Compiling Multi-Million Line C++ Code Bases Effortlessly with the Meson Build System

Jussi Pakkanen
@jpakkane
Rakettitiede oy

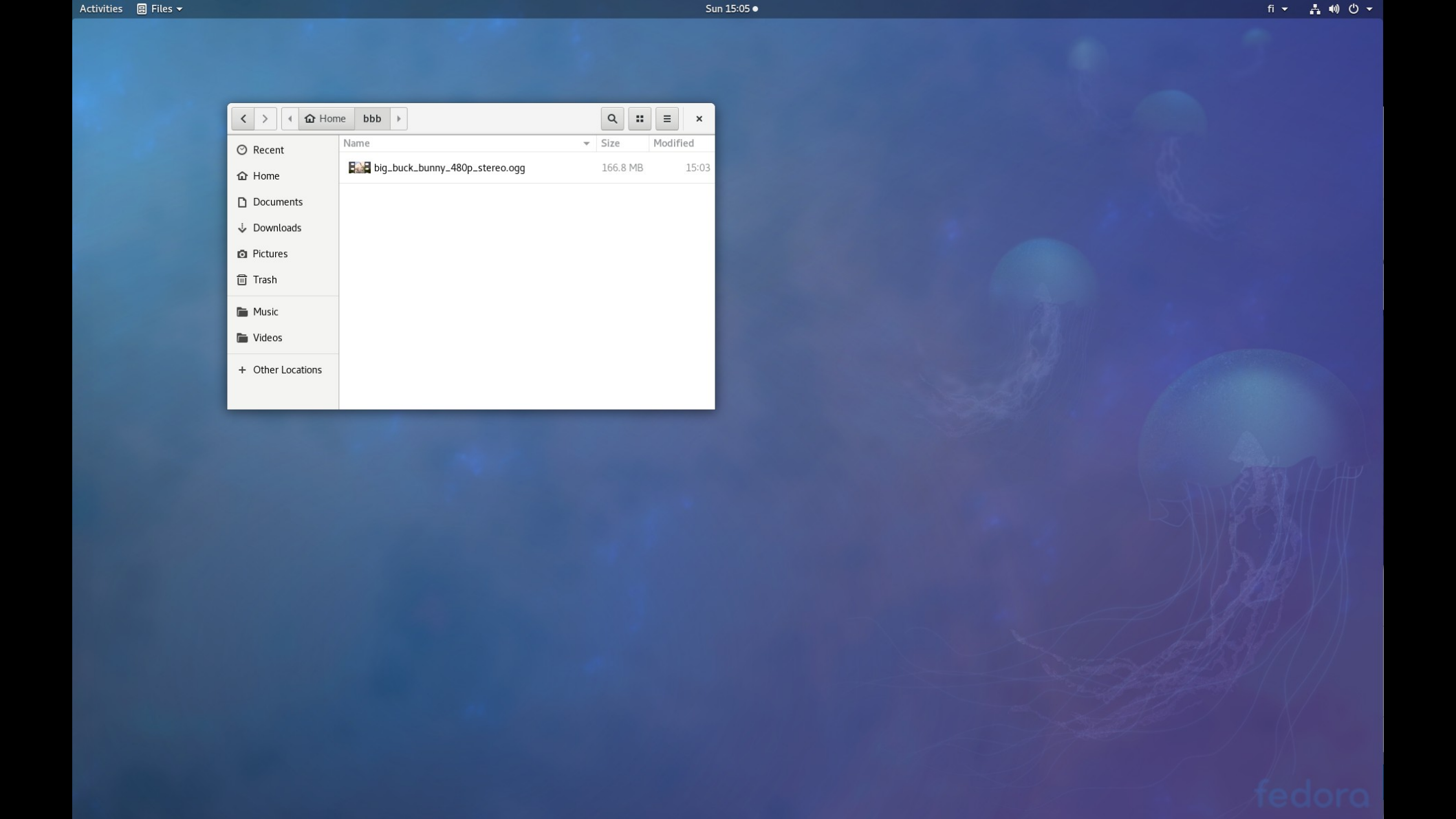
https://meson.build

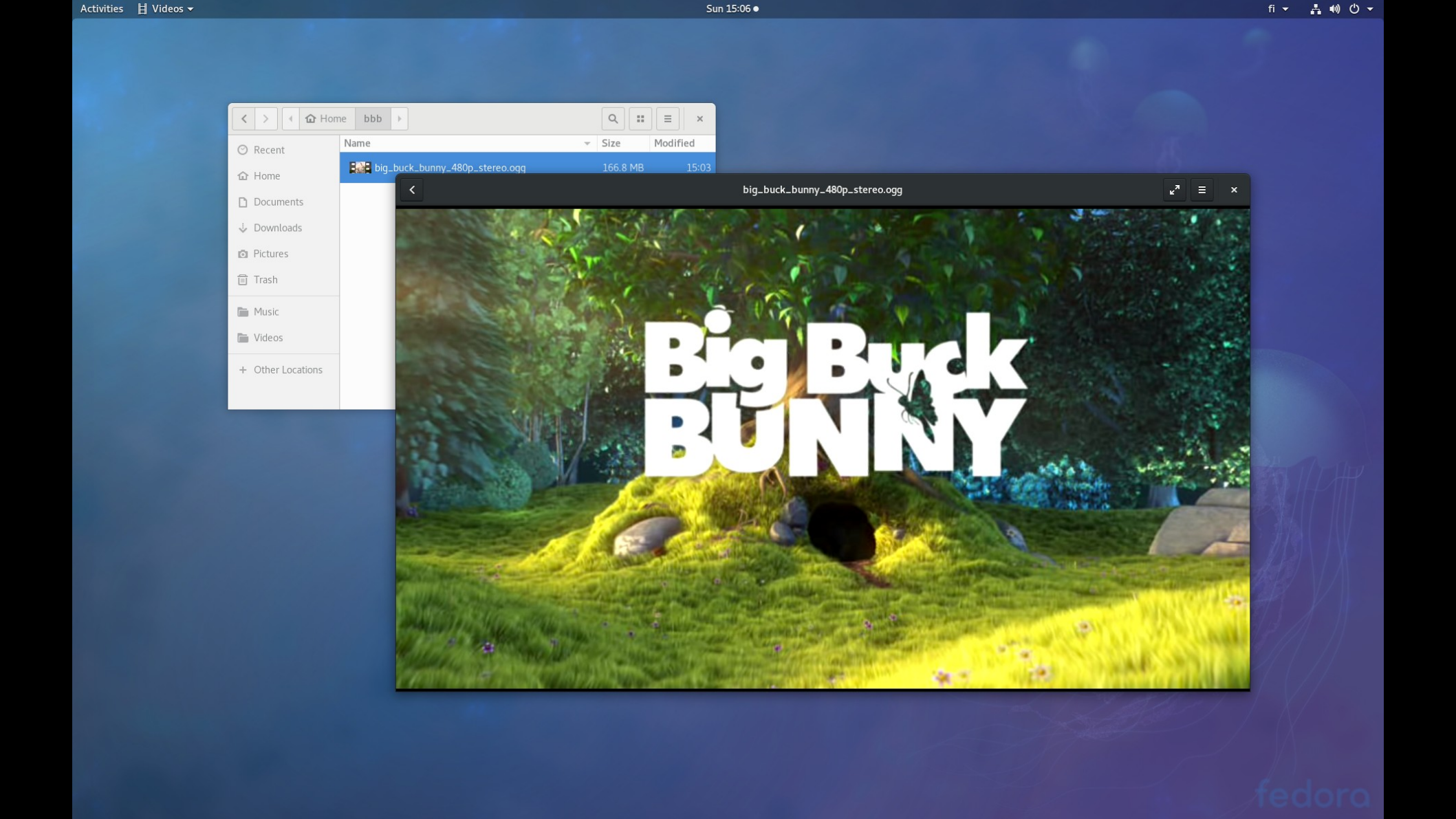
What is it?

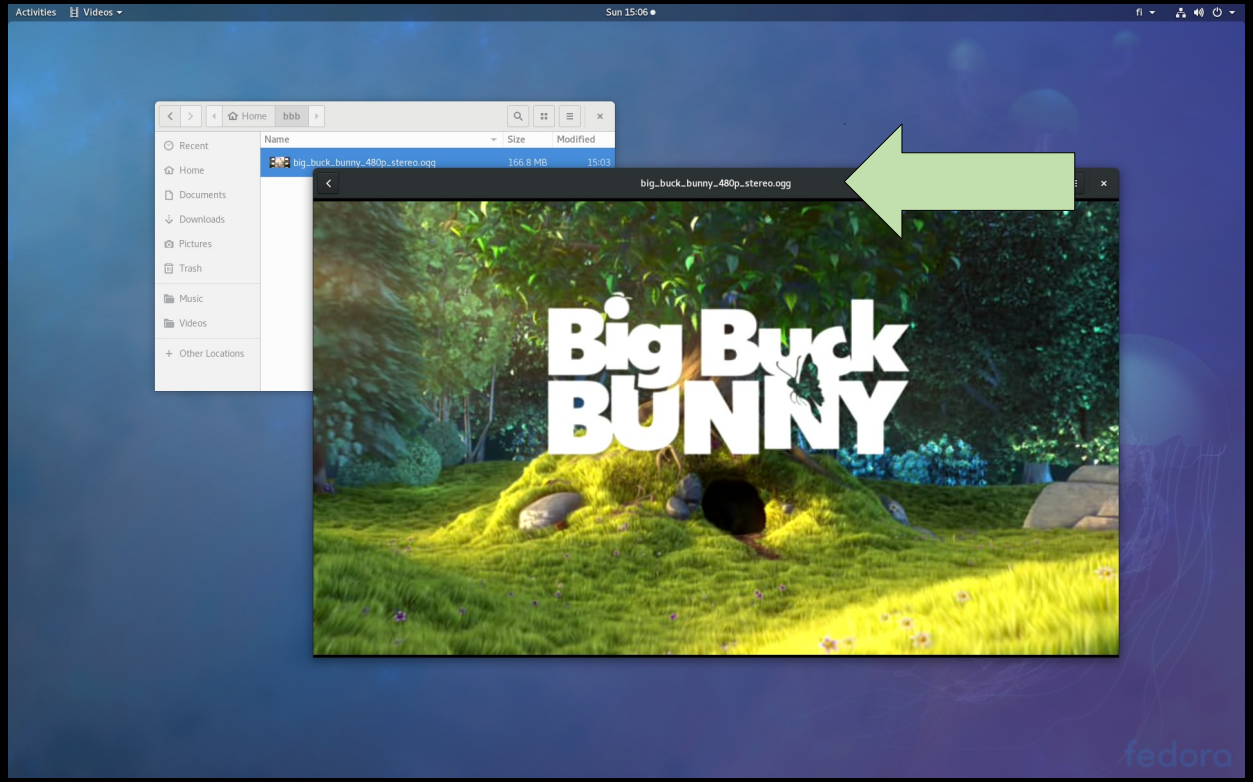
- A build system for multiple languages
- Optimized for modern OSs and toolchains, not HP-UX from 1989
- Fast, lean, efficient
- Scales to tens of thousands of source files
- Minimize the time developers have to interact with the build system
- Build definitions are simple, readable and not Turing complete

