Breakout / Lab 05

Preparing for Curses

Last Updated: February 15, 2017

Problem / Exercise

This lab is designed to get you familiar with the following:

- Compiling and installing C/C++ software and libraries that others have written.
- Linking third party libraries with your own code.
- Creating simple Text User Interfaces (TUIs) in C/C++ using the `ncurses` library.

The `ncurses` Library

The `ncurses` (new `curses`) library is a free software emulation of `curses` in System V Release 4.0 (SVr4), and more. It is a programming library that provides an API which allows the programmer to write text-based user interfaces in a terminal-independent manner. It is a toolkit for developing "GUI-like" application software that runs under a terminal emulator. Such user interfaces are generally known as Text User Interfaces (TUIs). It uses `terminfo` format, supports pads and color and multiple highlights and forms characters and function-key mapping.

In mid-June 1995, the maintainer of 4.4BSD `curses` declared that he considered 4.4BSD `curses` obsolete, and encouraged the keepers of Unix releases such as BSD/OS, FreeBSD and NetBSD to switch over to `ncurses`. As of the writing of this document, the latest version of `ncurses` is 6.0, which was released on August 8th, 2015.

1 References

You may find the following reference materials useful:

- `https://www.gnu.org/software/ncurses/`
- `http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/`
- Ch. 5 & 8.4 in Hoover. "System Programming with C and UNIX" (1st Ed.) (ISBN-13: 9780136067122)
- Ch. 3 in Stevens & Rago. "Advanced Programming in the UNIX Environment" (3rd Ed.) (ISBN-13: 9780321637734)

2 Group Brainstorm

You are NOT allowed to use the computers during this time.

Breakup into groups based on your seating and brainstorm about how to solve the problem or exercise. Make sure everyone understands the problem, and sketch out potential ways to move towards a solution. Perhaps something that was discussed during lecture might be useful?

3 Submit Individual Brainstorm

You may use a computer from this point forward.

Login to eLC and submit a version of your group's brainstorm, written in your own words. You may add additional information if you want. You need to write enough in order to convince the grader that you understand the problem or exercise and that you have a plan for moving forward towards a solution. Please include the last names of the other people in your group in your submission. The brainstorm submission should be available on eLC in your assignment dropbox. You should submit your individual brainstorms before the end of your breakout period. **NOTE:** Submissions that do not include an individual brainstorm will not be graded.

4 Worksheet Questions

You should try to answer the questions in this section while you are working on items in the subsequent sections. You may need to lookup some of these questions online. Make sure that all of your files are in a directory called `LastName-FirstName-lab05`, where `LastName` and `FirstName` are replaced with your actual last and first names, respectively. Create a file called `WORKSHEET.md` (plain text). In this file, please provide the answers to the following questions:

1. In (5.3), what is the entire terminal command that you used to download `ncurses`?
2. In (5.4), what do each of the options `zxvf` that were passed to the `tar` utility denote?
3. In (5.6), what is the purpose of executing the `configure` script?
4. In (5.6), what does the `--prefix` option that was passed to the `configure` script denote?
5. In (5.8), where do the compiled and linked files get installed to when "`make install`" is executed?
6. In (6.1), we change the `LD_LIBRARY_PATH` environmental variable. What is this variable generally used for?
7. In (6.1), why does `LD_LIBRARY_PATH` appear twice in the `export` statement?
8. In (6.1), why do we place the `export` statement in your `.bash_login` or `.bash_profile` file?
9. In (7), what do each of the lines of code (A)–(E) do?
10. In (7), why is the `-lncurses` option needed when compiling the sample code?
11. In (7), why is the `-lncurses` option needed when linking the sample code?

5 Compiling & Installing the `ncurses` Library

Follow the instructions presented below in order to compile and install the `ncurses` library to your home directory.

1. Login to your Nike account.
2. Go to `http://ftp.gnu.org/gnu/ncurses/` in your browser and copy the link for the latest version of `ncurses` (e.g., the link to `ncurses-6.0.tar.gz`).
3. Download `ncurses` using the `wget` command and the link you copied in the previous step. There should now be a `.tar.gz` file in the current directory (e.g., `ncurses-6.0.tar.gz`). **PROTIP:** Remember, you can lookup details about the `wget` command in the Linux Manual.
4. Extract the file you downloaded by using the `tar` command along with the `zxvf` command line options. **PROTIP:** You should lookup what each of those options to `tar` do by reading about them in the Linux Manual. For example, you might type the following:

   ```
   $ tar zxvf ncurses-6.0.tar.gz
   ```

   This will create a new sub-directory that contains the extracted files (e.g., the `ncurses-6.0` sub-directory).
5. Change into the sub-directory containing the extracted files.
6. Issue the following commands in order to configure the compilation:

   ```
   $ sed -i '/LIBTOOL_INSTALL/d' c++/Makefile.in
   $ ./configure --prefix=$HOME --with-shared --with-debug --enable-widec
   ```

   The first command is specific to `ncurses`, however the second command is something that you will likely issue before compiling most third-party software and libraries on Linux. I recommend issuing the following command in order to see what different options to the `./configure` script do:

   ```
   $ ./configure --help
   ```

7. If no errors occur, then issue the following command to compile the `ncurses` library:

   ```
   $ make
   ```

8. If no errors occur, then issue the following command to install the `ncurses` library:

   ```
   $ make install
   ```

9. Now change back into your home directory and delete both the `.tar.gz` file that you downloaded and the sub-directory containing the extracted files. You no longer need them.

6 Let the Linker know where to find the ncurses Library

Follow the instructions presented below in order to ensure that the `ncurses` library is added to your linker path.

1. Add the following line to your `.bash_login` or `.bash_profile` file (whichever exists). If you already have `export` statements, you should place this near them.

   ```
   export LD_LIBRARY_PATH=$HOME/lib:$LD_LIBRARY_PATH
   ```

2. Logout of Nike, then log back in.

7 Sample Code

Make sure that all of your files are in a directory called `LastName-FirstName-lab05`, where `LastName` and `FirstName` are replaced with your actual last and first names, respectively.

Follow the instructions presented below in order to test whether or not the `ncurses` library was properly installed.

1. Create a file called `lab05.cpp` and add the following to it:

   ```
   #include <cstdlib>
   #include <ncurses.h>

   int main() {
       initscr();                  // (A)
       printw("Hello World !!!"); // (B)
       refresh();                  // (C)
       getch();                    // (D)
       endwin();                   // (E)
       return EXIT_SUCCESS;
   } // main
   ```

2. Compile the program:

   ```
   $ g++ -Wall -std=c++14 -g -O0 -pedantic-errors -lncurses -c lab05.cpp
   ```

3. Link the program into an executable:

   ```
   $ g++ -Wall -g -lncurses -o lab05 lab05.o
   ```

4. Run the program:

   ```
   $ ./lab05
   ```

5. You will be able to tell if the program is working.

8 Some Nonfunctional Requirements

Your submission needs to satisfy the following nonfunctional requirements:

- **Directory Setup:** Make sure that all of your files are in a directory called `LastName-FirstName-lab05`, where `LastName` and `FirstName` are replaced with your actual last name and first name, respectively.
- **Documentation:** All classes, structs, and functions must be documented using Javadoc (or Doxygen) style comments. Use inline documentation, as needed, to explain ambiguous or tricky parts of your code.
- `Makefile`**:** You need to include a `Makefile`. Your `Makefile` needs to compile and link separately. That is, make sure that your `Makefile` is setup so that your `.cpp` files each compile to individual `.o` files. This is very important. The expectation is that the grader should be able to type `make clean` and `make` to clean and compile/link your submission, respectively.
- **Standards & Flags:** Make sure that when you compile, you pass the following options to `g++` in addition to the `-c` option:

  ```
  -Wall -std=c++14 -g -O0 -pedantic-errors
  ```

  Other compiler/linker options may be needed in addition to the ones mentioned above.
- `README`**:** Make sure to include a `README` file that includes the following information presented in a reasonably formatted way: i) your Name and 810/811; ii) instructions on how to compile and run your program; and iii) reflection section. Make sure that each line in your `README` file does not exceed 80 characters. Do not assume line-wrapping. Please manually insert a line break if a line exceeds 80 characters.
- **Compiler Warnings:** Since you should be compiling with both the `-Wall` and `-pedantic-errors` options, your code is expected to compile without `g++` issuing any warnings.

9 Submission

Before your next breakout lab session, you need to submit your code. You will still be submitting your project via nike. Make sure your work is on `nike.cs.uga.edu` in a directory called `LastName-FirstName-lab05`. From within the parent directory, execute the following command:

```
$ submit LastName-FirstName-lab05cs1730a
```

It is also a good idea to email a copy to yourself. To do this, simply execute the following command, replacing the email address with your email address:

```
$ tar zcvf LastName-FirstName-lab05.tar.gz LastName-FirstName-lab05
$ mutt -s "lab05" -a LastName-FirstName-lab05.tar.gz -- your@email.com < /dev/null
```