

Makefile 学习笔记

姜高晓

June 16, 2022

Abstract

这是摘要

1 背景

1.1 什么是Makefile

MakeFile 可以认为是工程文件编译链接的指导书，指挥着编译工程文件的时候的顺序和规则。这可以使得我们的工程目录编译链接自动化，而且有条理。

1.2 基本结构

Makefile的基本结构如下

目标：依赖文件名

命令

- 目标：可以是输出文件，也可是标签
- 依赖文件：编译前要搞定的
- 命令：编译链接命令，也可使shell命令

1.3 Makefile的执行流程

第一规则：在Makefile执行的时候，默认第一个目标就是最终目标main。我们执行 make 命令时，只有修改过的源文件或者是不存在的目标文件会进行重建，而那些没有改变的文件不用重新编译，这样在很大程度上节省时间，提高编程效率。小的工程项目可能体会不到，项目工程文件越大，效果才越明显。

ps: .o文件不要着急清理，等需要清理的时候在清理，这样就可以保证编译时的速度

伪目标：clean, 在执行make clean的时候清除一些垃圾，他和main是并列关系不需要依赖文件。

2 主要内容

2.1 参数引用和通配符

每一个目标-命令结构都可以当作一个脚本，这里可以用\$，引用参数。

- \$# 是传给脚本的参数个数
- \$0 是脚本本身的名字
- \$1 是传递给该shell脚本的第一个参数
- \$2 是传递给该shell脚本的第二个参数
- \$@ 是传给脚本的所有参数的列表，代表目标文件(target)
- \$* 是以一个单字符串显示所有向脚本传递的参数，与位置变量不同，参数可超过 9 个
- \$\$ 是脚本运行的当前进程ID号
- \$? 是显示最后命令的退出状态，0 表示没有错误，其他表示有错误
- @这个字符串通常用在“规则”行中，表示不显示命令本身，而只显示它的结果
- \$^代表所有的依赖文件(components)
- \$<代表第一个依赖文件(components中最左边的那个)。

\$@ 是目标 用以 -o \$@

\$^是依赖 需要处理的文件

Makefile支持shell中的通配符，*, ?, [,].除此之外还支持%.

- *:任意个任意字符。
- ?: 一个任意字符。
- [: 把指定的字符放入方括号中
- %: 匹配字符，例如依赖中依赖 x1.o x2.o x3.o 可以简单写一个 %.o 的目标，统一处理这些依赖

2.2 变量赋值

- 简单赋值 (:=) 编程语言中常规理解的赋值方式，只对当前语句的变量有效。
- 递归赋值 (=) 赋值语句可能影响多个变量，所有目标变量相关的其他变量都受影响。
- 条件赋值 (?:=) 如果变量未定义，则使用符号中的值定义变量。如果该变量已经赋值，则该赋值语句无效。
- 追加赋值 (+=) 原变量用空格隔开的方式追加一个新值。

2.3 GCC

3 总结

3.1 Subsection title

这里是总结

References

- [1] <http://c.biancheng.net/view/7097.html>
- [2] <https://blog.csdn.net/u014285914/article/details/124048754>