

知识点 1 【makefile 的概述】

1、make

make 是一个命令 当用户输入 make 指令 系统自动寻找 makefile、Makefile、GNUmakefile 中任何一个，根据该文件中的指令编译工程。

2、makefile

makefile 是一个编译指令的脚本 里面纪录的是一条条编译方式

3、采用 Makefile 的好处

- 1、**简化**编译程序的时候输入得命令，编译的时候只需要敲 make 命令就可以了
- 2、可以**节省编译时间**，提高**编译效率**。

知识点 2 【makefile 的语法规则】

目标：依赖文件列表

<Tab>命令列表

目标文件 依赖 "依赖文件列表" 而命令列表是具体将"依赖文件列表"生成"目标文件"的具体指令

用户在终端上输入 make 指令 自动寻找 makefile 只会执行 makefile 中第一条"目标:依赖文件列表"

```
main:main.c fun.c
```

```
gcc main.c fun.c -o main
```

make 命令格式

输入 make 默认寻找 makefile、Makefile、GNUmakefile 中任何一个。

make -f 用户指定的文件名

make 后跟其他目标 这样可以执行其他目标

知识点 3 【稍稍复杂的 makefile】

```
main:main.o fun.o

    gcc main.o fun.o -o main

main.o:main.c

    gcc -c main.c -o main.o

fun.o:fun.c

    gcc -c fun.c -o fun.o

clean:

    rm main *.o
```

知识点 4 【makefile 的变量】

1、自定义变量：用户自己定义的变量

makefile 的变量名：

makefile 变量名可以以数字开头

注意：

- 1、变量是大小写敏感的
- 2、变量一般都在 makefile 的头部定义
- 3、变量几乎可在 makefile 的任何地方使用

变量名=变量值

如果需要读取变量的值 必须给钱而且函数美元。num=10 \$num 才是取变量的值

2、系统环境变量

make 工具会拷贝系统的环境变量并将其设置为 makefile 的变量，在 makefile 中可直接读取或修改拷贝后的变量。

```
#export test=10 export 导出环境变量
```

```
#make clean
```

```
#echo $test
```

3、预定义变量 (makefile 定义好的变量)

\$@ 目标名

\$< 依赖文件列表中的第一个文件

\$^ 依赖文件列表中除去重复文件的部分

AR 归档维护程序的程序名，默认值为 ar

ARFLAGS 归档维护程序的选项

AS 汇编程序的名称，默认值为 as

ASFLAGS 汇编程序的选项

CC C 编译器的名称，默认值为 cc

CFLAGS C 编译器的选项

CPP C 预编译器的名称，默认值为 \$(CC) -E

CPPFLAGS C 预编译的选项

CXX C++ 编译器的名称，默认值为 g++

CXXFLAGS C++ 编译器的选项

高级的 makefile

```
CC=gcc
```

EXEC=main

OBJ=main.o fun.o

\$(EXEC):\$(OBJ)

\$(CC) \$^ -o \$@

%.o:%.c

\$(CC) -c \$< -o \$@

clean:

rm \$(EXEC) \$(OBJ)