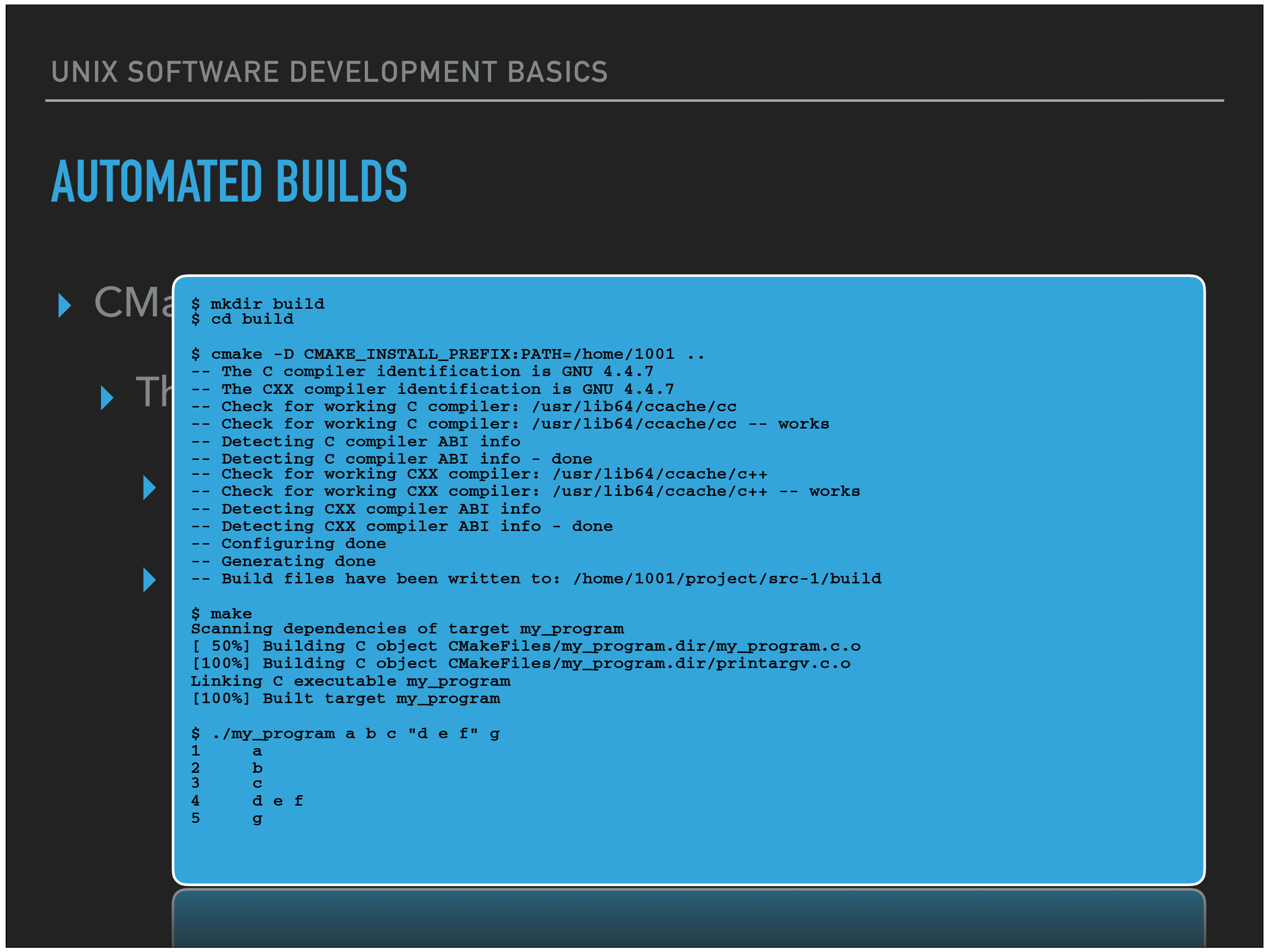


AUTOMATED BUILDS

- CMa
  - Th

```
$ cat CMakeLists.txt
cmake_minimum_required (VERSION 2.6)
project (my_program)
add_executable(my_program my_program.c printargv.c)

install (TARGETS my_program DESTINATION bin)

$ ls -l
total 12
-rw-r--r-- 1 frey everyone 110 Mar  9 12:13 CMakeLists.txt
-rw-r--r-- 1 frey everyone 339 Mar  8 17:05 Makefile
-rw-r--r-- 1 frey everyone 113 Mar  8 17:05 my_program.c
-rw-r--r-- 1 frey everyone 195 Feb 24 13:52 printargv.c
-rw-r--r-- 1 frey everyone 105 Feb 24 13:31 printargv.h
```

- No additional files/tools are mandated in your source tree
- Syntax of CMake language is pretty straightforward, where m4 syntax may be harder for you

