

Build Systems

Goals

- portable code
- reproducible code
- optimize compile times

program structure

- your userland code
- shared libraries
- operating system libraries
- kernel and system functions
- hardware

Popular Systems

- Makefiles / Autotools
- cmake
- Ant (java)
- qmake (Qt)
- Scons (python)

autotools

Developer:

1. Create the sources, Makefile, etc.
2. Run “autoscan” (“autoscan” creates “configure.scan”)
3. Rename “configure.scan” to “configure.ac”
4. Run “autoconf”
(“autoconf” uses “configure.ac” to create “configure”)
5. Rename “Makefile” to “Makefile.in”
(“configure” will use “Makefile.in” to create “Makefile”)
6. Run “./configure” and “make”

Build Steps

User:

- download source tar ball
- unpack
- ./configure
- make
- make install

Makefile

- simple text file
- can use variables
- target: dependencies
[Tab] files.c

```
# I am a comment, and I want to say that the variable CC will be
# the compiler to use.
CC=g++
# Hey!, I am comment number 2. I want to say that CFLAGS will be the
# options I'll pass to the compiler.
CFLAGS=-c -Wall
```

```
all: hello
```

```
hello: main.o factorial.o hello.o
    $(CC) main.o factorial.o hello.o -o hello
```

```
main.o: main.cpp
    $(CC) $(CFLAGS) main.cpp
```

```
factorial.o: factorial.cpp
    $(CC) $(CFLAGS) factorial.cpp
```

```
hello.o: hello.cpp
    $(CC) $(CFLAGS) hello.cpp
```

```
clean:
    rm *o hello
```

CMake

- Works on Unix, Windows, Mac OS X
- Relies only on C++ compiler
- Can create installers
- Can create IDE specific project definitions for Eclipse, Visual Studio, Xcode

Cmake Example

```
cmake_minimum_required (VERSION 2.6)
project (Tutorial)
# The version number.
set (Tutorial_VERSION_MAJOR 1)
set (Tutorial_VERSION_MINOR 0)

# configure a header file to pass some of the CMake settings
# to the source code
configure_file (
    "${PROJECT_SOURCE_DIR}/TutorialConfig.h.in"
    "${PROJECT_BINARY_DIR}/TutorialConfig.h"
)

# add the binary tree to the search path for include files
# so that we will find TutorialConfig.h
include_directories("${PROJECT_BINARY_DIR}")

# add the executable
add_executable(Tutorial tutorial.cxx)
```

CTest CDash

- Provides tools for Test Driven Development (Unit tests)
- Provides web interface for builds and test

CDash Example

[Login](#) [All Dashboards](#)

Wednesday, November 11 2015 09:48:26 UTC

 **ROOT**

[Dashboard](#) [Calendar](#) [Previous](#) [Current](#) [Project](#)

No file changed as of **Tuesday, November 10 2015 - 22:00 UTC**

48 minutes ago: 2 tests failed on master-x86_64-centos7-gcc48-dbg
1 hours ago: 1 warning introduced on master-x86_64-centos7-gcc48-dbg
1 hours ago: 17 tests failed on master-x86_64-centos7-gcc48-opt
1 hours ago: 1 warning introduced on master-x86_64-centos7-gcc48-opt
1 hours ago: 17 tests failed on v6-04-00-patches-x86_64-centos7-gcc48-opt

[See full feed](#)

Nightly

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
lcgapp-cc7-x86-64-6.cern.ch	v6-02-00-patches-x86_64-centos7-gcc49-opt-slc7	0	0	0	0	0	0	172 ⁺¹⁶⁷	724 ⁻¹⁶⁷	10 hours ago
moonshot-arm64-04	master-aarch64-ubuntu14-gcc49-dbg	47	0	0	0	1	0	21	1203	9 hours ago
moonshot-arm64-04	master-aarch64-ubuntu14-gcc49-fst	47	0	0	0	2	0	18	1206	9 hours ago
moonshot-arm64-04	master-aarch64-ubuntu14-gcc49-opt	47	0	0	0	2	0	15	1209	9 hours ago
lcgapp-slc6-x86-64-22.cern.ch	master-x86_64-slc6-gcc49-fst	100	0	0	0	3 ⁺¹ ₋₁	0	12 ⁺¹¹ ₋₁	1287 ⁻¹¹	7 hours ago
lcgapp-slc6-x86-64-9.cern.ch	master-x86_64-slc6-clang36-dbg	84	0	0	0	1 ⁺¹	0	12 ⁺¹¹ ₋₁	1234 ⁺¹⁷ ₋₁₁	8 hours ago
lcgapp-slc6-x86-64-6.cern.ch	master-x86_64-slc6-gcc49-opt	84	0	0	0	19 ⁺¹	0	11 ⁺¹¹ ₋₁	1294 ⁺² ₋₁₁	8 hours ago
lcgapp-slc6-x86-64-22.cern.ch	master-x86_64-slc6-gcc48-dbg	47	0	0	0	18 ⁺¹	0	11 ⁺¹¹	1294 ⁻¹¹	8 hours ago
macitois14.cern.ch	v6-04-00-patches-x86_64-mac1010-clang70-opt	0	0	0	0	8	0	11	1139	9 hours ago
sft-fedora-1.cern.ch	master-x86_64-fedora21-gcc49-opt	47	0	0	0	2	0	11	1262	9 hours ago

<http://cdash.cern.ch/index.php?project=ROOT>