PART ONE

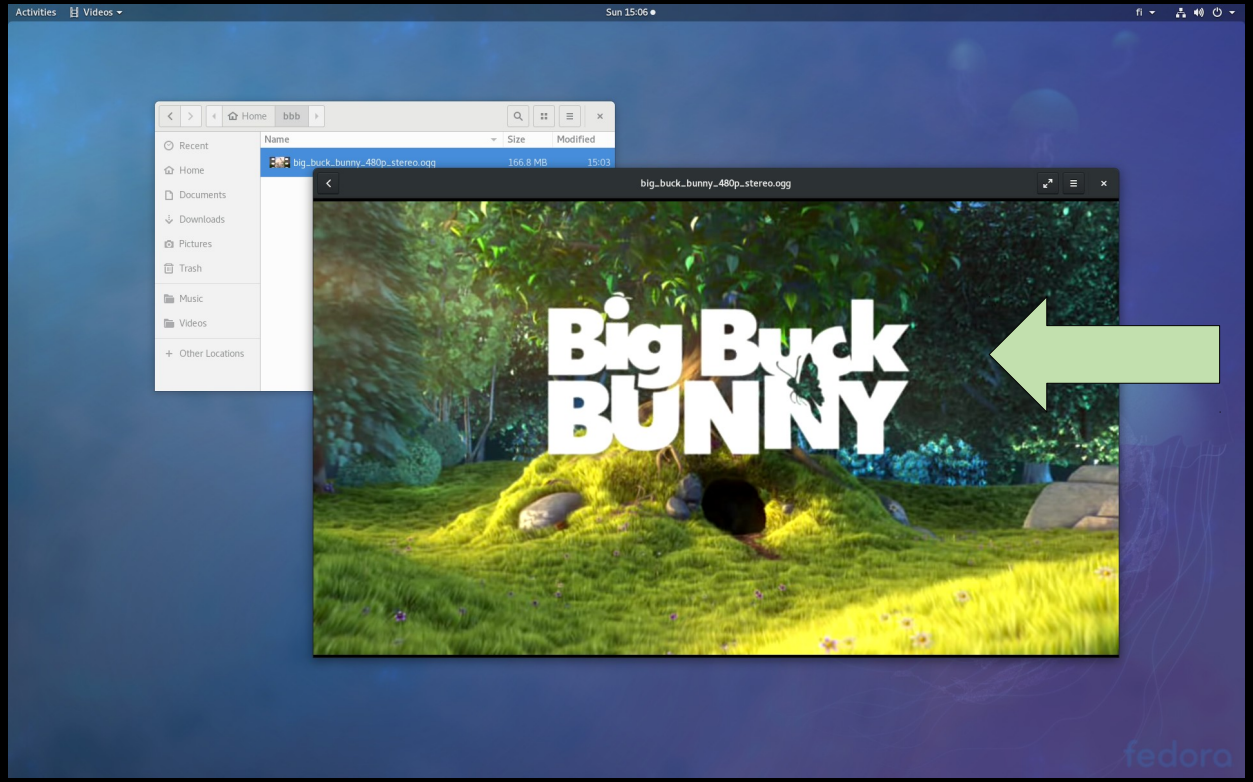
In which we ask whether
anyone is actually using it?


Activities
Files
Sun 15:05
fi
Home
bbb
Recent
Home
Documents
Downloads
Pictures
Trash
Music
Videos
Other Locations
Name
Size
Modified
big_buck_bunny_480p_stereo.ogg
166.8 MB
15:03
fedora

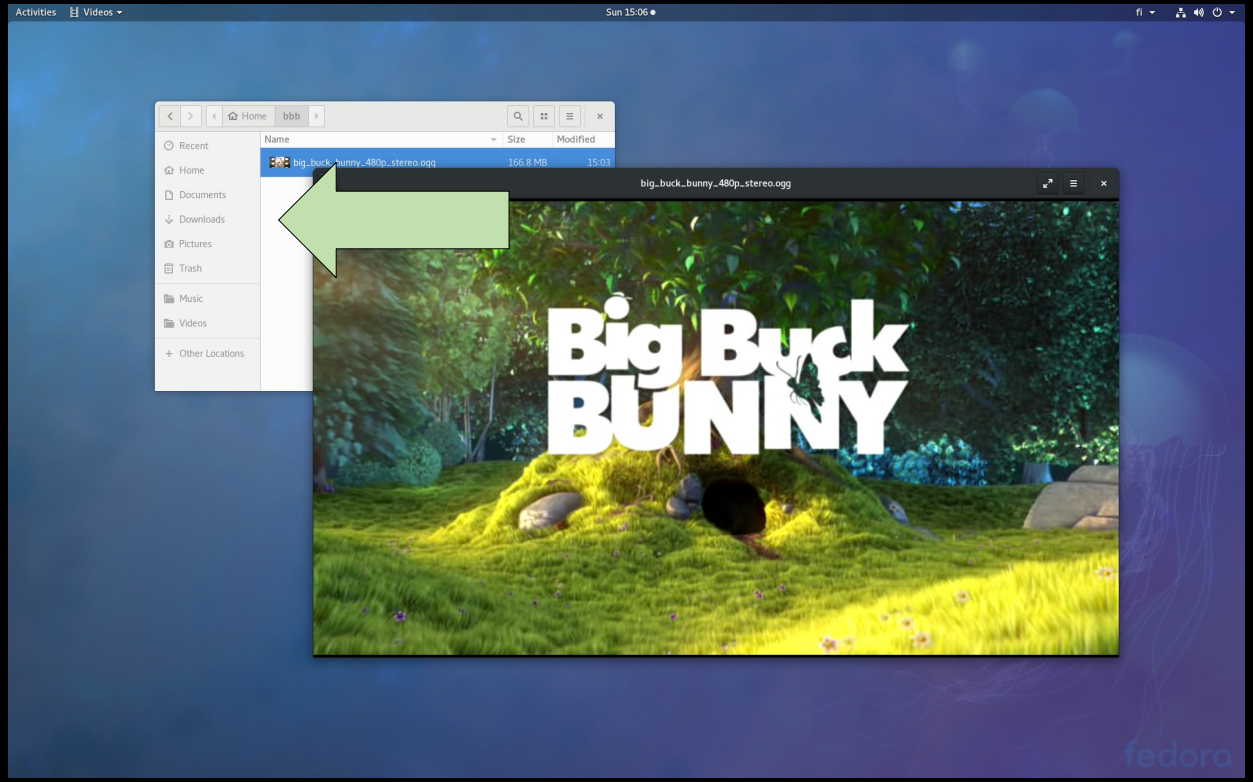

Activities
Videos
Sun 15:06
fi
Home
bbb
Recent
Home
Documents
Downloads
Pictures
Trash
Music
Videos
Other Locations
Name
Size
Modified
big_buck_bunny_480p_stereo.ogg
166.8 MB
15:03
big_buck_bunny_480p_stereo.ogg
Big Buck
BUNNY
fedora