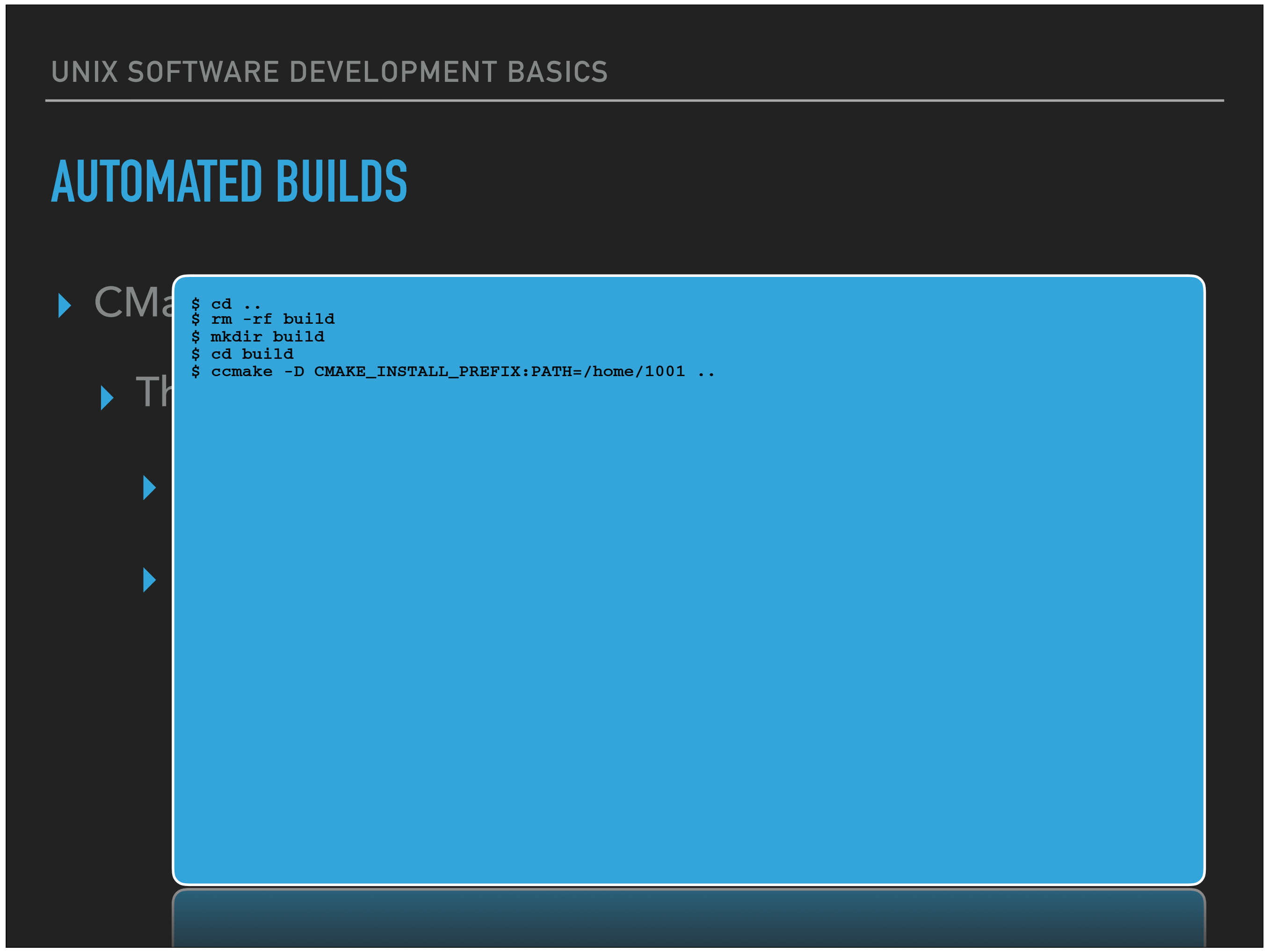


AUTOMATED BUILDS

- CMa
  - Th

```
$ mkdir build
$ cd build

$ cmake -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
-- The C compiler identification is GNU 4.4.7
-- The CXX compiler identification is GNU 4.4.7
-- Check for working C compiler: /usr/lib64/ccache/cc
-- Check for working C compiler: /usr/lib64/ccache/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/lib64/ccache/c++
-- Check for working CXX compiler: /usr/lib64/ccache/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/1001/project/src-1/build

$ make
Scanning dependencies of target my_program
[ 50%] Building C object CMakeFiles/my_program.dir/my_program.c.o
[100%] Building C object CMakeFiles/my_program.dir/printargv.c.o
Linking C executable my_program
[100%] Built target my_program

$ ./my_program a b c "d e f" g
1      a
2      b
3      c
4      d e f
5      g
```

- Again, builds done in a standalone directory, NOT in the source itself
- Build configuration can be guided solely by discovery and CLI options
  - Assign typed values to variables using "-D" arguments

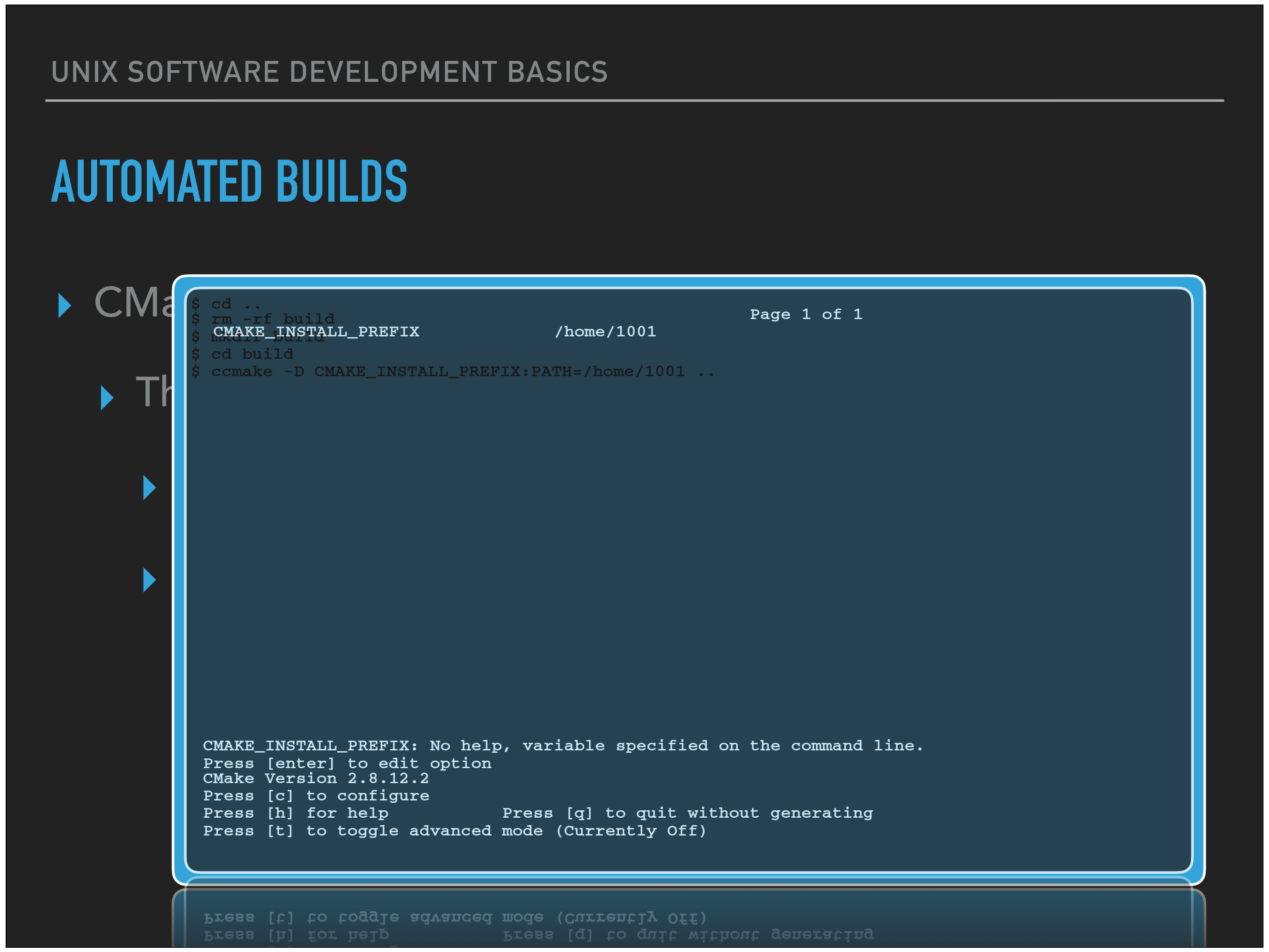


AUTOMATED BUILDS

- CMa
  - Th

```
$ cd ..
$ rm -rf build
$ mkdir build
$ cd build
$ ccmake -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
```

- Build configuration can also be interactive with menu-driven interface
  - "advanced mode" shows all variables and their values
-



