# Makefile从何而来

Makefile系列

本文从编译入手介绍makefile的由来，了解makefile的产生原因，加深对makefile的理解，从而有助于后续学习makefile.

## Make的出现

makefile是由make命令执行的，要想了解mkefile不得不提make。

1977年计算机科学家斯图亚特·费尔德曼（Stuart Feldman）在贝尔实验室（Bell Labs）创立了make。make是一个自动化构建工具，是一个应用程序，作用是通过读取一种叫做Makefile的文件，将源代码自动构建成可执行程序和库文件。

Unix系统中包含了make，GNU/Linux衍生自Unix衍生保留并扩展了原始的make，并加入了许多内置函数和自动变量对make进行了发展，形成了GNU make。make已经被写入了系统环境变量。GNU/Linux下make的路径一般为：/usr/bin/make。目前大多数的IDE都集成了这个命令，比如：Delphi的make，Visual C++的nmake。

make早期主要用于构建C语言开发的项目，后来逐渐发展，make可以适用于任何语言了，并且广泛用于构建C、C++、java等各种语言开发的项目。事实上，make并不局限于程序。可以使用它来描述任何适合make规则的任务，在这个任务中当其他文件发生更改时，必须从其他文件自动更新某些文件。

make通过读入并解释Makefile文件获取指令，调度gcc/g++，执行一组以gcc/g++为主的[shell命令](https://so.csdn.net/so/search?q=shell%E5%91%BD%E4%BB%A4&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/m0_46172703/article/details/_blank)序列。make除了可以做到编译和连结，也可以把如何产生文档(如manual page，info档及dvi档)，以及打包压缩都完成。

## Makefile简介

了解了make之后咱们再来看Makefile。

Makefile是供make执行的脚本，Makefile文件中描述了整个工程所有文件的依赖关系、编译规则，以此告诉make以何种方式编译源代码和链接程序。由此可见Makefile是伴随make出现的。

像C/C++、java等编程语言一样，Makefile有自己的书写格式、关键字、函数。在Makefile中可以使用系统shell所提供的任何命令来完成想要的工作。

## Makefile的文件名

一般Makefile文件取名为“Makefile”、“makefile”或“GNUmakefile”，如果当前目录同时存在三个名字的文件，在执行make命令时，经过测试会优先解析“GNUmakefile”，如果没有“GNUmakefile”会解析“makefile”，最后以上两者都没有才轮到解析“Makefile”。不过一般情况只存在三个文件名中的一个，大部分情况推荐“Makefile”，一些特殊情况make只识别全小写的“makefile”文件名。

当然，也可以使用别的文件名来书写Makefile，比如：“Make.Linux”，“Make.Solaris”，“Make.AIX”等，如果要指定特定的Makefile，可以使用make的“-f”和“--file”参数，如：make -f Make.Linux或make --file Make.AIX。

## Make和Makefile解决的问题

当项目工程非常小时甚至只有一个源文件，直接就可以用gcc命令编译它。但是当项目工程包含很多个源文件时，用gcc命令逐个去编译，就很容易出现混乱而且工作量大。所以我们需要一个工具来管理这些编译过程，那就是make。

make工具类似于批处理程序，可以自动化编译大量源文件，可以使用一条命令实现完全编译，大大提高了编译效率。当项目工程经过一次编译后，在对程序进行修改，根据我们编写的Makefile中的规则和依赖关系，用make进行编译时，可以自动确定哪些文件需要重新编译，只编译修改的文件和对其有依赖关系的文件，而不用再次把所有文件都编译一遍，这样大大节省了时间。

## 涉及的面试问题

make和Makefile的关系是什么？

make是编译项目工程的工具，Makefile是被make解析的文件。输入make命令后查找当前目录的MAkefile文件，然后进行解析并依据解析结果进行编译工作。

## 参考资料

版权声明：本文参考了其他CSDN博主的文章，遵循CC 4.0 BY-SA版权协议，现附上原文出处链接及本声明。

1. Makefile入门一、helloworld 原文链接：https://blog.csdn.net/hjxu2016/article/details/100672126
2. https://www.cnblogs.com/qytan36/archive/2010/05/25/1743955.html