Department of Computer Science
University of Massachusetts Lowell
91.102 Computing II
Fall 2015

Lab 6: makefile (Part 2--Libraries)

In this lab, you will learn how to create a library and use it in your code. The files to be used in this lab may be found under http://www.cs.uml.edu/~kseethar/Fall2015/91.102/labs/lab6

Library

- A library is a collection of .o files (and thus functions) which can be shared by many applications. For example, the linux 'math' library provides support for math functions such as sqrt, sin, etc. All C programs link automatically with the C runtime library--libc—printf is part of this library. Libraries are of two kinds:
  - static: a static library [http://www.tldp.org/HOWTO/Program-Library-HOWTO/static-libraries.html] has an extension of .a (archive) and is created using the ar command. The linker links against the .a file and the object code for the referenced functions becomes part of the executable being linked to—this means the executable is complete and independent of any external references.
  - shared or dynamic: a shared library [http://www.tldp.org/HOWTO/Program-Library-HOWTO/shared-libraries.html] has an extension of .so (shared object) and is created using the gcc command with some special flags. The linker links against the shared lib and the library will be loaded at runtime only if the function defined in it are actually called. So, an executable linking against a shared library will have unresolved references at runtime if it cannot find the .so file.
- Naming convention for libraries:
  - the filename for a library foo is libfoo.a for static library
  - for shared object, there are 2 names: real filename in the format libfoo.so.1.2.3 and linker name in the format libfoo.so
- Using libraries:
  - include the .h file as usual
  - for the link line, use -lfoo, where foo is the name of the library [libfoo.a or libfoo.so]
- Windows has similar concepts—static library has an extension of .lib and the dynamic library has an extension of .dll (Dynamic Linked Library).

Lab6

- This is an extension of Lab 3.
- define rules for creating static and shared libraries
- use these libs to create the executable
- following source files will be used to create the library called 'mymath':
  - my_add.c
  - my_mult.c
- following additional targets should be supported by the makefile—you can replace the existing target for the executable:
  - static_mymathlib: libmymath.a
  - shared_mymathlib: libmymath.so
  - static_lab3: executable lab3 linking against libmymath.a

    - shared_lab3: executable lab3 linking against libmymath.so
- build the executable by executing makefile—make sure you are not directly linking against my_add.o and my_mult.o
- learn about environment variables needed for use with .so file—specifically LD_LIBRARY_PATH.
- What happens when LD_LIBRARY_PATH is not set or does not contain the path for the .so file?
- learn about the various options of ar command through man ar
- use the ar -t command to check the .a file and verify
- learn about the nm [http://www.tldp.org/HOWTO/Program-Library-HOWTO/miscellaneous.html#NM] command through man nm
- use the nm command to check the libmymath.a and libmymath.so file and verify
- check out the libraries on installed on your system—start with /usr/lib and its subdirectories
- as an extra credit, repeat on a Windows system

Deliverables

- a report of the observations
    - static vs shared
    - ar command
    - nm command
    - environment variables
    - libraries on your system
- makefile