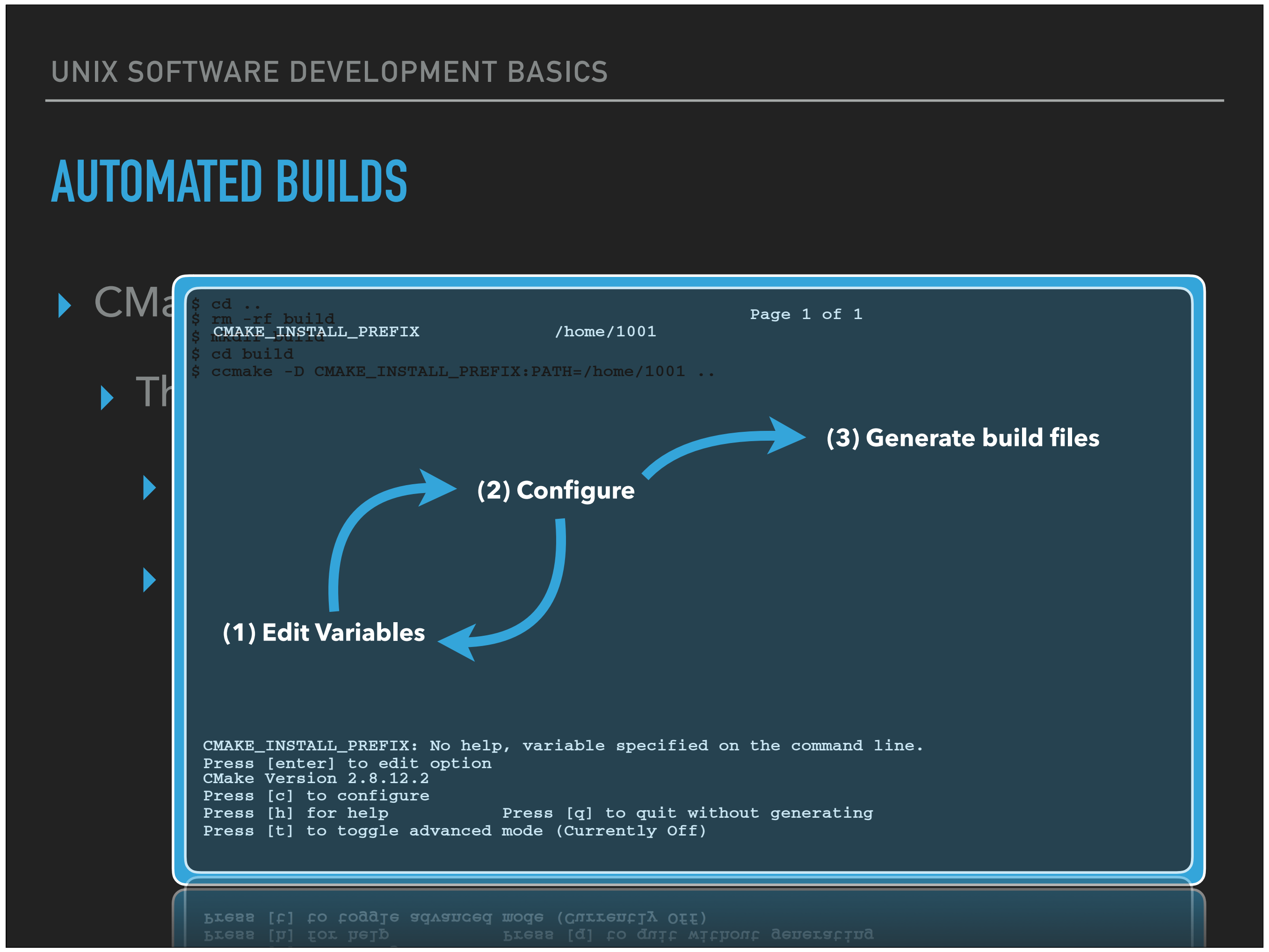
AUTOMATED BUILDS

- CMa
  - Th

```
$ cd ..
$ rm -rf build
$ mkdir build
$ cd build
$ ccmake -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
```

```
                                                      Page 1 of 1
 CMAKE_INSTALL_PREFIX             /home/1001









CMAKE_INSTALL_PREFIX: No help, variable specified on the command line.
Press [enter] to edit option
CMake Version 2.8.12.2
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

- Build configuration can also be interactive with menu-driven interface
  - "advanced mode" shows all variables and their values
-



AUTOMATED BUILDS

- CMa
  - Th

```
$ cd ..
$ rm -rf build
$ mkdir build
$ cd build
$ ccmake -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
```

```
                                                          Page 1 of 1
 CMAKE_INSTALL_PREFIX             /home/1001









CMAKE_INSTALL_PREFIX: No help, variable specified on the command line.
Press [enter] to edit option
CMake Version 2.8.12.2
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

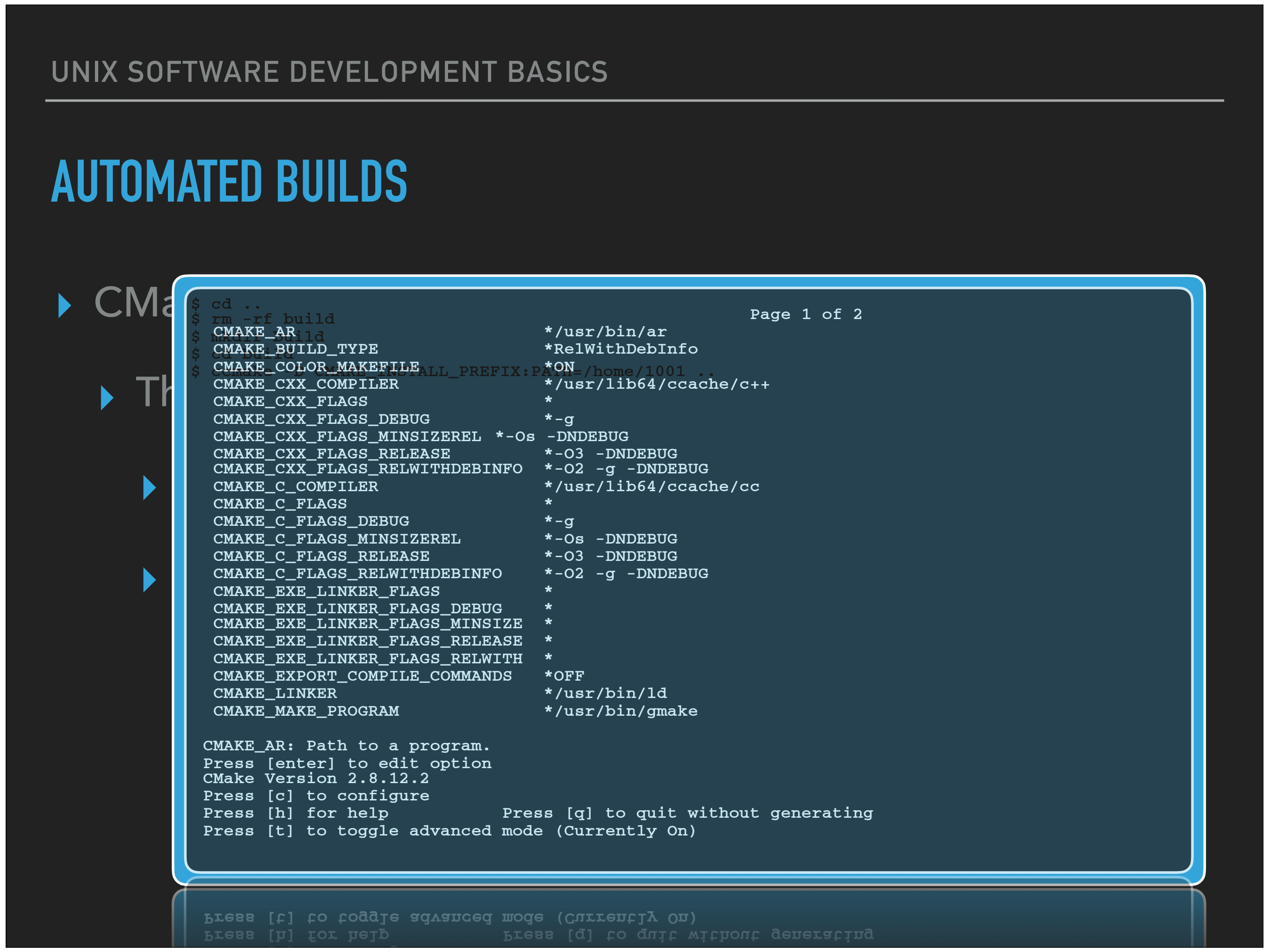
- Build configuration can also be interactive with menu-driven interface
  - "advanced mode" shows all variables and their values
-

AUTOMATED BUILDS

- CMa
  - Th

```
$ cd ..
$ rm -rf build
$ mkdir build
$ cd build
$ ccmake -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
```

- Configure, then toggle to advanced mode
- Use arrow keys to move between variables
  - Hit return/enter to start editing
  - Hit return/enter to keep changes, esc to discard

