Courtney Makua

01/12/2022

CSCI 3731 A

Dr Edward Pier

1. What are libraries good for?

Libraries are good for using code from somewhere else. Instead of linking to possibly hundreds or thousands of .o files you can just link to a library.

2. What are the two types of libraries and how are they different?

Static and shared.

In a static library, its object files become part of the executable, so you can run it without having the library file around.

In a shared library, objects are separate from the executable and the code is loaded by the executable at runtime. A single library can be used by several different executables, so you only need one copy of the library of your system.

3. Why is it better to get a library from a package manager, rather than compile it from source yourself? When might it be better to compile a library yourself?

When compiling from source you might have to customize the way the code is built for your specific system. The project you want might depend on other projects that you would have to install. There might be other programs on your system that are using a different version of the library and having two different versions of the same library installed is not good. If you don't find what you want in package manager and instead find it on GitHub, that is probably when you'll be compiling it yourself.

4. 4. Suppose you are on MacOS and you want to use a library in the file /usr/local/lib/libpng.dylib that has the accompanying header file /usr/local/include/png.h. Suppose also that your program consists of one file called main.cc. Write the Makefile for this project.

```
All: main
```

```
Main.o: main.cc png.h

        g++ -c main.cc -I /usr/local/include/

main: main.o

        g++ -o main main.o -L /usr/local/ libpng.dylib -lfft
```

5. What is a struct?

A class with no methods, and all its data members pubic.