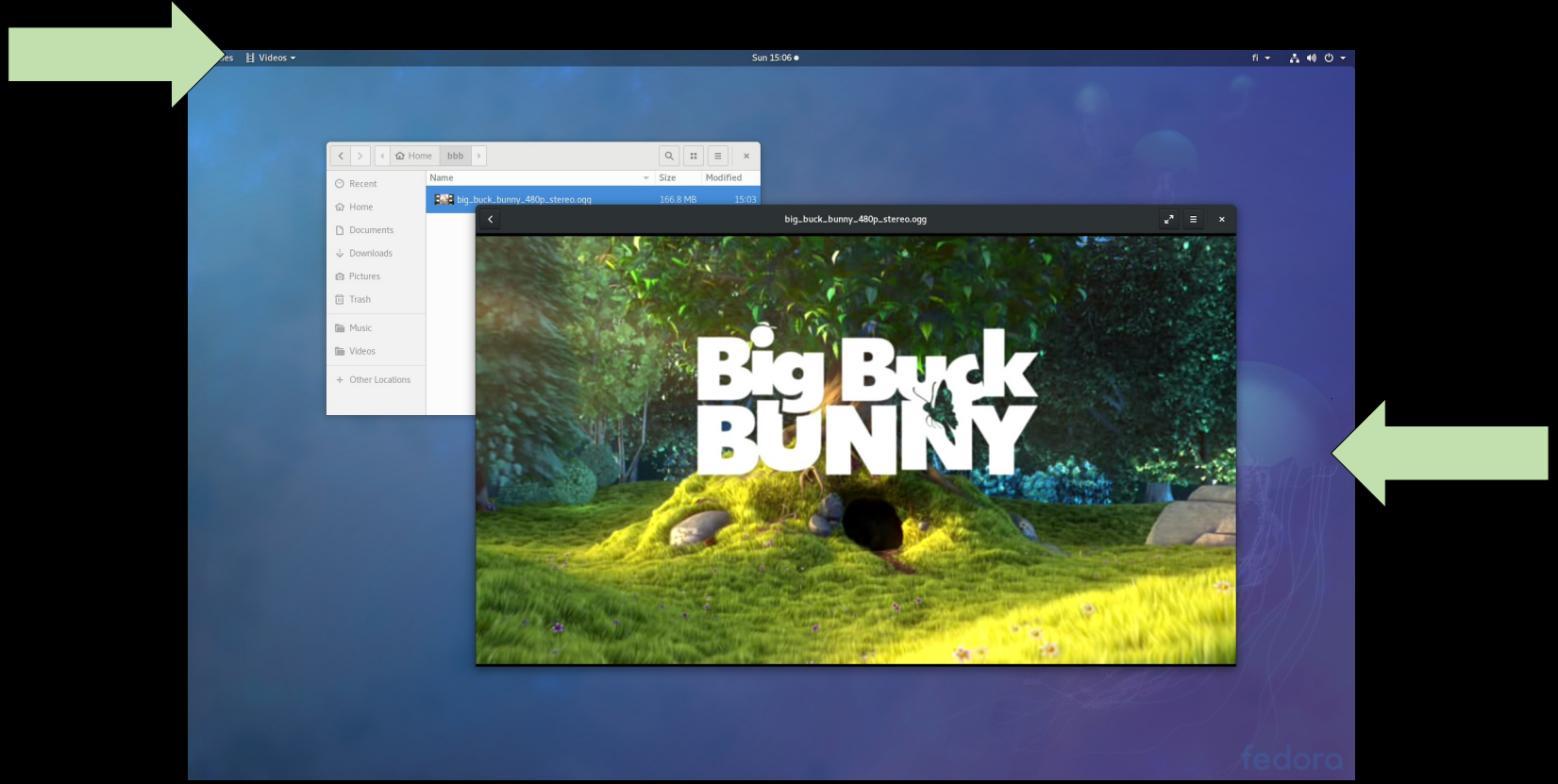
GNOME Videos



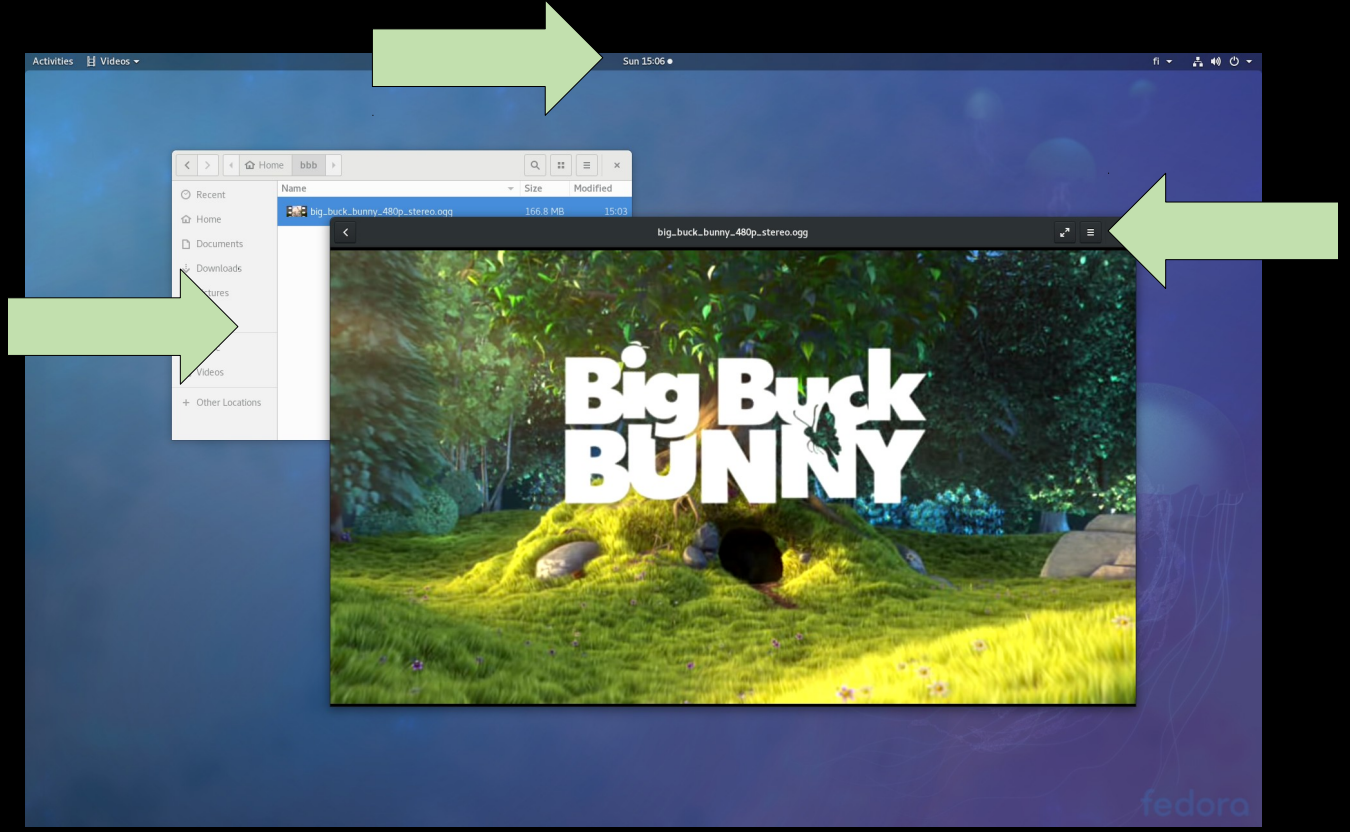
GStreamer multimedia framework



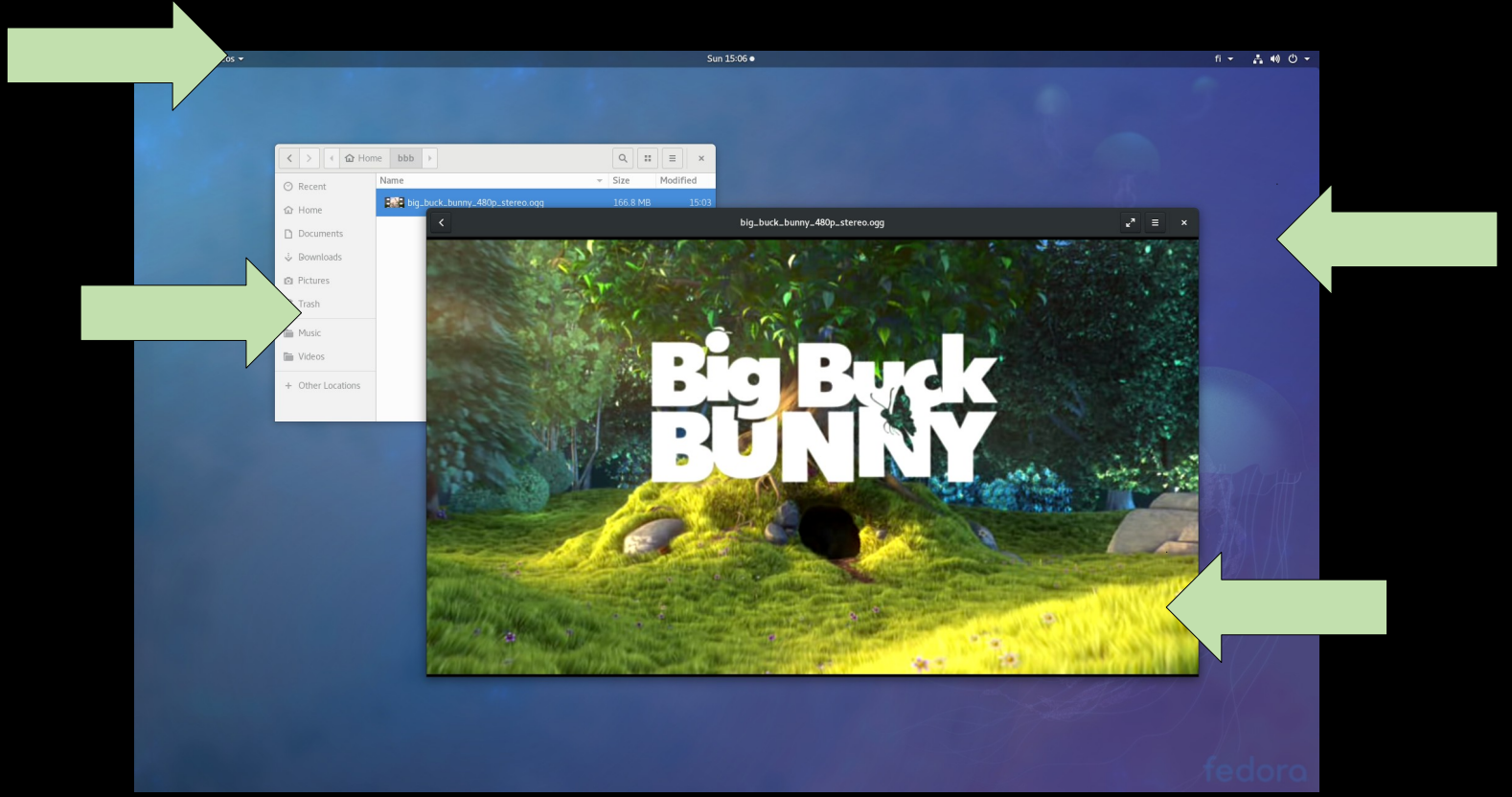
Nautilus file manager


Videos
Sun 15:06
fi
Home
bbb
Recent
Home
Documents
Downloads
Pictures
Trash
Music
Videos
Other Locations
Name
Size
Modified
big_buck_bunny_480p_stereo.ogg
166.8 MB
15:03
big_buck_bunny_480p_stereo.ogg
Big Buck
BUNNY
fedora