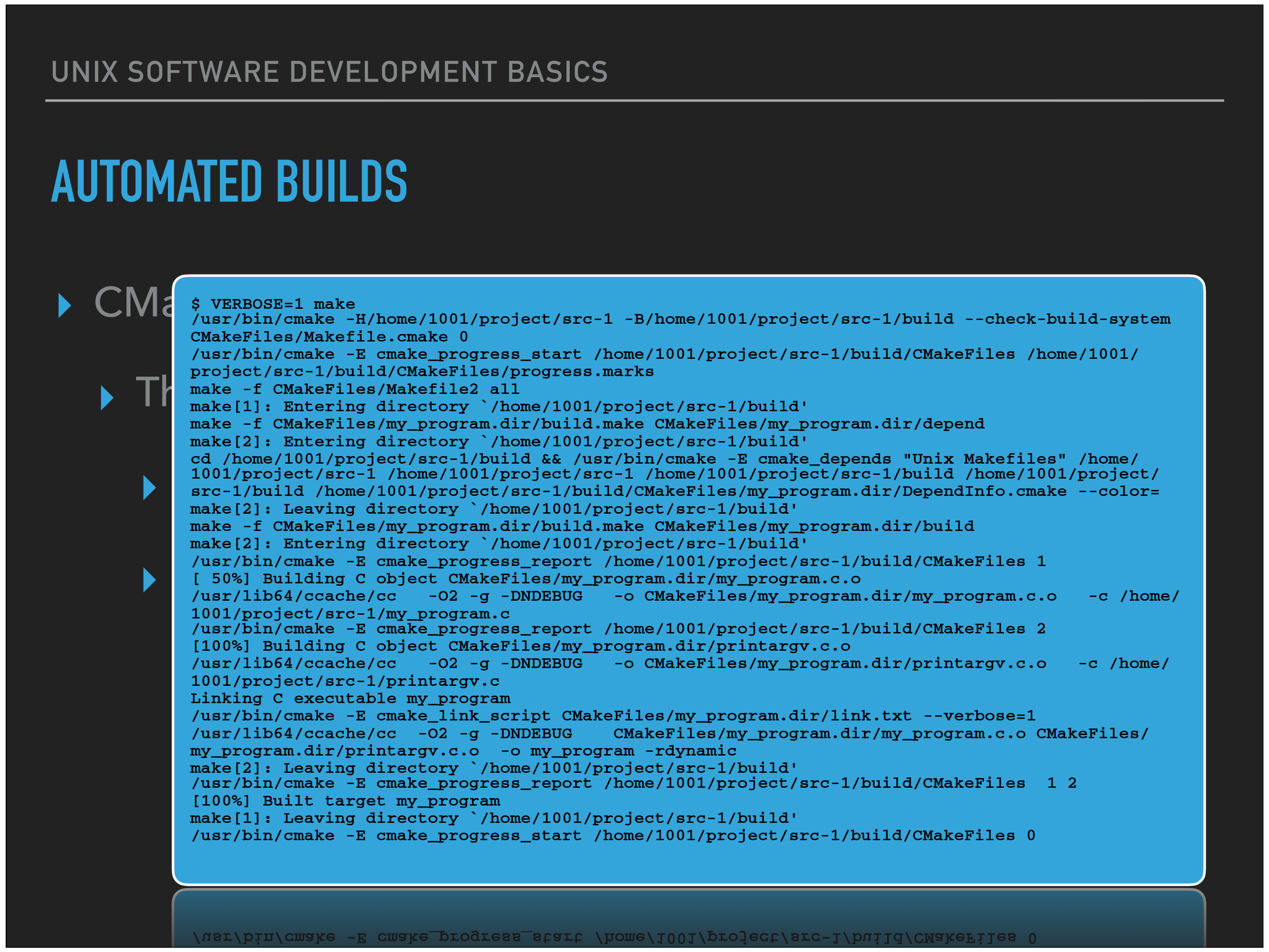


AUTOMATED BUILDS

- CMa
  - Th

```
                                                     Page 1 of 2
 CMAKE_AR                         */usr/bin/ar
 CMAKE_BUILD_TYPE                 *RelWithDebInfo
 CMAKE_COLOR_MAKEFILE             *ON
 CMAKE_CXX_COMPILER               */usr/lib64/ccache/c++
 CMAKE_CXX_FLAGS                  *
 CMAKE_CXX_FLAGS_DEBUG            *-g
 CMAKE_CXX_FLAGS_MINSIZEREL  *-Os -DNDEBUG
 CMAKE_CXX_FLAGS_RELEASE          *-O3 -DNDEBUG
 CMAKE_CXX_FLAGS_RELWITHDEBINFO   *-O2 -g -DNDEBUG
 CMAKE_C_COMPILER                 */usr/lib64/ccache/cc
 CMAKE_C_FLAGS                    *
 CMAKE_C_FLAGS_DEBUG              *-g
 CMAKE_C_FLAGS_MINSIZEREL         *-Os -DNDEBUG
 CMAKE_C_FLAGS_RELEASE            *-O3 -DNDEBUG
 CMAKE_C_FLAGS_RELWITHDEBINFO     *-O2 -g -DNDEBUG
 CMAKE_EXE_LINKER_FLAGS           *
 CMAKE_EXE_LINKER_FLAGS_DEBUG     *
 CMAKE_EXE_LINKER_FLAGS_MINSIZE   *
 CMAKE_EXE_LINKER_FLAGS_RELEASE   *
 CMAKE_EXE_LINKER_FLAGS_RELWITH   *
 CMAKE_EXPORT_COMPILE_COMMANDS    *OFF
 CMAKE_LINKER                     */usr/bin/ld
 CMAKE_MAKE_PROGRAM               */usr/bin/gmake

CMAKE_AR: Path to a program.
Press [enter] to edit option
CMake Version 2.8.12.2
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently On)
```

- Configure, then toggle to advanced mode
- Use arrow keys to move between variables
  - Hit return/enter to start editing
  - Hit return/enter to keep changes, esc to discard

