Program #2
Creating a Software Release
Due Friday, October 15, 2004

Name: _____

Lab Time: _____

Grade: _____/30

On my honor, as a University of Colorado at Boulder student, I
have neither given nor received unauthorized assistance on this work.
Signature: _____

In this program you will be given the source code of a simple software system, and it is your job to create a "release" of this system that automates the installation process to a significant degree.

The Source Code

Create a directory named `program02` in your `src` directory. The source code for this program is in `~csci3308/src/program02.cc`. Copy this file to your `program02` directory. You may want to compile and run the program just to make sure it works. (The program is written in C++, so you should use the g++ compiler to compile it.)

The Makefile

You must create a makefile for this program. The makefile must have at least two variables, `SRCDIR`, the directory for the source code, and `BINDIR`, the directory in which to install the executable. Assume that the user has the same directory structure as you and set the values of those two variables appropriately. The makefile should look in the directory `SRCDIR` to find any source files. (You should not assume that the makefile will be invoked in `SRCDIR`.) In addition, you are NOT allowed to create a variable that references the architecture-specific build directory for program02. Instead, you will write an install script (see below) that makes sure that this makefile gets invoked in that directory.

The makefile must have at least three targets, `program02`, `install`, and `clean`. If we run `make program02` the makefile should compile `program02.cc` into a program named `program02`. Both the compiled program and any intermediate files should be placed in the current directory because the install script will have changed to the build directory before running make. If we run `make install` the makefile should copy `program02` from the current directory

to `BINDIR`. Also, if we run `make install` and the program hasn't been built the makefile should build the program before installing it. If we run `make clean` it should remove every file that was created by the makefile during the build phase. It should not remove program02's source code and it should not remove an installed version of program02.

The Shell Scripts

You must create a shell script called `program02-install`. This script will be similar to `gnuchess-install` in that it automatically builds and installs program02. You will need to make sure that the architecture-specific directories for program02 are created, so be sure to include and use your `archdir-setup` script with this release. Your makefile uses the view path to look for source files in the correct place, and uses `BINDIR` to install in the correct place, but it creates intermediate files in the current directory, so the script should create and `cd` to the correct build directory before building program02.

Note: Your makefile and install script should be configured such that if you run the install script twice in a row, without calling `make clean` in between, that:

1. the install script does not crash the second time it is run
2. and the program will only be installed in the `bin` directory once.

In other words, do not copy program02 from the `build` directory into the `bin` directory unless its newer than the installed version. Also, note, even though you have been asked to define a target called `clean` in your makefile, your install script should NOT invoke this target. That is the job of the `program02-clean` script.

After you have finished creating and testing your `program02-install` script, create a `program02-clean` script. This script should check to see if the program02 `build` directory exists and if so, it should `cd` to it, invoke `make clean` to clean up the intermediate files created by the `program02-install` script and then delete program02's `build` directory. This script should do nothing to the installed version of program02, nor should it touch program02's source directory.

In addition, this script should be configured such that it can be run multiple times in a row, without crashing. Thus, if you typed the following commands at the prompt (assume that these two scripts are in your path):

```
program02-install; program02-install; program02-clean; program02-clean
```

Then, program02 is compiled and installed only once (even though the install script was invoked twice), the intermediate build files are correctly cleaned up, and the second invocation of program02-clean does not crash.

The README File

You must create a README file with simple instructions on what each file in the distribution is, and how to proceed with installing the program (and

cleaning up after the install). The README file should also have your name and your lab section listed at the top.

Putting it Into a Tar File

You will pack all of this into a tar file so that the user can get all the files at once. You should create a tar file named `program02.tar` that contains a directory named `program02`. In this directory there should be six files: `program02.cc`, `makefile`, `program02-install`, `program02-clean`, `archdir-setup`, and `README`.

Please submit the tar file via the moodle-based website for CSCI 3308. (There will be a program02 assignment link that can be used to submit the tar file. If you have questions about this, send e-mail to Dr. Anderson or the TA.) The tar file must be received by 10 AM on October 15th.

Evaluation

All six files of the tar file will be checked for correctness. (Be sure that your archdir-setup script meets all the requirements defined for it in Lab 3.) A complete and correct program is worth 30 points. For each problem found with your program (for instance, your install script crashes or has a syntax error, or your makefile behaves incorrectly, etc.), five points will be taken off. As such, six or more problems with the program will result in a score of zero points.