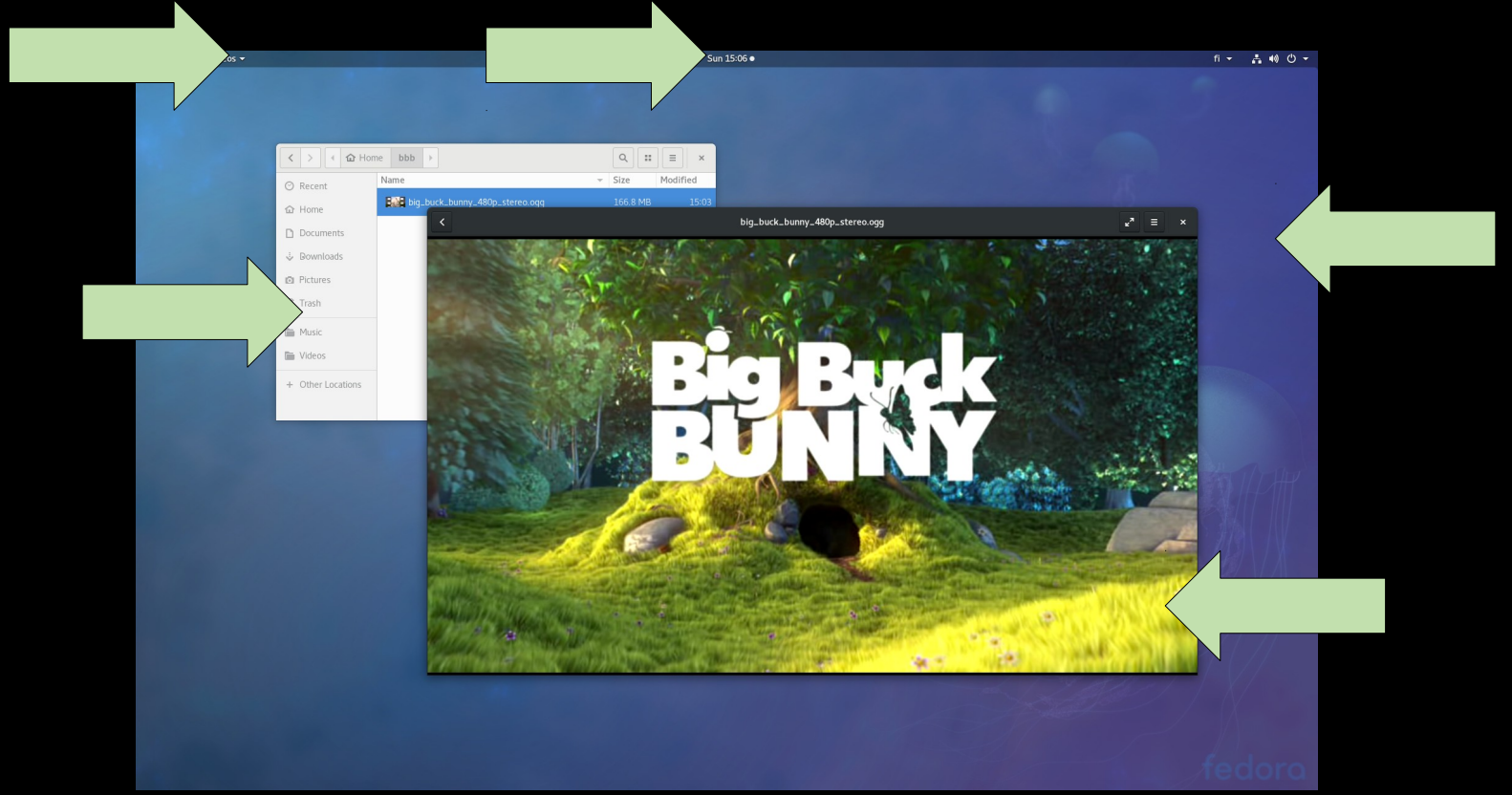
GNOME Shell



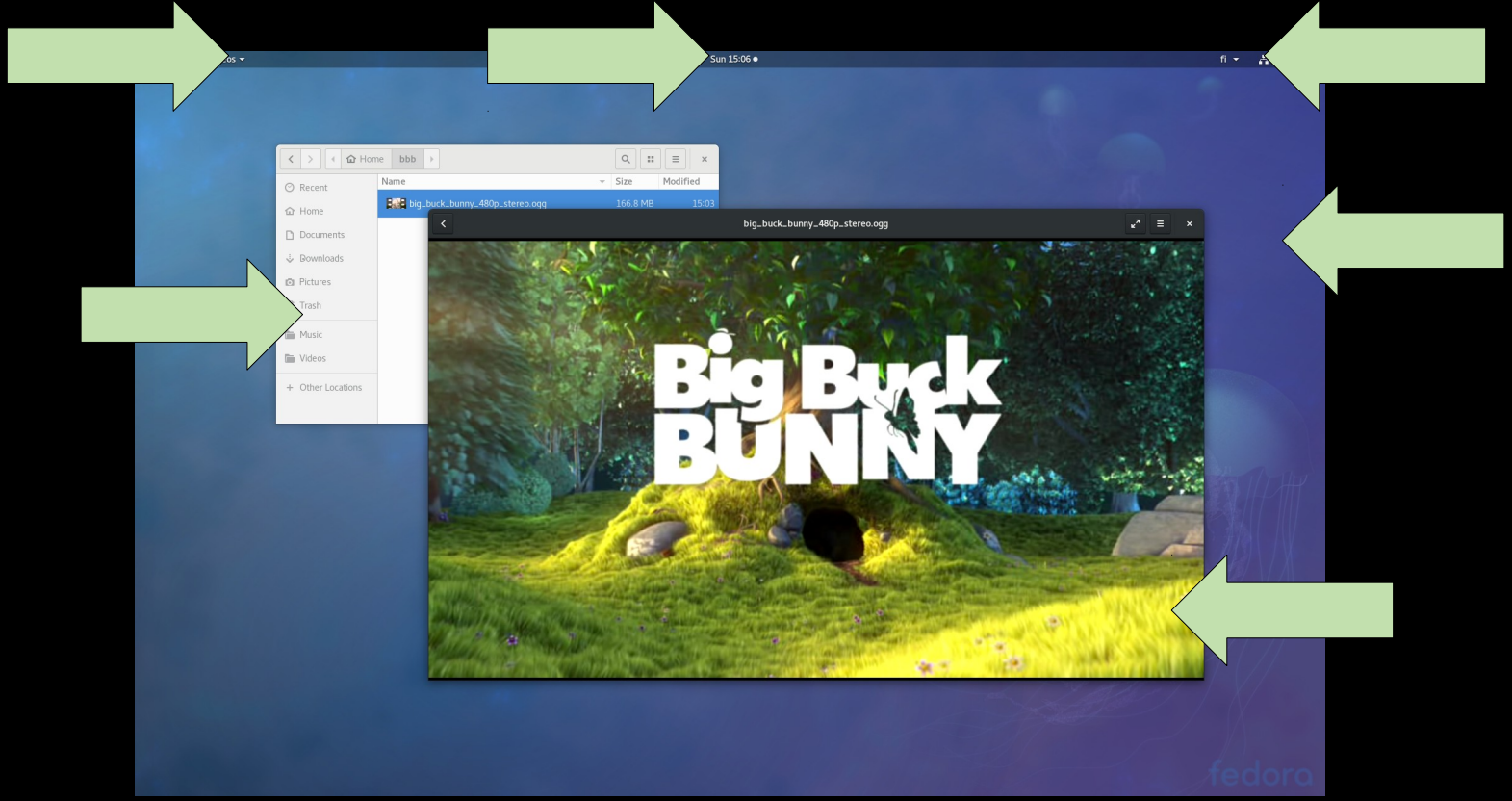
GTK widget toolkit


Sun 15:06
fi
Home
bbb
Recent
Home
Documents
Downloads
Pictures
Trash
Music
Videos
Other Locations
Name
Size
Modified
big_buck_bunny_480p_stereo.ogg
166.8 MB
15:03
big_buck_bunny_480p_stereo.ogg
Big Buck
BUNNY
fedora

X.org


Sun 15:06
Home
bbb
Recent
Home
Documents
Downloads
Pictures
Trash
Music
Videos
Other Locations
Name
Size
Modified
big_buck_bunny_480p_stereo.ogg
166.8 MB
15:03
big_buck_bunny_480p_stereo.ogg
Big Buck
BUNNY
fedora

Mesa3D


Sun 15:06
Home
bbb
Recent
Home
Documents
Downloads
Pictures
Trash
Music
Videos
Other Locations
Name
Size
Modified
big_buck_bunny_480p_stereo.ogg
166.8 MB
15:03
big_buck_bunny_480p_stereo.ogg
Big Buck
BUNNY
fedora