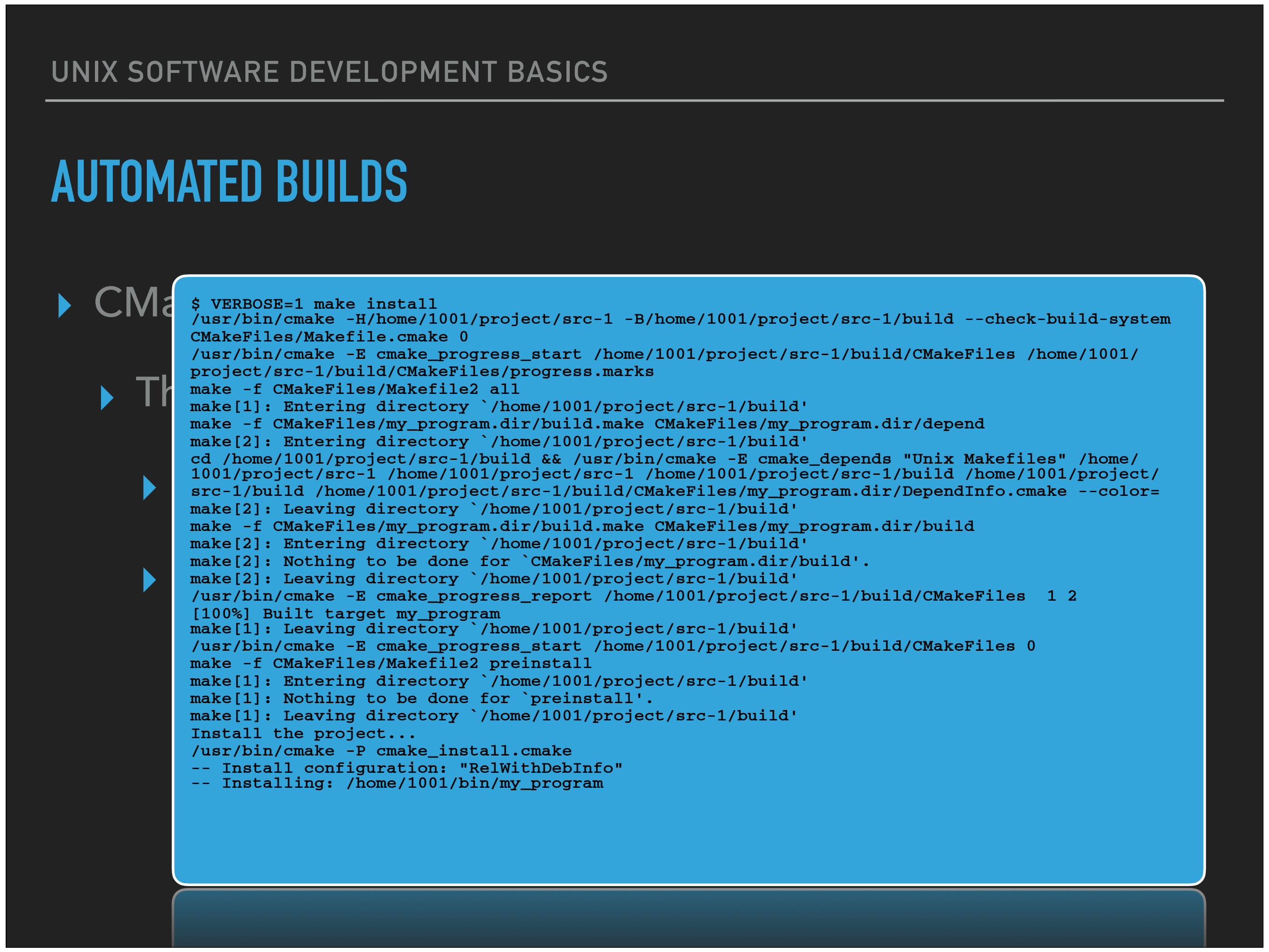


AUTOMATED BUILDS

- CMa
  - Th

```
$ VERBOSE=1 make
/usr/bin/cmake -H/home/1001/project/src-1 -B/home/1001/project/src-1/build --check-build-system
CMakeFiles/Makefile.cmake 0
/usr/bin/cmake -E cmake_progress_start /home/1001/project/src-1/build/CMakeFiles /home/1001/
project/src-1/build/CMakeFiles/progress.marks
make -f CMakeFiles/Makefile2 all
make[1]: Entering directory `/home/1001/project/src-1/build'
make -f CMakeFiles/my_program.dir/build.make CMakeFiles/my_program.dir/depend
make[2]: Entering directory `/home/1001/project/src-1/build'
cd /home/1001/project/src-1/build && /usr/bin/cmake -E cmake_depends "Unix Makefiles" /home/
1001/project/src-1 /home/1001/project/src-1 /home/1001/project/src-1/build /home/1001/project/
src-1/build /home/1001/project/src-1/build/CMakeFiles/my_program.dir/DependInfo.cmake --color=
make[2]: Leaving directory `/home/1001/project/src-1/build'
make -f CMakeFiles/my_program.dir/build.make CMakeFiles/my_program.dir/build
make[2]: Entering directory `/home/1001/project/src-1/build'
/usr/bin/cmake -E cmake_progress_report /home/1001/project/src-1/build/CMakeFiles 1
[ 50%] Building C object CMakeFiles/my_program.dir/my_program.c.o
/usr/lib64/ccache/cc   -O2 -g -DNDEBUG   -o CMakeFiles/my_program.dir/my_program.c.o   -c /home/
1001/project/src-1/my_program.c
/usr/bin/cmake -E cmake_progress_report /home/1001/project/src-1/build/CMakeFiles 2
[100%] Building C object CMakeFiles/my_program.dir/printargv.c.o
/usr/lib64/ccache/cc   -O2 -g -DNDEBUG   -o CMakeFiles/my_program.dir/printargv.c.o   -c /home/
1001/project/src-1/printargv.c
Linking C executable my_program
/usr/bin/cmake -E cmake_link_script CMakeFiles/my_program.dir/link.txt --verbose=1
/usr/lib64/ccache/cc  -O2 -g -DNDEBUG    CMakeFiles/my_program.dir/my_program.c.o CMakeFiles/
my_program.dir/printargv.c.o  -o my_program -rdynamic
make[2]: Leaving directory `/home/1001/project/src-1/build'
/usr/bin/cmake -E cmake_progress_report /home/1001/project/src-1/build/CMakeFiles  1 2
[100%] Built target my_program
make[1]: Leaving directory `/home/1001/project/src-1/build'
/usr/bin/cmake -E cmake_progress_start /home/1001/project/src-1/build/CMakeFiles 0
```

- Verbose build can be used to see the commands being issued by make

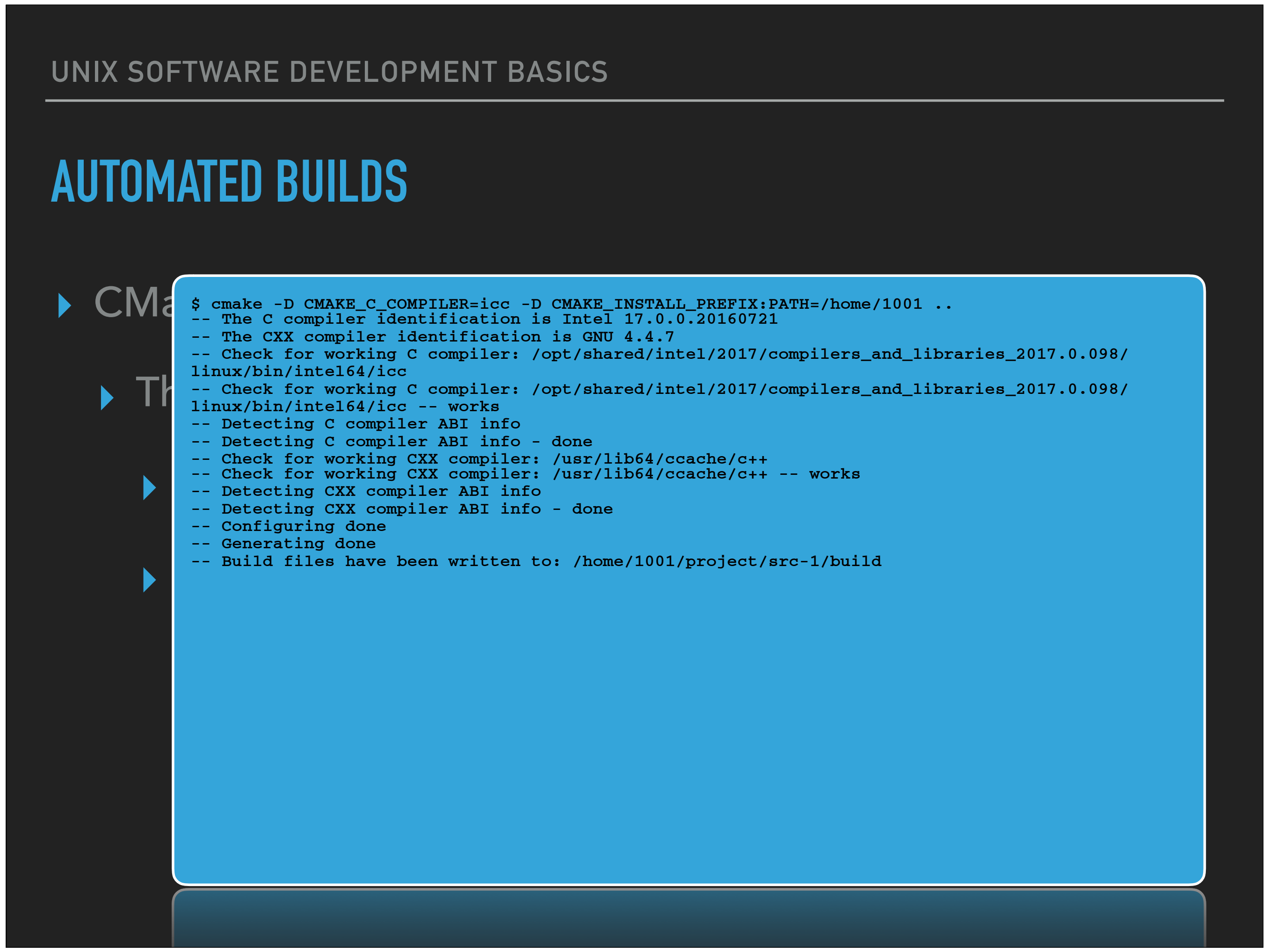


AUTOMATED BUILDS

- CMa
  - Th

```
$ VERBOSE=1 make install
/usr/bin/cmake -H/home/1001/project/src-1 -B/home/1001/project/src-1/build --check-build-system
CMakeFiles/Makefile.cmake 0
/usr/bin/cmake -E cmake_progress_start /home/1001/project/src-1/build/CMakeFiles /home/1001/
project/src-1/build/CMakeFiles/progress.marks
make -f CMakeFiles/Makefile2 all
make[1]: Entering directory `/home/1001/project/src-1/build'
make -f CMakeFiles/my_program.dir/build.make CMakeFiles/my_program.dir/depend
make[2]: Entering directory `/home/1001/project/src-1/build'
cd /home/1001/project/src-1/build && /usr/bin/cmake -E cmake_depends "Unix Makefiles" /home/
1001/project/src-1 /home/1001/project/src-1 /home/1001/project/src-1/build /home/1001/project/
src-1/build /home/1001/project/src-1/build/CMakeFiles/my_program.dir/DependInfo.cmake --color=
make[2]: Leaving directory `/home/1001/project/src-1/build'
make -f CMakeFiles/my_program.dir/build.make CMakeFiles/my_program.dir/build
make[2]: Entering directory `/home/1001/project/src-1/build'
make[2]: Nothing to be done for `CMakeFiles/my_program.dir/build'.
make[2]: Leaving directory `/home/1001/project/src-1/build'
/usr/bin/cmake -E cmake_progress_report /home/1001/project/src-1/build/CMakeFiles  1 2
[100%] Built target my_program
make[1]: Leaving directory `/home/1001/project/src-1/build'
/usr/bin/cmake -E cmake_progress_start /home/1001/project/src-1/build/CMakeFiles 0
make -f CMakeFiles/Makefile2 preinstall
make[1]: Entering directory `/home/1001/project/src-1/build'
make[1]: Nothing to be done for `preinstall'.
make[1]: Leaving directory `/home/1001/project/src-1/build'
Install the project...
/usr/bin/cmake -P cmake_install.cmake
-- Install configuration: "RelWithDebInfo"
-- Installing: /home/1001/bin/my_program
```

