**1**

Three layers: list logic interface

list: list\_t.c/list\_t.h

logic: logic.c/logic.h

interface: interface.c/interface.h

Two main functions:

main.c: dispatch the work to function *Commanline* implemented in interface.c, which will run the command line user interface

mainmenu.c: dispatch the work to function *Menu* implemented in interface.c, which will run the Menu user interface

All the source files are in *layer* directory.

**2**

Three stages performed when compiling a file:

1. Pre-processing: When preprocessing, macro are substituted, comments are stripped off and included files are expanded.
2. Compilation: The compiler will generate a .s assembly file and the assembler will take .s file as input and generate an object file.
3. Linking: Combine the program object code with other object code and libraries to produce executable machine code.

Difference between static and dynamic libraries:

Static libraries will be included in the executable and become part of the program, while dynamic libraries are loaded at run time and not linked into the program.

• Create two static libraries, one for each of the two lowest layers in the previous program. Compile the command line version of the program using these two static libraries.

**gcc –c list\_t.c logic.c**

**ar –rc liblist\_static.a list\_t.o**

**ar –rc liblogic\_static.a logic.o**

**gcc –o commandlinemain main.c interface.c –L. –llogic\_static –llist\_static**

*gcc –c \*.c* generates object files.

*ar –rc liblist\_static.a list\_t.o*, *ar –rc liblogic\_static.a logic.o*creates archives (lib is prefix).

*gcc –o commandlinemain main.c interface.c –L. –llogic\_static –llist\_static* creates static libraries. -l ***<***libraryname without lib prefix and extension>. –L : specifies the path to the library. –L. means point to the current directory. list\_static is lower-level than logic\_static, so –llist\_static should be placed behind –llogic\_static.

These four lines create a command line version of the program. To run this program, use ./commandlinemain rand\_int.txt inc, for example.

• Generate two dynamic libraries, one for each of the two lowest layers in the previous program. Compile the whole program. Compile the Menu version of the program using these two dynamic libraries.

**gcc –c –fpic list\_t.c logic.c**

**gcc –shared –o liblist\_dynamic.so list\_t.o**

**gcc –shared –o liblogic\_dynamic.so logic.o**

**gcc –L. –o menumain mainmenu.c interface.c –llogic\_dynamic –llist\_dynamic**

**export LD\_LIBRARY\_PATH=.**

**./menumain**

*gcc –c –fpic list\_t.c logic.c* compiles the library source code into position-independent code(PIC).

*gcc –shared –o liblist\_dynamic.so list\_t.o*, *gcc –shared –o liblogic\_dynamic.so logic.o* turn the object file into a shared library.

*gcc –L. –o menumain mainmenu.c interface.c –llogic\_dynamic –llist\_dynamic* links the program with shared library.

To run this program, first use *export LD\_LIBRARY\_PATH=.*  specifies the location of the libraries, then use *./menumain* to run.

Difference between library and API:

A library is a collection of reusable codes, while an API is a contract of libraries and etc. without implementation.

The API implemented for the two libraries is in *API\_list* directory.