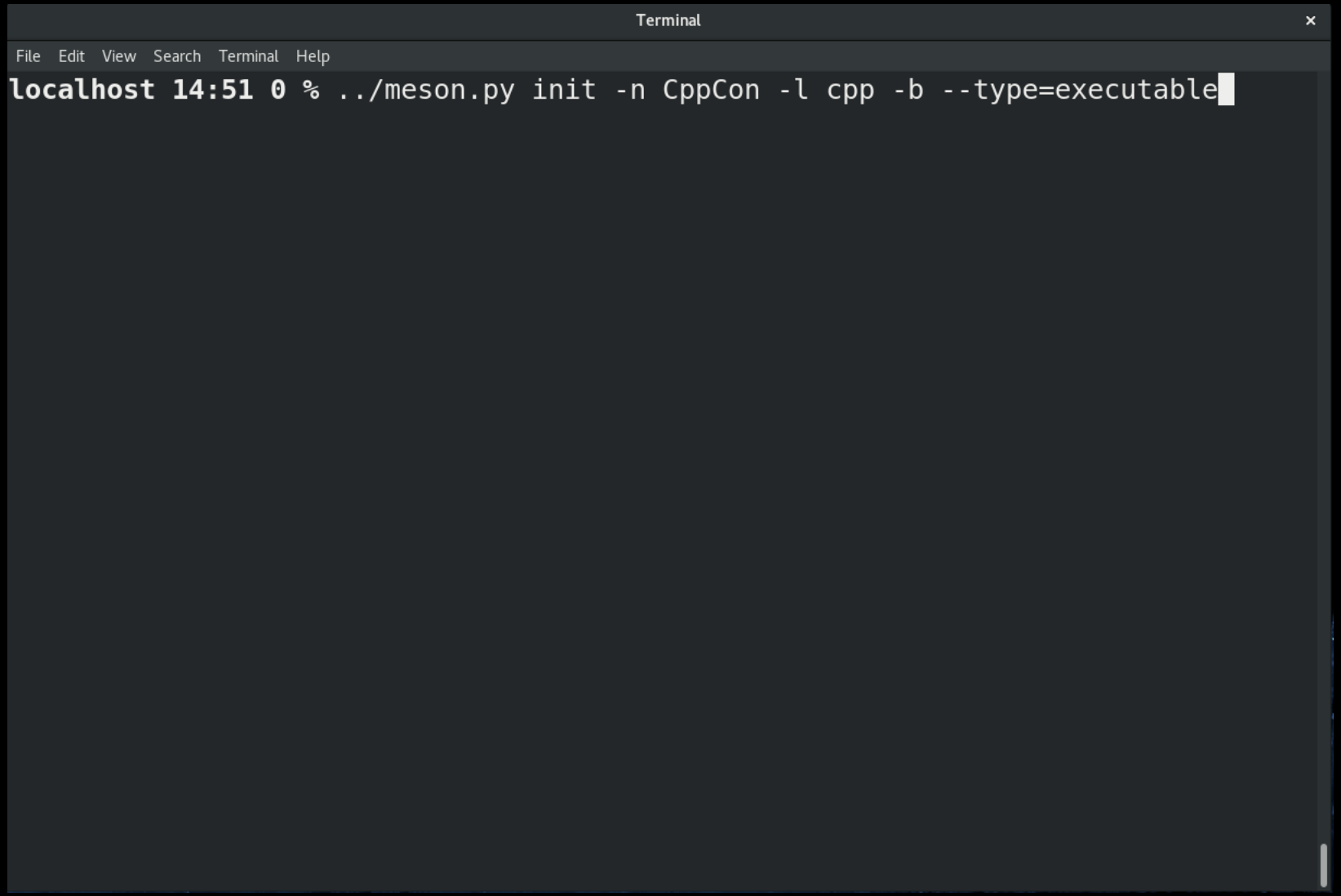
systemd

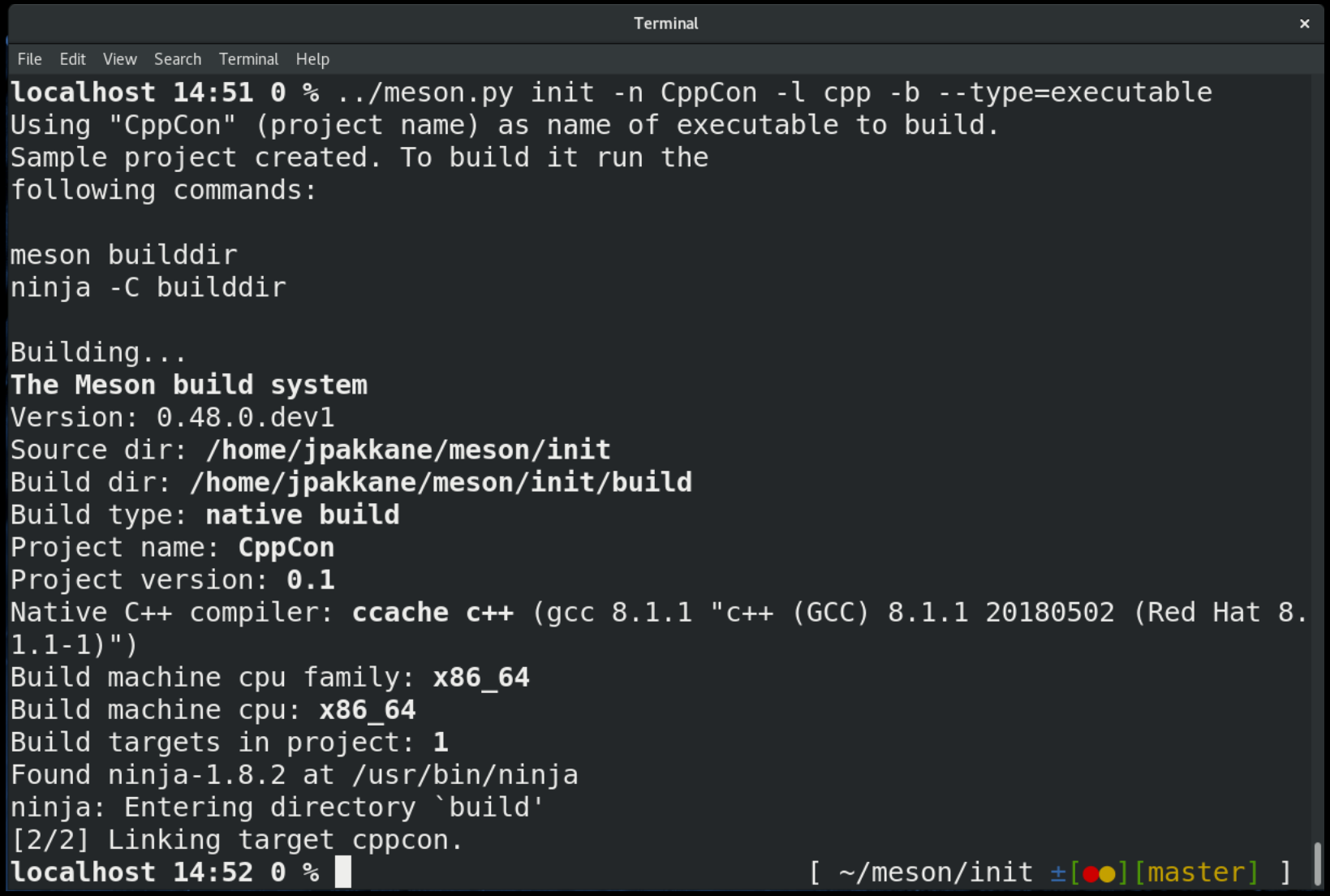
Proprietary projects

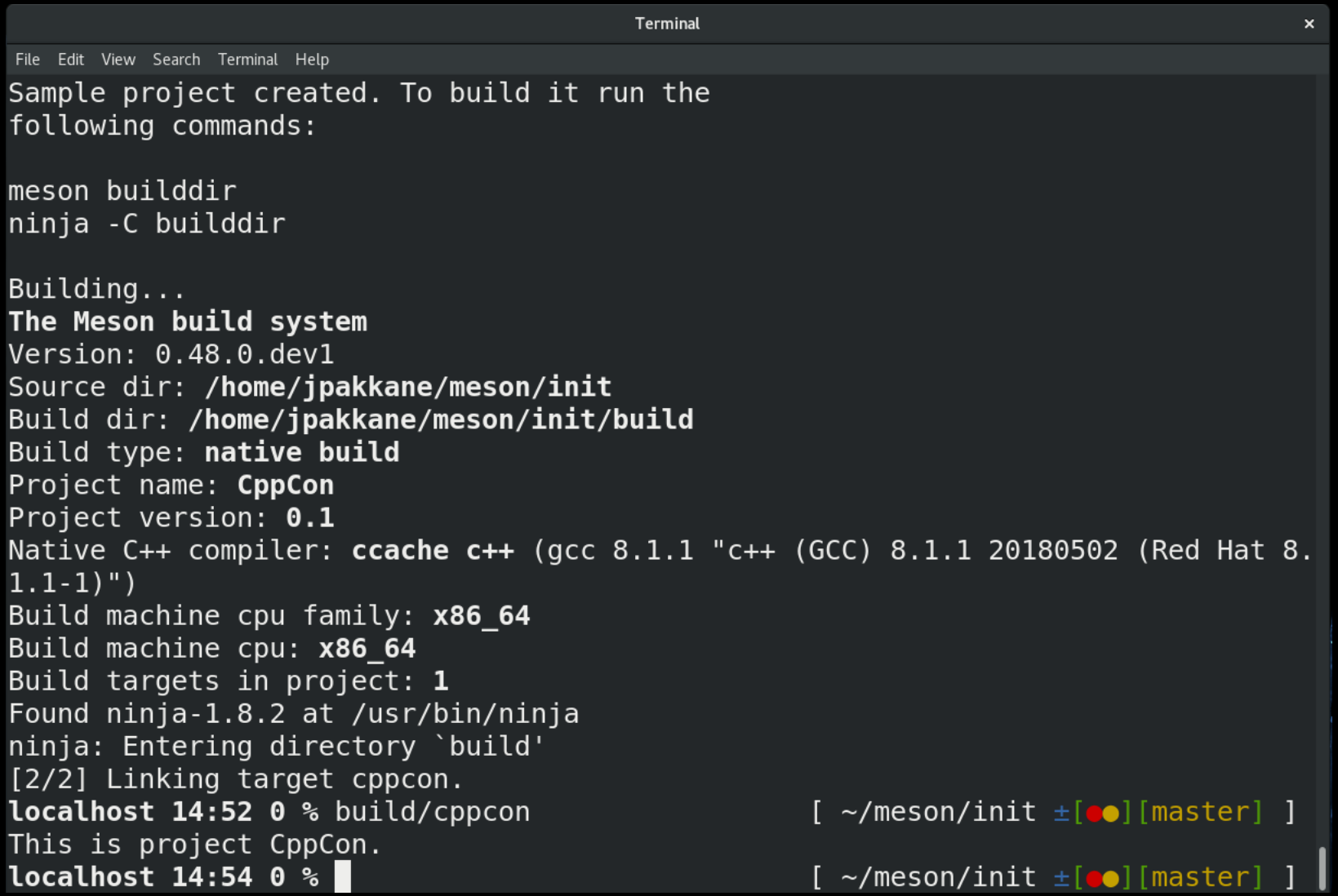
PART TWO

In which code is compiled.

Let’s create a simple project


Terminal
File Edit View Search Terminal Help
localhost 14:51 0 % ../meson.py init -n CppCon -l cpp -b --type=executable


Terminal
File Edit View Search Terminal Help
localhost 14:51 0 % ../meson.py init -n CppCon -l cpp -b --type=executable
Using "CppCon" (project name) as name of executable to build.
Sample project created. To build it run the
following commands:
meson builddir
ninja -C builddir
Building...
The Meson build system
Version: 0.48.0.dev1
Source dir: /home/jpakkane/meson/init
Build dir: /home/jpakkane/meson/init/build
Build type: native build
Project name: CppCon
Project version: 0.1
Native C++ compiler: ccache c++ (gcc 8.1.1 "c++ (GCC) 8.1.1 20180502 (Red Hat 8.
1.1-1)")
Build machine cpu family: x86_64
Build machine cpu: x86_64
Build targets in project: 1
Found ninja-1.8.2 at /usr/bin/ninja
ninja: Entering directory `build'
[2/2] Linking target cppcon.
localhost 14:52 0 %
[~/meson/init ±[●●][master]]


Terminal
File Edit View Search Terminal Help
Sample project created. To build it run the
following commands:
meson builddir
ninja -C builddir
Building...
The Meson build system
Version: 0.48.0.dev1
Source dir: /home/jpakkane/meson/init
Build dir: /home/jpakkane/meson/init/build
Build type: native build
Project name: CppCon
Project version: 0.1
Native C++ compiler: ccache c++ (gcc 8.1.1 "c++ (GCC) 8.1.1 20180502 (Red Hat 8.
1.1-1)")
Build machine cpu family: x86_64
Build machine cpu: x86_64
Build targets in project: 1
Found ninja-1.8.2 at /usr/bin/ninja
ninja: Entering directory `build'
[2/2] Linking target cppcon.
localhost 14:52 0 % build/cppcon
[~/meson/init ±[●●][master]]
This is project CppCon.
localhost 14:54 0 %
[~/meson/init ±[●●][master]]

The build definition

```
project('cppcon', 'cpp',
  version : '0.1',
  default_options : 'cpp_std=c++14')

exe = executable('cppcon', 'cppcon.cpp',
  install : true)

test('basic', exe)
```

Things this build definition gives you.

- Build types
- Language standard
- Test suite runner
- Parallelized unit tests
- Coverage reports
- Warning level toggle
- Unity builds

- Precompiled headers
- Cross compilation support
- LTO/PGO
- Sanitizer support
- Scan-build support
- MSVCRT type selection
- Symbol based relink skipping

PART THREE

In which we look at dependencies,
i.e. the thing that is actually hard.

Why is it so hard?