- Each invocation of "make" appears to be more complicated than the autotools example…

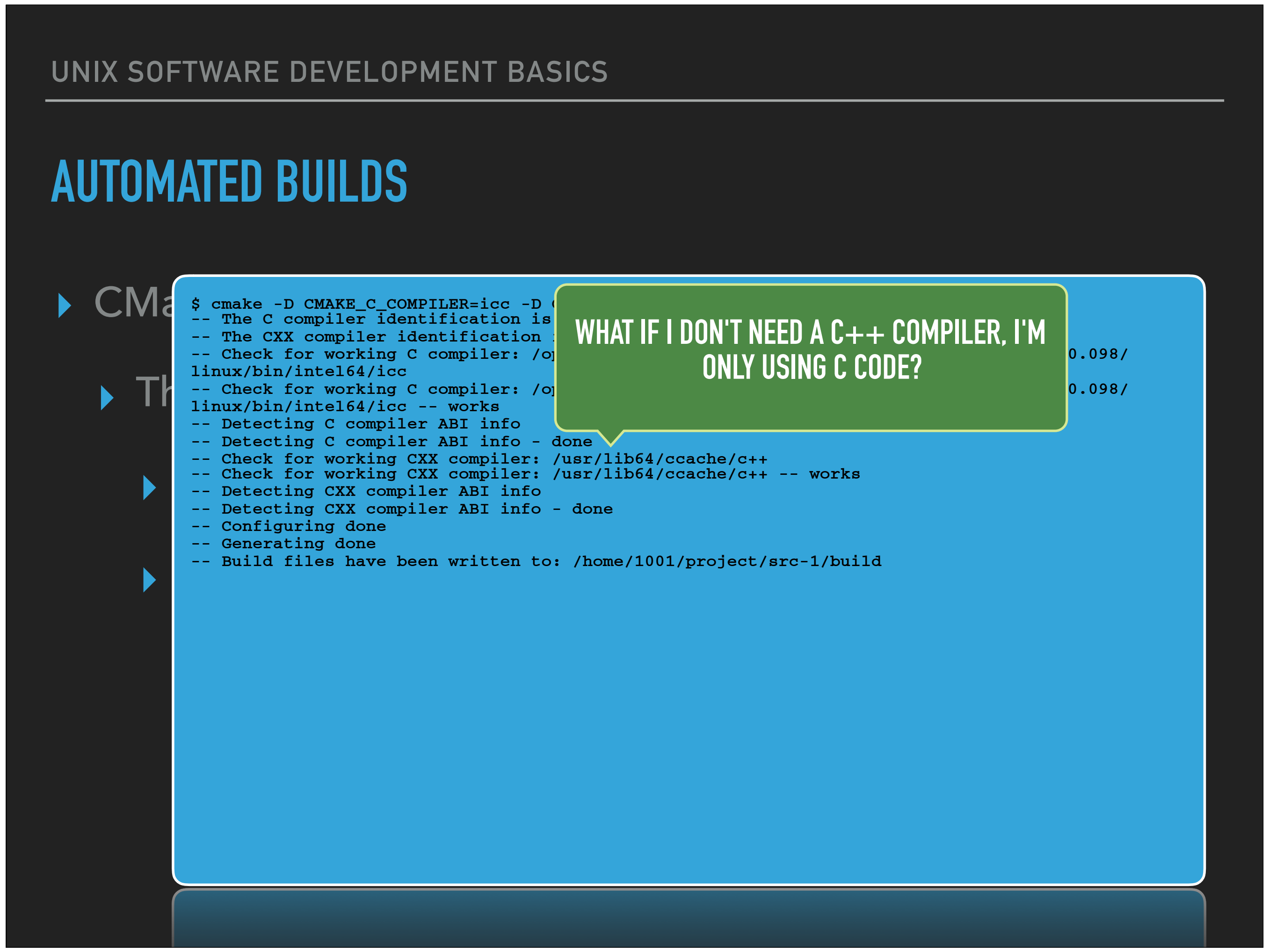


AUTOMATED BUILDS

- CMa
  - Th

```
$ cmake -D CMAKE_C_COMPILER=icc -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
-- The C compiler identification is Intel 17.0.0.20160721
-- The CXX compiler identification is GNU 4.4.7
-- Check for working C compiler: /opt/shared/intel/2017/compilers_and_libraries_2017.0.098/
linux/bin/intel64/icc
-- Check for working C compiler: /opt/shared/intel/2017/compilers_and_libraries_2017.0.098/
linux/bin/intel64/icc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/lib64/ccache/c++
-- Check for working CXX compiler: /usr/lib64/ccache/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/1001/project/src-1/build
```

