“Check It Out From The Library” Lab

### Introduction

During this lab you will learn:

* How source code programs become running programs, specifically focusing on the compile, link, and load phases.
* The differences between, and consequences of, compile-time, link-time, and load-time binding.

### Starters

Using the NoMachine client, download liblab.zip from the course web site on phoenix, unzip it, and open a terminal in the LibLab folder. Follow the instructions below, answering the questions, which are in italics. Type your answers into the text box following each question. You may work with someone else on this lab, and submit one copy of this, including your answers, on Canvas.

*What are the names of everyone who worked on this lab with you, including yourself?*

|  |
| --- |
|  |

### Linking to a Shared System Library, and Binding at Load-Time

Run the command:

**make test2 2>&1 | less -e**

(Use the 'd' key to move down a half-page, the 'u' key to move up a half-page, and 'q' to quit and get back to a terminal prompt.)

*Referencing the output of the nm and ldd programs, explain why the sqrt function is not initially found when hello2 is linked.*

|  |
| --- |
|  |

*Comparing the ldd output from hello1 to the ldd output from hello2, what additional shared library does hello2 depend upon?*

|  |
| --- |
|  |

### Linking to a User Static Library, and Binding at Link-Time

Run the command:

**make test3 2>&1 | less -e**

(You may find it easier to use the up-arrow key on your keyboard to bring up the previous make command and then edit and run it.)

*The output from nm shows that the functions f1, f2, and f3 are undefined in main.o, but defined in mainst when it is linked to liblabst. Explain why.*

|  |
| --- |
|  |

*What is the size of mainst?*

|  |
| --- |
|  |

*Explain why mainst's output doesn't change when liblabst is rebuilt with a new version of the function f1.*

|  |
| --- |
|  |

*Explain what happens when liblabst is removed, and mainst is run.*

|  |
| --- |
|  |

### 

### Linking to a User Shared Library, and Binding at Load-Time

Run the command:

**make test4 2>&1 | less -e**

*ldd is run twice, once including the location of liblabsh for the loader and once without including the location. What is the difference between the two outputs, and explain the difference.*

|  |
| --- |
|  |

*What is the size of mainsh?*

|  |
| --- |
|  |

*Explain why mainsh's output changes when liblabsh is rebuilt with a new version of the function f1, without having to re-link mainsh to the new liblabsh.*

|  |
| --- |
|  |

*Explain what happens when liblabsh is removed, and mainsh is run. Compare this to the earlier experiment in which liblabst was removed and mainst run.*

|  |
| --- |
|  |

*Taking into account the size difference between mainst and mainsh and their behavior based upon when binding occurs, if you had 4,200 programs that used a library you had written and had to revise periodically, would you use a static or a shared library? Why?*

|  |
| --- |
|  |