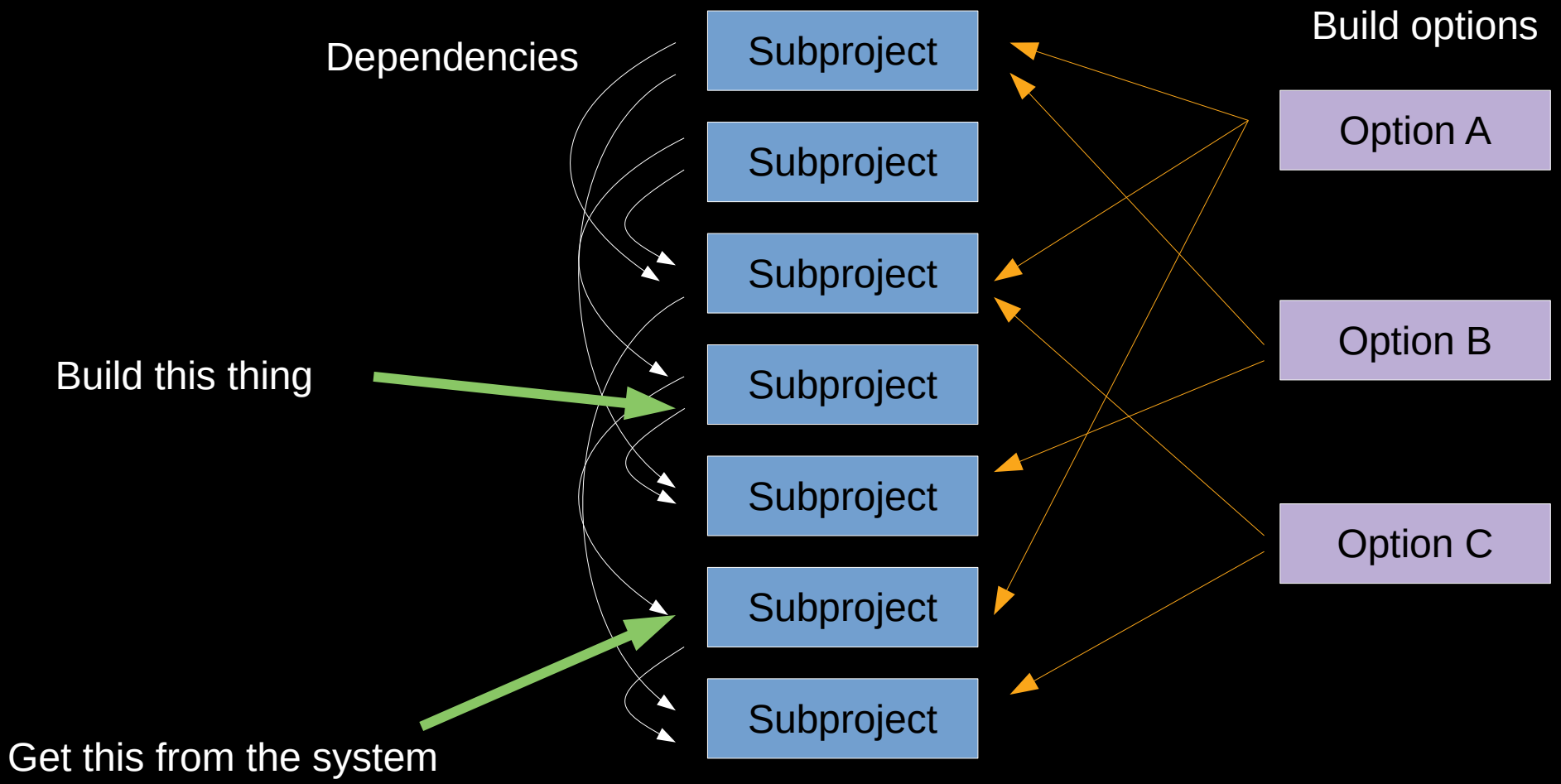
The big project menagerie

- Lots of subprojects with many dependencies
- External and internal dependencies
- Can come from the system or be self built
- Shared or dynamic libraries or header-only
- Monorepo or Git submodules or tarballs
- Diamond dependencies
- Several distinct “deliverables” such as games or hardware specific bundles
- Shared project options across (random) subprojects

Extra bonus challenge

Must be able to compile any subproject
and its direct dependencies only,
even in a monorepo.

Editing low level libraries like a string class
is not feasible if every change rebuilds the world.


Dependencies
Build options
Subproject
Subproject
Subproject
Subproject
Subproject
Subproject
Subproject
Option A
Option B
Option C
Build this thing
Get this from the system

What if I told you ...

... that Meson supports all these cases?

With only three primitives.

1. Sandboxed subprojects

2. Interchangeable dependency objects

3. Yielding project options

Build definition with dependency

```
project('cppcon', 'cpp',
  version : '0.1',
  default_options : 'cpp_std=c++14')

lua_dep = dependency(‘lua’, fallback : [‘lua’, ‘lua_dep’])

exe = executable('cppcon', 'cppcon.cpp',
  install : true,
  dependencies : lua_dep)
```

Now you can build it!
Any subsection.
Any dependency combination.
With no changes to build definition files.

(Live demo omitted due to lack of time)

PART FOUR

In which we ask about that Python thing.

“The C++ build system/package manager should be implemented in C++.”

“The Rust build system/package manager should be implemented in Rust.”

“The Go build system/package manager should be implemented in Go.”

“The D build system/package manager should be implemented in D.”

Who can spot the obvious problem?

“Anyone claiming to have a perfect programming language is either a salesman or a fool and probably both.”

Bjarne Stroustrup

Single-language build systems and dependency managers are silos.

They isolate and hinder cooperation.

In a cooperative multilanguage project almost no language will have a package manager written in itself.

A main design goal of Meson has been to be able to mix and match languages freely.

Let’s build a Python extension

C
C++
Rust
Fortran

in the same module

https://github.com/jpakkane/polysnake

```
project('polysnake', 'c', 'cpp', 'rust', 'fortran')

py3_mod = import('python3')
py3_dep = dependency('python3')

rustlib = static_library('func', 'func.rs') # Rust is special.

py3_mod.extension_module('polysnake',
  'polysnake.c', 'func.cpp', 'ffunc.f90',
  link_with : rustlib,
  dependencies : py3_dep)
```

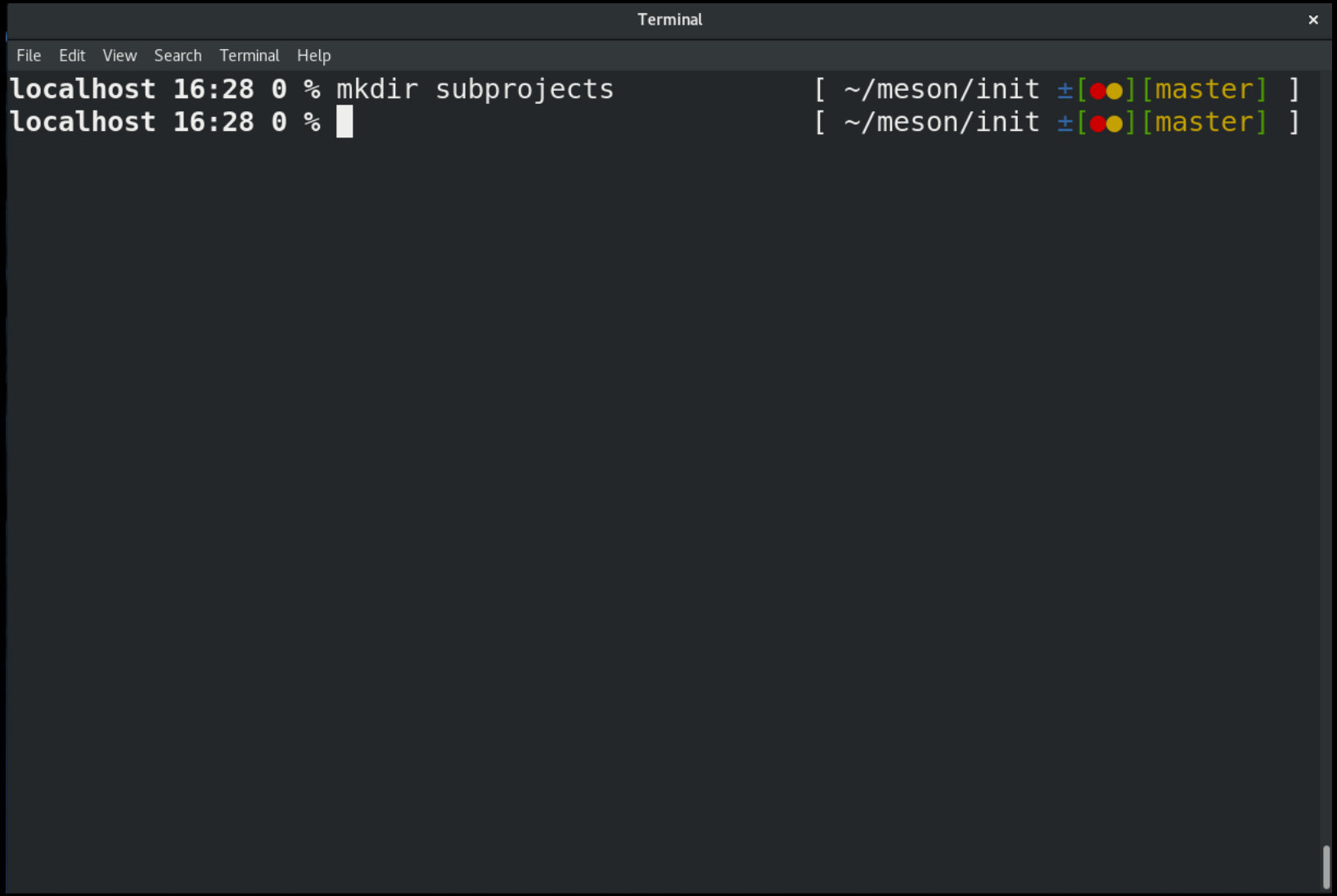
Meson is conceptually similar to SQL.

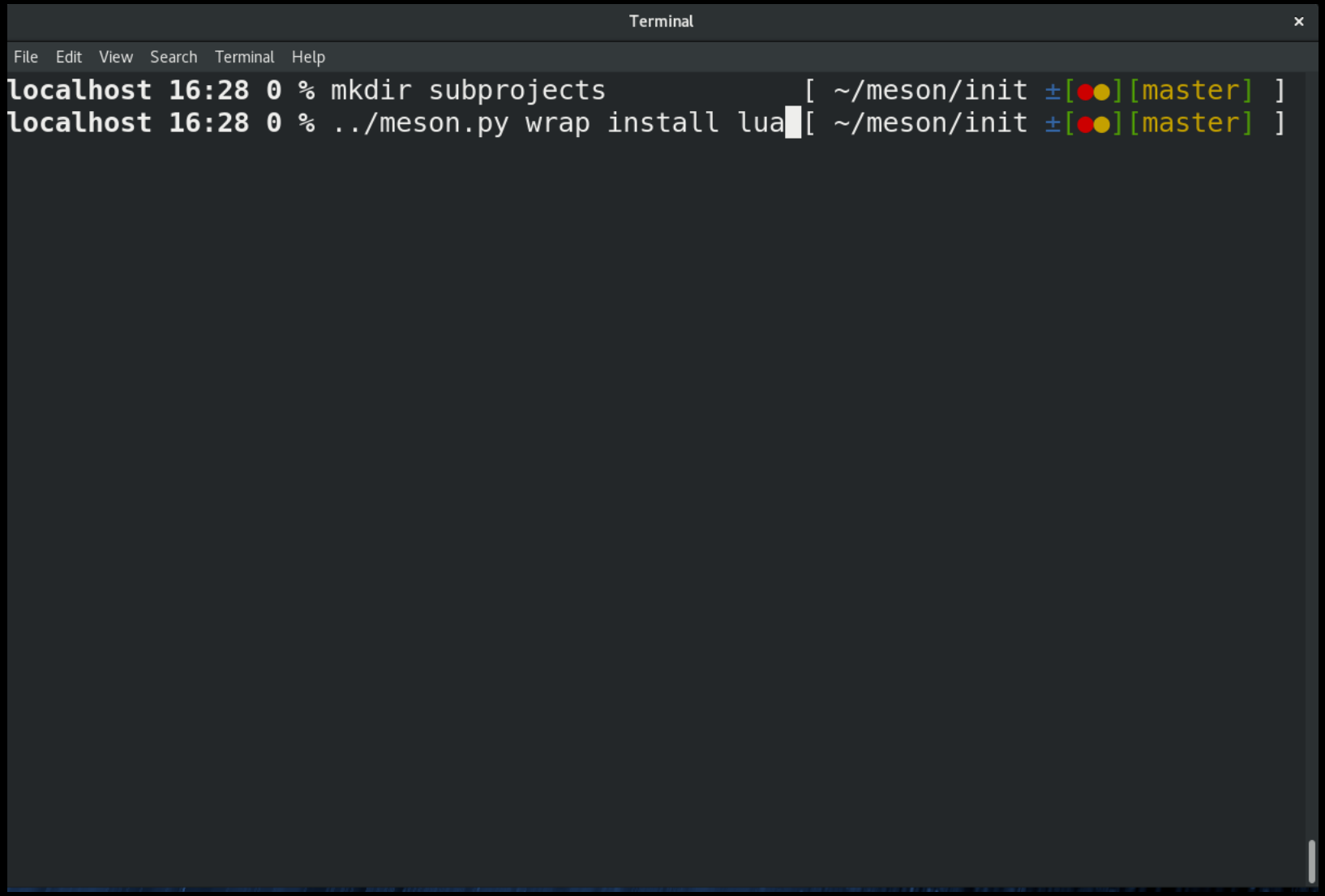
Explain *what*.
Not *how*.