- Source type can be limited in the project() call in CMakeLists.txt



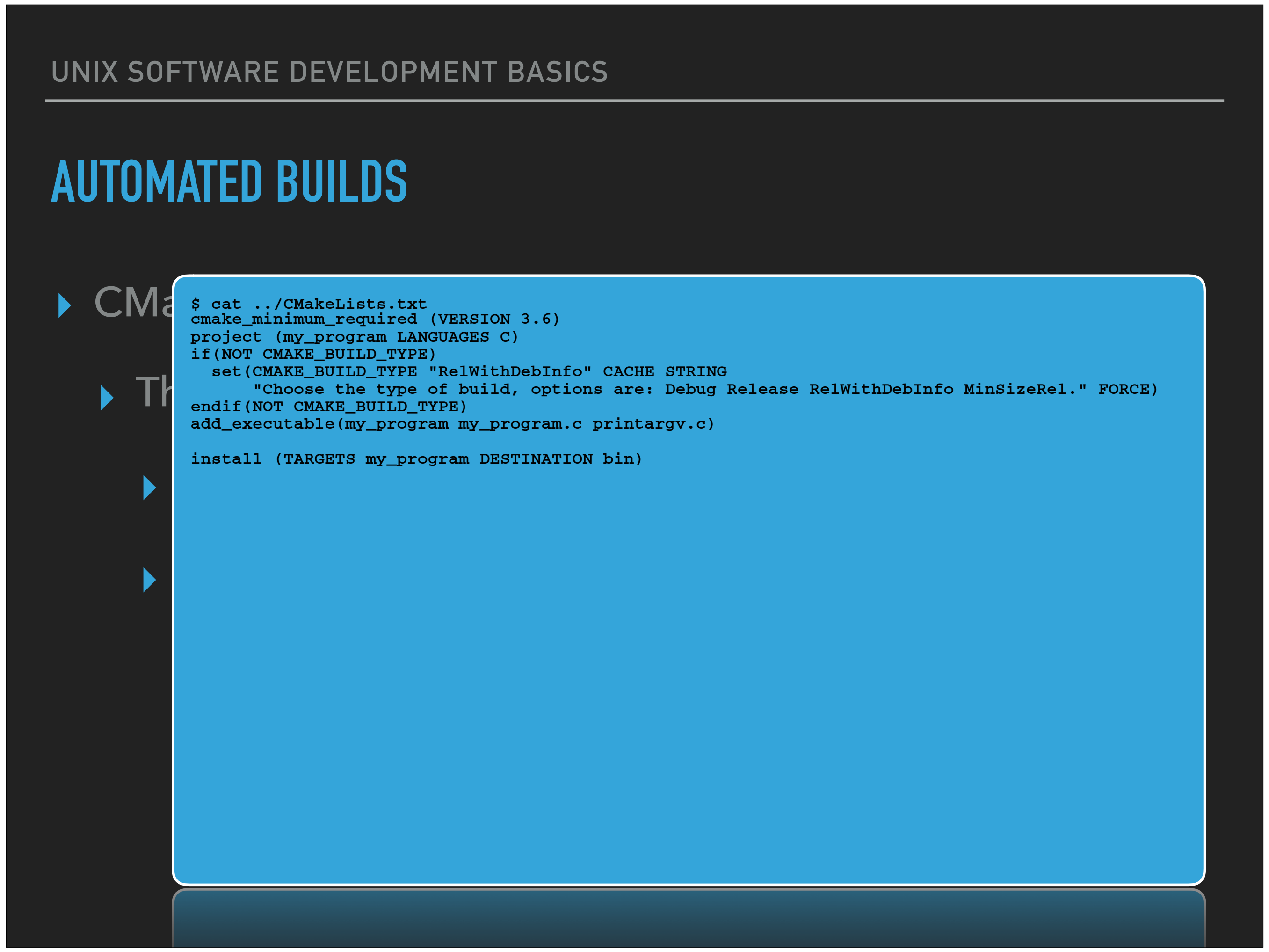
AUTOMATED BUILDS

- CMa
  - Th

```
$ cmake -D CMAKE_C_COMPILER=icc -D
-- The C compiler identification is
-- The CXX compiler identification
-- Check for working C compiler: /o                                    0.098/
linux/bin/intel64/icc
-- Check for working C compiler: /o                                    0.098/
linux/bin/intel64/icc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/lib64/ccache/c++
-- Check for working CXX compiler: /usr/lib64/ccache/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/1001/project/src-1/build
```

WHAT IF I DON'T NEED A C++ COMPILER, I'M ONLY USING C CODE?

- Source type can be limited in the project() call in CMakeLists.txt

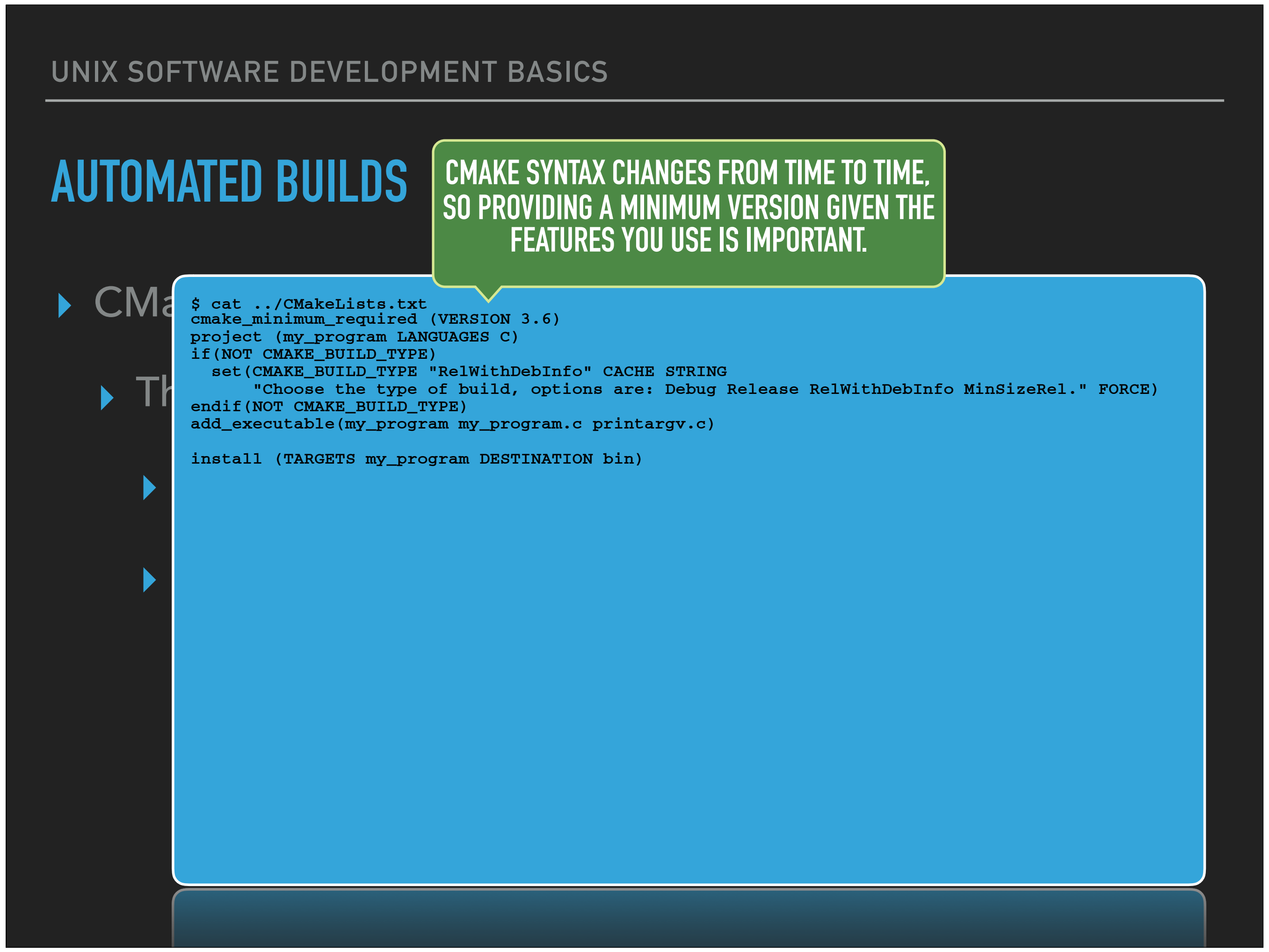


AUTOMATED BUILDS

- CMa
  - Th

```
$ cat ../CMakeLists.txt
cmake_minimum_required (VERSION 3.6)
project (my_program LANGUAGES C)
if(NOT CMAKE_BUILD_TYPE)
  set(CMAKE_BUILD_TYPE "RelWithDebInfo" CACHE STRING
      "Choose the type of build, options are: Debug Release RelWithDebInfo MinSizeRel." FORCE)
endif(NOT CMAKE_BUILD_TYPE)
add_executable(my_program my_program.c printargv.c)

install (TARGETS my_program DESTINATION bin)
```

- Source type can be limited in the project() call in CMakeLists.txt
- Also add a "default build type" if none was provided



