chapter1.9 cmake_Building_with_ninja

构建工具：ninja
CMake是一个元构建系统，可用于为许多其他构建工具创建构建文件。 这个例子展示了如何让CMake使用ninja构建工具

1. 文件构成：

```
1   $ tree
2   .
3   ├── CMakeLists.txt
4   ├── main.cpp
5   |
6   |
7   |--- build.ninja
```

2. 按顺序介绍：

2.1 查看电脑适配的生成器(Generators)

> 运行cmake --help将显示可用的生成器：

1. 我使用的是ubuntu22.04虚拟机，显示cmake版本：

```
huluobo@ubuntu:/Users/huluobo/cmake-examples/myCmake/chapter1.9$ cmake --version
cmake version 3.25.1

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

2. cmake --help

```
Generators

The following generators are available on this platform (* marks default):
  Green Hills MULTI            = Generates Green Hills MULTI files
                                 (experimental, work-in-progress).
* Unix Makefiles               = Generates standard UNIX makefiles.
  Ninja                        = Generates build.ninja files.
  Ninja Multi-Config           = Generates build-<Config>.ninja files.
  Watcom WMake                 = Generates Watcom WMake makefiles.
  CodeBlocks - Ninja           = Generates CodeBlocks project files.
  CodeBlocks - Unix Makefiles  = Generates CodeBlocks project files.
  CodeLite - Ninja             = Generates CodeLite project files.
  CodeLite - Unix Makefiles    = Generates CodeLite project files.
  Eclipse CDT4 - Ninja         = Generates Eclipse CDT 4.0 project files.
  Eclipse CDT4 - Unix Makefiles= Generates Eclipse CDT 4.0 project files.
  Kate - Ninja                 = Generates Kate project files.
  Kate - Unix Makefiles        = Generates Kate project files.
  Sublime Text 2 - Ninja       = Generates Sublime Text 2 project files.
  Sublime Text 2 - Unix Makefiles
                               = Generates Sublime Text 2 project files.

huluobo@ubuntu:/Users/huluobo/cmake-examples/myCmake/chapter1.9$
```

2.2 生成器分类介绍

如本文所指定，CMake包括不同类型的生成器，例如命令行，IDE和其他生成器。

[1] Command-Line Build Tool Generators命令行编译工具生成器

这些生成器用于命令行构建工具，例如Make和Ninja。 在使用CMake生成构建系统之前，必须先配置所选的工具链。

支持的生成器包括：

* Borland Makefiles
* MSYS Makefiles

- MinGW Makefiles
- NMake Makefiles
- NMake Makefiles JOM
- Ninja ninja用的
- Unix Makefiles make用的
- Watcom WMake

[2] IDE Build Tool Generators IDE内置编译工具生成器

这些生成器用于自己有编译器的IDE。 示例是Visual Studio和Xcode。

The supported generators include:

- Visual Studio 6
- Visual Studio 7
- Visual Studio 7 .NET 2003
- Visual Studio 8 2005
- Visual Studio 9 2008
- Visual Studio 10 2010
- Visual Studio 11 2012
- Visual Studio 12 2013
- Xcode

[3] Extra Generators 可嵌入IDE的编译工具生成器

These are generators create a configuration to work with an alternative IDE tool and must be included with either an IDE or Command-Line generator.这些生成器，用于其他IDE工具一起使用的配置，并且必须包含在IDE或命令行生成器中。

The supported generators include:

- CodeBlocks
- CodeLite
- Eclipse CDT4
- KDevelop3
- Kate
- Sublime Text 2

3. 调用生成器：

- 使用-G参数来唤醒CMake的生成器

```
cmake .. —G Ninja
```

- 完成上述操作后，CMake将生成所需的Ninja构建文件，可以使用ninja命令运行该文件

```
ls
build.ninja  CMakeCache.txt  CMakeFiles  cmake_install.cmake  rules.ninja
```

4. 总览：

注意：orb在我这里是进入虚拟机的意思

```
问题   输出   调试控制台   终端   端口                                      + ∨  >_ scli - build.ninja  □  🗑  ···  ∧  ✕

 huluobo@huluobodeMacBook-Pro    ~/cmake-examples/myCmake/chapter1.9/build.ninja    main ±   orb
huluobo@ubuntu:/Users/huluobo/cmake-examples/myCmake/chapter1.9/build.ninja$ cmake .. -G Ninja
-- The C compiler identification is GNU 12.3.0
-- The CXX compiler identification is GNU 12.3.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/huluobo/cmake-examples/myCmake/chapter1.9/build.ninja
huluobo@ubuntu:/Users/huluobo/cmake-examples/myCmake/chapter1.9/build.ninja$ ninja -v
[1/2] /usr/bin/c++    -MD -MT CMakeFiles/hello_cmake.dir/main.cpp.o -MF CMakeFiles/hello_cmake.dir/main.cpp.o.d -
o CMakeFiles/hello_cmake.dir/main.cpp.o -c /Users/huluobo/cmake-examples/myCmake/chapter1.9/main.cpp
[2/2] : && /usr/bin/c++    CMakeFiles/hello_cmake.dir/main.cpp.o -o hello_cmake   && :
huluobo@ubuntu:/Users/huluobo/cmake-examples/myCmake/chapter1.9/build.ninja$  ls
CMakeCache.txt   CMakeFiles   build.ninja  cmake_install.cmake   hello_cmake
huluobo@ubuntu:/Users/huluobo/cmake-examples/myCmake/chapter1.9/build.ninja$ ./hello_cmake
Hello CMake!
huluobo@ubuntu:/Users/huluobo/cmake-examples/myCmake/chapter1.9/build.ninja$ []
```

review:

- `ninja -v` 是一个命令行指令，用于运行 Ninja 构建系统并显示详细的构建信息。
- Ninja 是一个小型的构建系统，专注于速度。它被设计用于大型项目，如 Chrome 等，其中构建文件可以非常大。
- `-v` 是一个命令行选项，表示 "verbose"（详细）。当使用 `-v` 选项时，Ninja 会显示更多的信息，包括它正在执行的每个命令。这对于调试构建问题非常有用