

了解tlib如何生成list文件，查看list文件中是什么内容？

- 通过查看使用说明尝试生成list文件

```
C:\>tlib
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International
Syntax: TLIB libname [/C] [/E] commands, listfile
libname      library file pathname
commands     sequence of operations to be performed (optional)
listfile     file name for listing file (optional)

A command is of the form: <symbol>modulename, where <symbol> is:
+           add modulename to the library
-           remove modulename from the library
*           extract modulename without removing it
-+ or +-   replace modulename in library
-* or *-   extract modulename and remove it

/C          case-sensitive library
/E          create extended dictionary

Use @filepath to continue from file "filepath".
Use '&' at end of a line to continue onto the next line.

C:\>TLIB.EXE CS.LIB ,cslistfile
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International
C:\>
```

- 通过生成的 `CSLISTFI.LST` 可以看到文件中包含着标号，大小，但不知道标号是 `obj` 文件的名称还是其中的函数名称故通过自己的 `c` 文件生成 `obj` 然后加入 `cs.lib` 中来验证

- 生成的list文件

```
1  Publics by module
2
3  ABS          size = 16
4      _abs
5
6  ABSREAD      size = 76
7      _absread          _abswrite
8
9  ACCESS       size = 57
10     _access
11
12  ALLOCMEM     size = 36
13     _allocmem
14
15  ATEXIT       size = 104
16     __atexitcnt        __atexittbl
17     _atexit
18
19  ATOL         size = 128
20     .....
```

- test.c

```

1  int t1(int a, int b) {
2      int c;
3      c = a + b;
4      return c;
5  }
6  int t2(int a, int b) {
7      int c;
8      c = a - b;
9      return c;
10 }
11 int t3(int a, int b) { return a + b + 1; }

```

```

C:\>tcc -nsrc\four -c \SRC\FOUR\TEST.C 生成 obj
Turbo C Version 2.0 Copyright (c) 1987, 1988 Borland International
\src\four\test.c:

    Available memory 458096

C:\>TLIB.EXE CS.LIB +\SRC\FOUR\TEST.OBJ 将 obj 加入 cs.lib中
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International

C:\>TLIB.EXE CS.LIB CSLISTFI.LST
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International

Error: unexpected char 'C' in command line

C:\>TLIB.EXE CS.LIB ,CSLISTFI.LST 生成list文件
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International

C:\>

```

- 查看 cslistfile

- 可以看到标号为obj文件的名称，大小的obj文件的大小，带下划线的为函数名

```

778 TELL      size = 23
779 |  _tell
780
781 TEST      size = 48
782 |  _t1          _t2
783 |  _t3
784
785 TEXTMODE  size = 46

```

2

使用tlib进行操作时，会进行检测。写程序测试是检测函数名、obj文件名，还是两个都检测？如果都检测，那先检测哪一个？

- 紧接着在添加完test.obj后重新写一个不一样的test.c生成obj并加入cs.lib可以看到添加失败可以看到如果obj名称一致尽管函数名不一致也会添加失败
 - 不一样的test.c

```

1  int f3(int a, int b) { return a + b + 1; }

```

```

C:\>TLIB.EXE CS.LIB + \TEST.OBJ
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: 'TEST' already in LIB, not changed!

```

- 然后将test.c的obj文件拷贝重命名然后在添加到cs.lib中，可以看到虽然obj文件名不一致tlib还检查函数名

```
C:\>TLIB.EXE CS.LIB + \SRC\FOUR\TESTCO~1.OBJ
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International
Error: public '_t3' in module 'TESTCO~1' clashes with prior module 'TEST'
C:\>_
```

- 通过以上可以看出tlib会对函数名和obj文件名进行检查，现在接着将原始的test.obj再加入cslib中tlib会提示obj文件名一致添加失败，所以tlib先检查obj文件名然后检查函数名

```
C:\>TLIB.EXE CS.LIB + TEST.OBJ
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: 'TEST' already in LIB, not changed!
```

3

将同一个.obj文件放入cs.lib和maths.lib中，会从中取该文件中的函数？

- 稍微修改一下test.c然后再生成obj加入maths.lib
 - test.c

```
1  int t1(int a, int b) {
2  +   int c = 9;
3      c = a + b;
4      return c;
5  }
6  int t2(int a, int b) {
7  +   int c = 8;
8      c = a - b;
9      return c;
10 }
11 int t3(int a, int b) { return a + b + 1; }
```

```
C:\>tcc -n\src\four -c \SRC\FOUR\TEST.C          生成obj
Turbo C Version 2.0 Copyright (c) 1987, 1988 Borland International
\src\four\test.c:

    Available memory 458094

C:\>TLIB.EXE MATHS.LIB + \SRC\FOUR\TEST.OBJ      添加到maths.lib
TLIB Version 2.0 Copyright (c) 1987, 1988 Borland International
```

- 编写测试程序
 - main.c

```
1  main() {
2      int a = 1;
3      int b = 2;
4      int c;
5      c = t1(a, b);
6      printf("c = %d", c);
7  }
```

- debug查看生成的可执行文件
 - 可以看出会从maths.lib中去函数

```
-u 0224
076A:0224 55      PUSH    BP
076A:0225 8BEC     MOV     BP,SP
076A:0227 56      PUSH    SI
076A:0228 BE0900   MOV     SI,0009
076A:022B 8B7604   MOV     SI,[BP+04]
076A:022E 037606   ADD     SI,[BP+06]      t1
076A:0231 8BC6     MOV     AX,SI
076A:0233 EB00     JMP     0235
076A:0235 5E      POP     SI
076A:0236 5D      POP     BP
076A:0237 C3      RET
076A:0238 55      PUSH    BP
076A:0239 8BEC     MOV     BP,SP
076A:023B 56      PUSH    SI
076A:023C BE0800   MOV     SI,0008      t2
076A:023F 8B7604   MOV     SI,[BP+04]
076A:0242 2B7606   SUB     SI,[BP+06]
```