

The following materials have been collected from the numerous sources such as Stanford CS106 and Harvard CS50 including my own and my students over the years of teaching and experiences of programming. Please help me to keep this tutorial up-to-date by reporting any issues or questions. Please send any comments or criticisms to idebtor@gmail.com. Your assistances and comments will be appreciated.

Using Static Library

A library is a collection of pre-compiled object files that can be linked into your programs via the linker. Examples are the system functions such as `printf()` and `sqrt()`. There are two types of external libraries: *static library* and *shared library*.

- A **static library** has file extension of **".a"** (archive file) in Unixes or **".lib"** (library) in Windows.
- A **shared library** has file extension of **".so"** (shared objects) in Unixes or **".dll"** (dynamic link library) in Windows. Because of the advantage of dynamic linking, GCC, by default, links to the shared library if it is available. You can list the contents of a library via "nm filename".

How to build a static library

Let's suppose that you have a source file (**nowic.cpp**) to turn it into a static library (**libnowic.a**). Assume that you keep **lib/nowic.cpp** and **~/include/nowic.h**.

```
> g++ -c nowic.cpp -o nowic.o -I../include
> ar rcs libnowic.a nowic.o
> ar          // list all the options available
> ar t libnowic.a // list ~.o files archived
> ar x libnowic.a // extract ~.o files archived
> nm nowic.o     // list the actual function names in .o file
```

```
ar flags:
  -r: Insert the files member... into archive (with replacement).
  -c: Indicates you are creating an archive file
  -s: Write an object-file index into the archive, change is made to the archive
  -t: display contents of archive
      (show the list of .o files, use nm ~.o to see functions in ~.o)
```

NOTE: It is important that you recognize that the GCC compiler requires that you prefix your static library with the keyword **lib** and suffix **.a**, like **libnowic.a**. The **lib** prefix is required by the linker to find the static library.

How to reference a static library

Assuming that you are in **~/src** folder and already have **../lib/libnowic.a**

```
> g++ selection.cpp sortDriver.c -o sort -I../include -L../lib -lnowic
```

```
g++ flags:  -L: Indicates the location of the library you want to reference
            -l: Specifies the specific library you want to attach
```