

Makefile 入门实验

假设我们有一个程序由5 个文件组成，源代码如下：

```
/*main.c*/
#include "mytool1.h"
#include "mytool2.h"
int main()
{
    mytool1_print("hello mytool1!");
    mytool2_print("hello mytool2!");
    return 0;
}
/*mytool1.c*/
#include "mytool1.h"
#include <stdio.h>
void mytool1_print(char *print_str)
{
    printf("This is mytool1 print : %s ",print_str);
}
/*mytool1.h*/
#ifndef _MYTOOL_1_H
#define _MYTOOL_1_H
void mytool1_print(char *print_str);
#endif
/*mytool2.c*/
#include "mytool2.h"
#include <stdio.h>
void mytool2_print(char *print_str)
{
    printf("This is mytool2 print : %s ",print_str);
}
/*mytool2.h*/
#ifndef _MYTOOL_2_H
#define _MYTOOL_2_H
void mytool2_print(char *print_str);
#endif
```

常规法写第一个Makefile

```
main:main.o mytool1.o mytool2.o
```

```
    gcc -o main main.o mytool1.o mytool2.o
```

```
main.o:main.c mytool1.h mytool2.h
```

```
    gcc -c main.c
```

```
mytool1.o:mytool1.c mytool1.h
```

```
    gcc -c mytool1.c
```

```
mytool2.o:mytool2.c mytool2.h
```

```
    gcc -c mytool2.c
```

```
clean:
```

```
    rm -f *.o main
```

在shell 提示符下输入make，执行显示：

```
gcc -c main.c
```

```
gcc -c mytool1.c
```

```
gcc -c mytool2.c
```

```
gcc -o main main.o mytool1.o mytool2.o
```

执行结果如下：

```
[armlinux@lqm makefile-easy]$ ./main
```

```
This is mytool1 print : hello mytool1!
```

```
This is mytool2 print : hello mytool2!
```