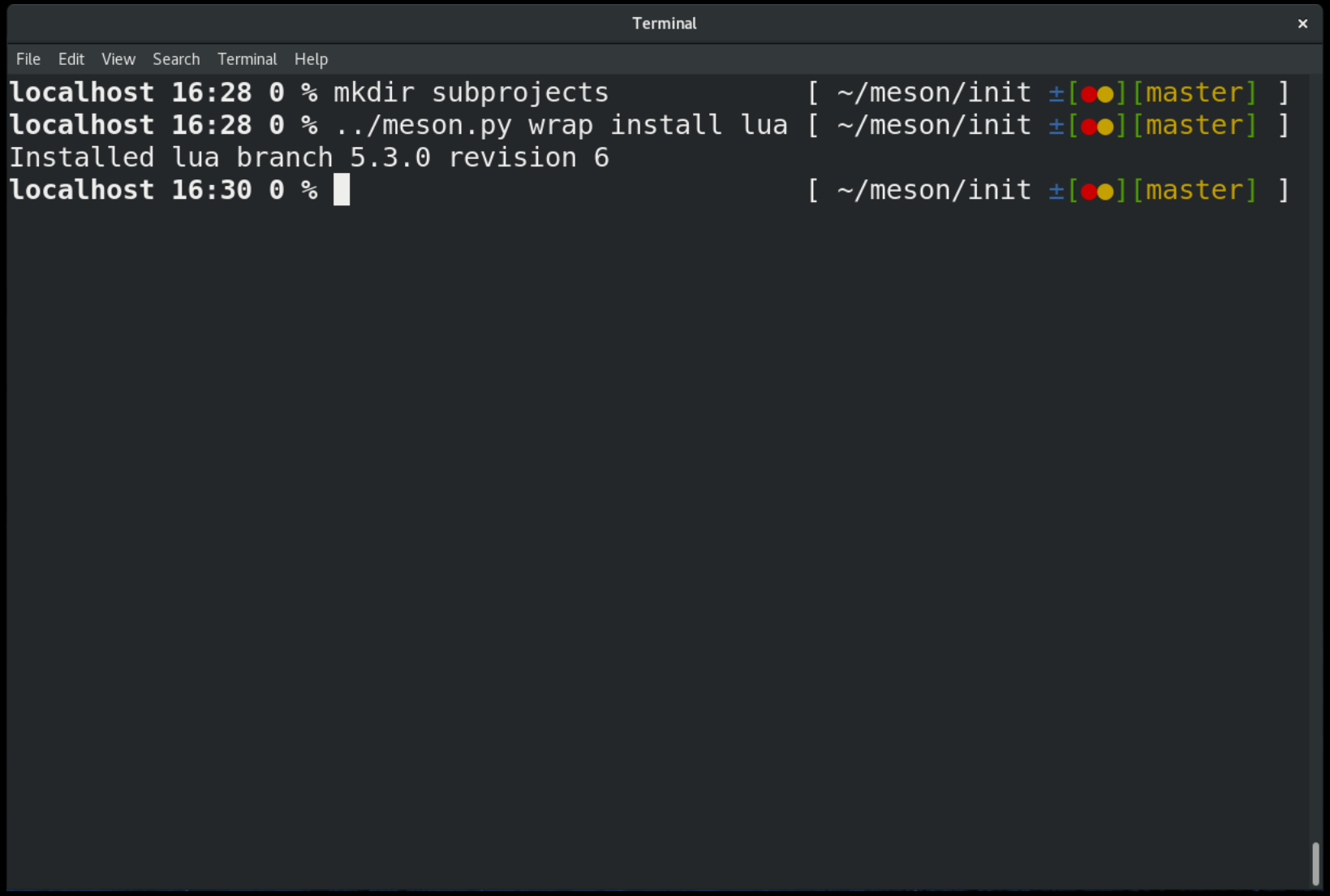
PART FIVE

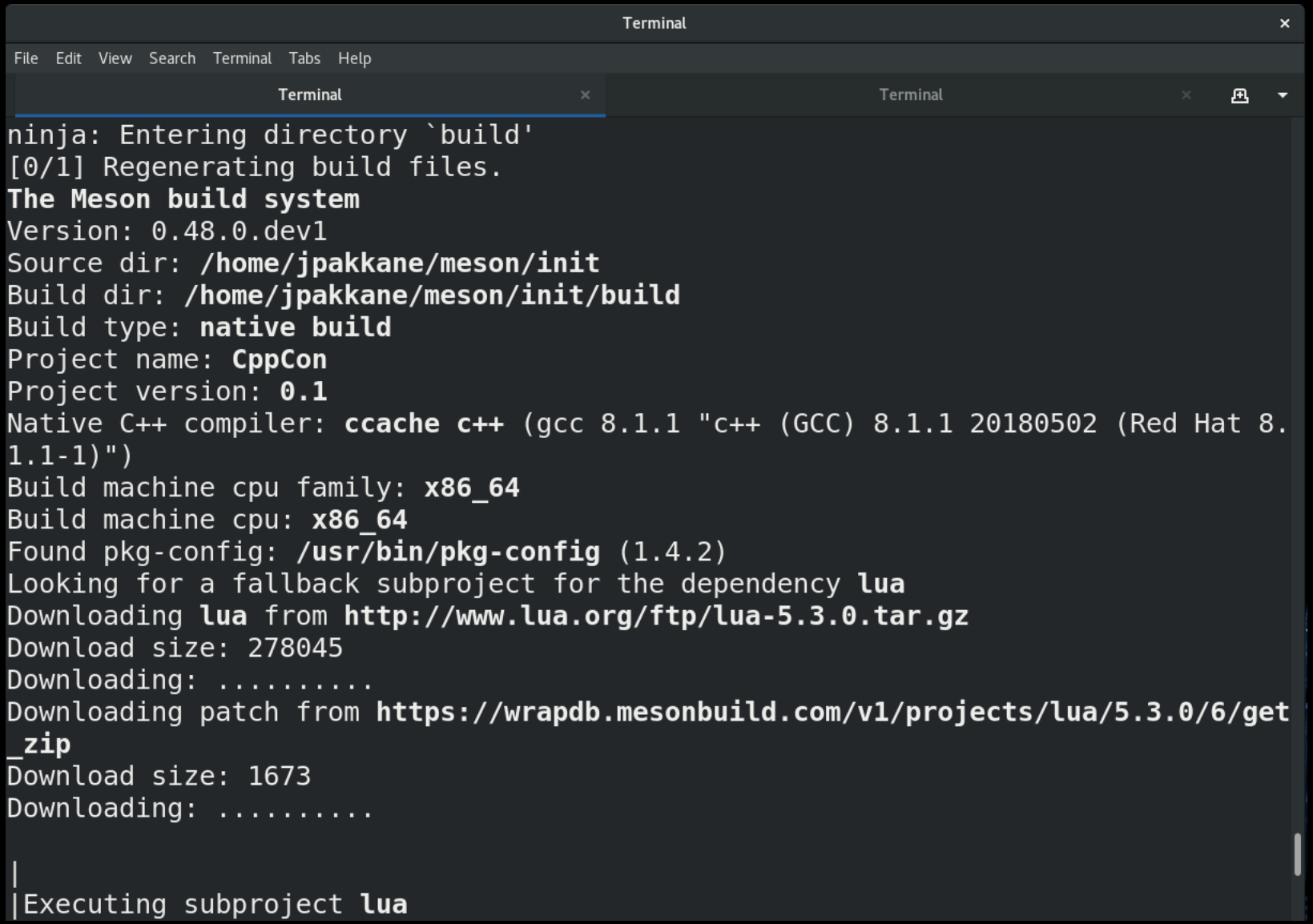
In which there is time for bonus slides.