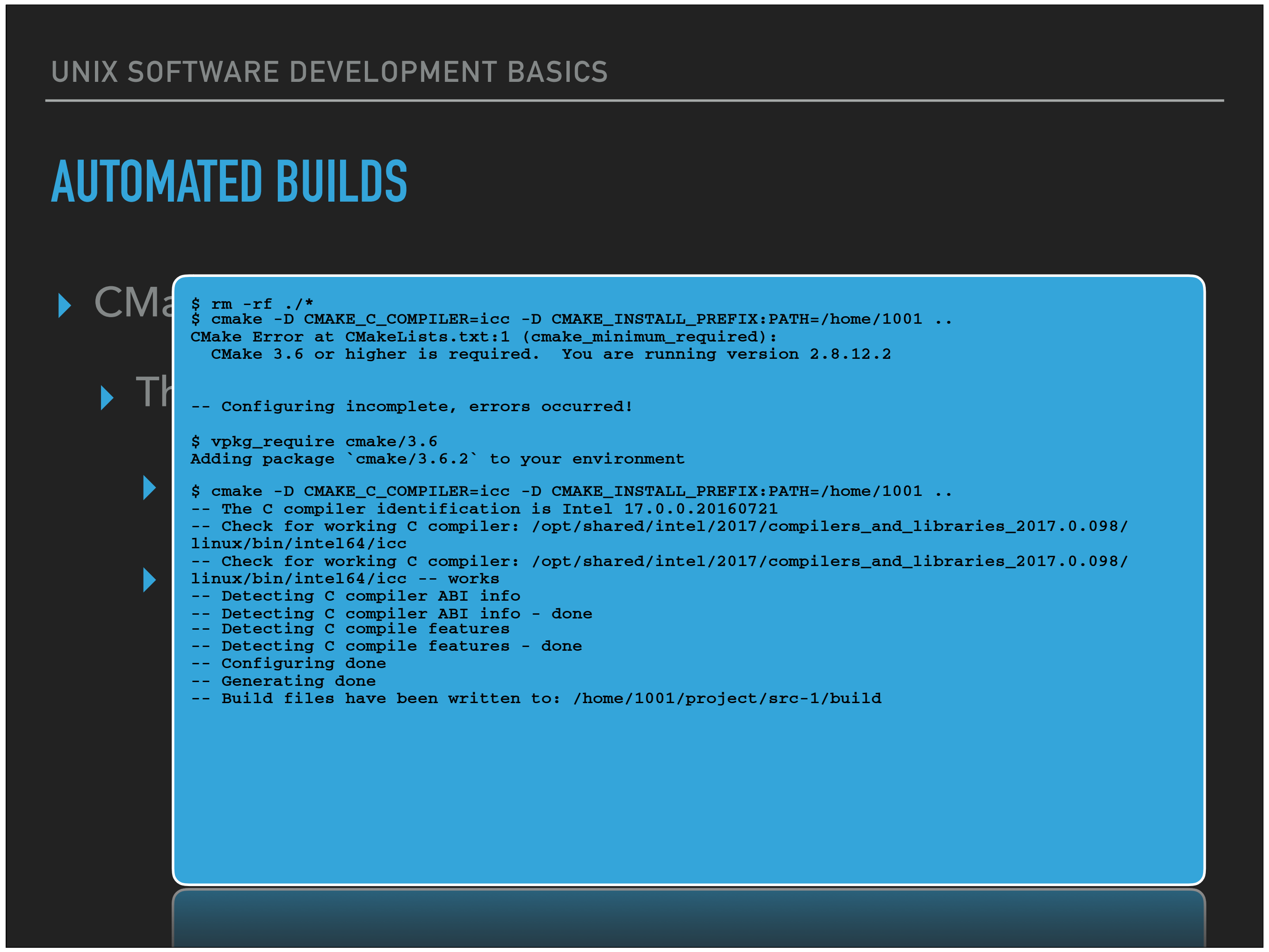
AUTOMATED BUILDS

CMAKE SYNTAX CHANGES FROM TIME TO TIME, SO PROVIDING A MINIMUM VERSION GIVEN THE FEATURES YOU USE IS IMPORTANT.

- CMa
  - Th

```
$ cat ../CMakeLists.txt
cmake_minimum_required (VERSION 3.6)
project (my_program LANGUAGES C)
if(NOT CMAKE_BUILD_TYPE)
  set(CMAKE_BUILD_TYPE "RelWithDebInfo" CACHE STRING
      "Choose the type of build, options are: Debug Release RelWithDebInfo MinSizeRel." FORCE)
endif(NOT CMAKE_BUILD_TYPE)
add_executable(my_program my_program.c printargv.c)

install (TARGETS my_program DESTINATION bin)
```

- Source type can be limited in the project() call in CMakeLists.txt
- Also add a "default build type" if none was provided

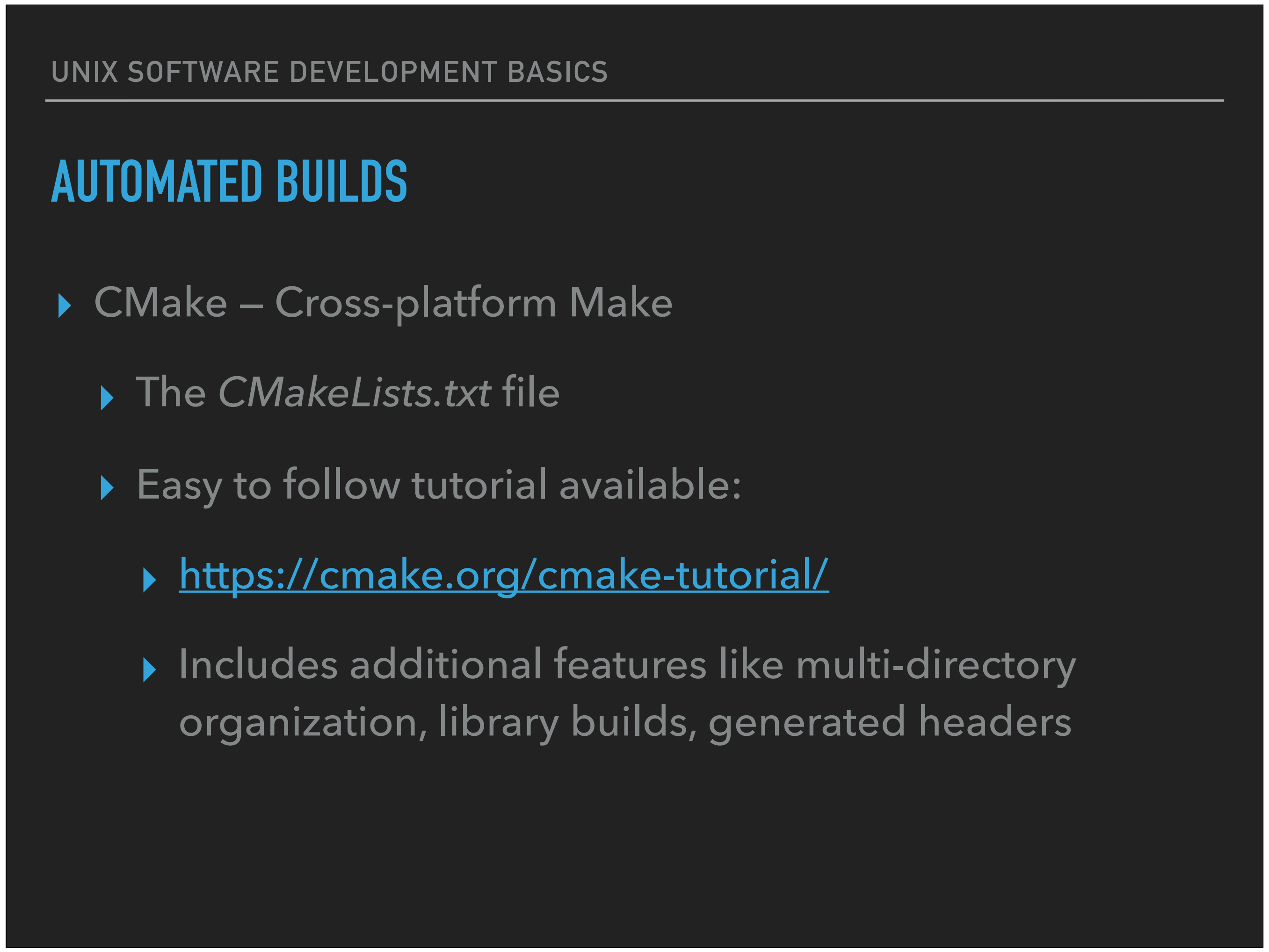

UNIX SOFTWARE DEVELOPMENT BASICS

AUTOMATED BUILDS

- CMa
  - Th

```
$ rm -rf ./*
$ cmake -D CMAKE_C_COMPILER=icc -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
CMake Error at CMakeLists.txt:1 (cmake_minimum_required):
  CMake 3.6 or higher is required.  You are running version 2.8.12.2


-- Configuring incomplete, errors occurred!

$ vpkg_require cmake/3.6
Adding package `cmake/3.6.2` to your environment

$ cmake -D CMAKE_C_COMPILER=icc -D CMAKE_INSTALL_PREFIX:PATH=/home/1001 ..
-- The C compiler identification is Intel 17.0.0.20160721
-- Check for working C compiler: /opt/shared/intel/2017/compilers_and_libraries_2017.0.098/
linux/bin/intel64/icc
-- Check for working C compiler: /opt/shared/intel/2017/compilers_and_libraries_2017.0.098/
linux/bin/intel64/icc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/1001/project/src-1/build
```

- On Farber, additional (newer) versions of CMake are available via VALET



AUTOMATED BUILDS

- CMake – Cross-platform Make
  - The *CMakeLists.txt* file
  - Easy to follow tutorial available:
    - https://cmake.org/cmake-tutorial/
    - Includes additional features like multi-directory organization, library builds, generated headers

- Documentation present online, also via man pages

FOR EXAMPLES, SEE

```
src-4/
src-5/
```