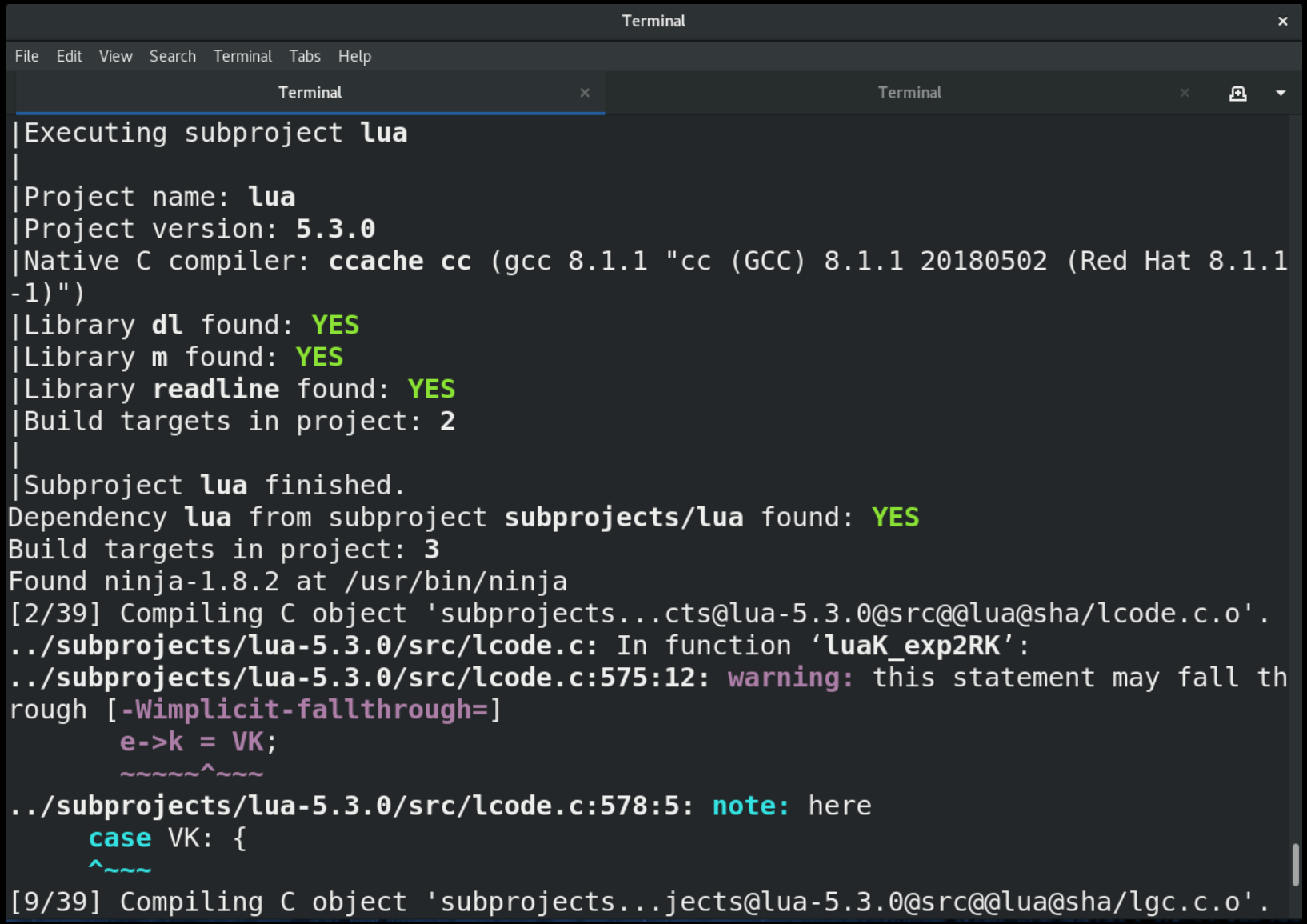
Adding Lua to the sample project

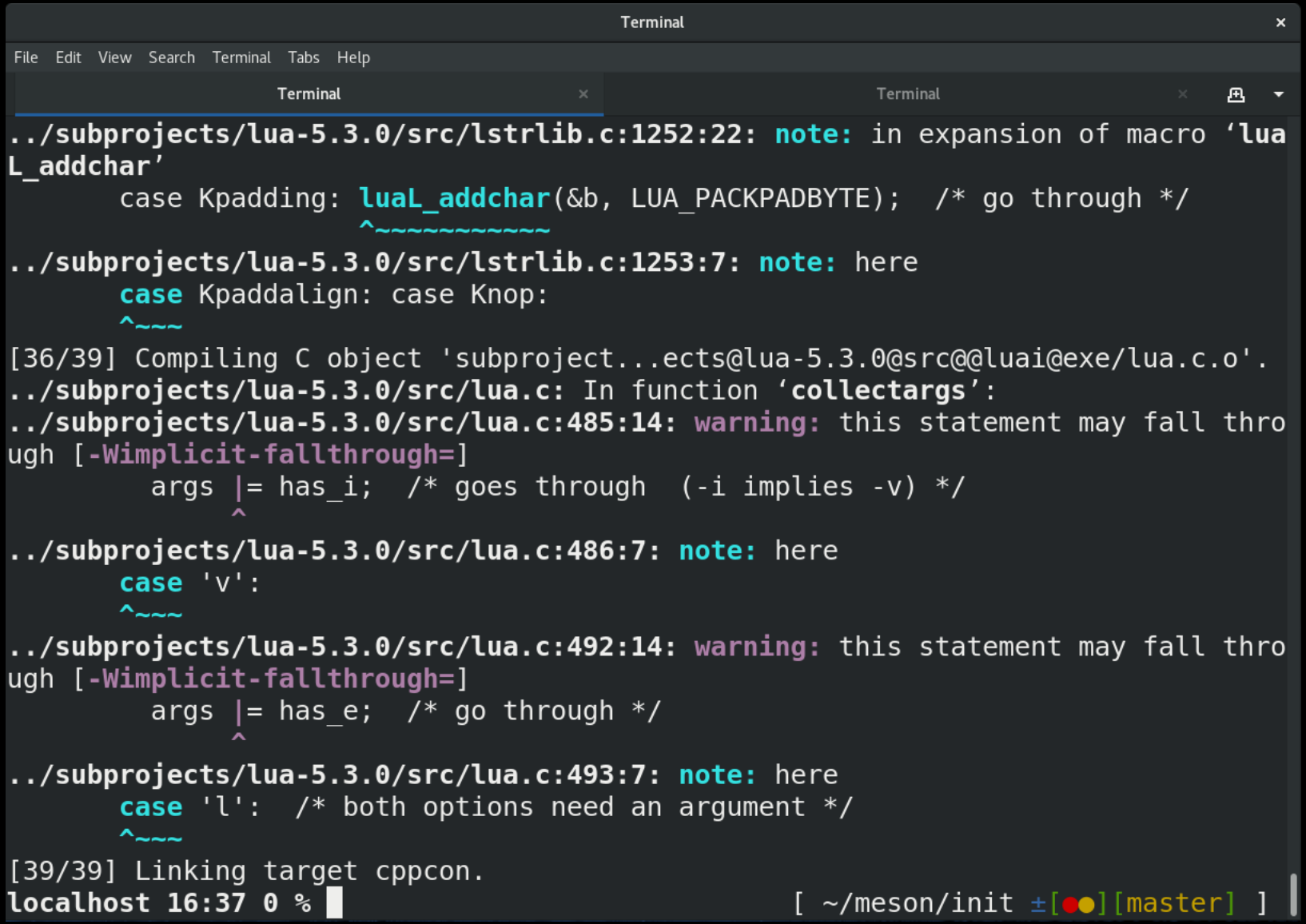

Terminal
File Edit View Search Terminal Help
localhost 16:28 0 % mkdir subprojects [~/meson/init ±[●●][master]]
localhost 16:28 0 % [~/meson/init ±[●●][master]]


Terminal
File Edit View Search Terminal Help
localhost 16:28 0 % mkdir subprojects [~/meson/init ±[●●][master]]
localhost 16:28 0 % ../meson.py wrap install lua[~/meson/init ±[●●][master]]


Terminal
File Edit View Search Terminal Help
localhost 16:28 0 % mkdir subprojects [~/meson/init ±[●●][master]]
localhost 16:28 0 % ../meson.py wrap install lua [~/meson/init ±[●●][master]]
Installed lua branch 5.3.0 revision 6
localhost 16:30 0 % [~/meson/init ±[●●][master]]


Terminal
File Edit View Search Terminal Tabs Help
Terminal
Terminal
ninja: Entering directory `build'
[0/1] Regenerating build files.
The Meson build system
Version: 0.48.0.dev1
Source dir: /home/jpakkane/meson/init
Build dir: /home/jpakkane/meson/init/build
Build type: native build
Project name: CppCon
Project version: 0.1
Native C++ compiler: ccache c++ (gcc 8.1.1 "c++ (GCC) 8.1.1 20180502 (Red Hat 8.1.1-1)")
Build machine cpu family: x86_64
Build machine cpu: x86_64
Found pkg-config: /usr/bin/pkg-config (1.4.2)
Looking for a fallback subproject for the dependency lua
Downloading lua from http://www.lua.org/ftp/lua-5.3.0.tar.gz
Download size: 278045
Downloading: ....
Downloading patch from https://wrapdb.mesonbuild.com/v1/projects/lua/5.3.0/6/get_zip
Download size: 1673
Downloading: ....
|
|Executing subproject lua


Terminal
File Edit View Search Terminal Tabs Help
Terminal
Terminal
|Executing subproject lua
|
|Project name: lua
|Project version: 5.3.0
|Native C compiler: ccache cc (gcc 8.1.1 "cc (GCC) 8.1.1 20180502 (Red Hat 8.1.1
-1)")
|Library dl found: YES
|Library m found: YES
|Library readline found: YES
|Build targets in project: 2
|
|Subproject lua finished.
Dependency lua from subproject subprojects/lua found: YES
Build targets in project: 3
Found ninja-1.8.2 at /usr/bin/ninja
[2/39] Compiling C object 'subprojects...cts@lua-5.3.0@src@@lua@sha/lcode.c.o'.
../subprojects/lua-5.3.0/src/lcode.c: In function ‘luaK_exp2RK’:
../subprojects/lua-5.3.0/src/lcode.c:575:12: warning: this statement may fall th
rough [-Wimplicit-fallthrough=]
e->k = VK;
~~~~~^~~~
../subprojects/lua-5.3.0/src/lcode.c:578:5: note: here
case VK: {
^~~~
[9/39] Compiling C object 'subprojects...jects@lua-5.3.0@src@@lua@sha/lgc.c.o'.
~~~~~


Terminal
File Edit View Search Terminal Tabs Help
Terminal
Terminal
../subprojects/lua-5.3.0/src/lstrlib.c:1252:22: note: in expansion of macro ‘luaL_addchar’
case Kpadding: luaL_addchar(&b, LUA_PACKPADBYTE); /* go through */
^~~~~~~~~~~~
../subprojects/lua-5.3.0/src/lstrlib.c:1253:7: note: here
case Kpaddalign: case Knop:
^~~~
[36/39] Compiling C object 'subproject...ects@lua-5.3.0@src@@luai@exe/lua.c.o'.
../subprojects/lua-5.3.0/src/lua.c: In function ‘collectargs’:
../subprojects/lua-5.3.0/src/lua.c:485:14: warning: this statement may fall through [-Wimplicit-fallthrough=]
args |= has_i; /* goes through (-i implies -v) */
^
../subprojects/lua-5.3.0/src/lua.c:486:7: note: here
case 'v':
^~~~
../subprojects/lua-5.3.0/src/lua.c:492:14: warning: this statement may fall through [-Wimplicit-fallthrough=]
args |= has_e; /* go through */
^
../subprojects/lua-5.3.0/src/lua.c:493:7: note: here
case 'l': /* both options need an argument */
^~~~
[39/39] Linking target cppcon.
localhost 16:37 0 %
[~/meson/init ±[●●][master]]