CSCI 347 - Computer Systems II

Fall 2022

Assignment 2

Start with the code you turned in for assignment 1. You should have a single file “ush.c” in a “ush” directory which has three primary functions, arg parse(), processline() and main(). (You may have helper functions for arg parse().)

This assignment’s work consists of the following steps:

Part 1: Adding Makefile (20 points)

Add a new file, “Makefile”, that makes the executable “ush” from your ush.c file. Just giving the command “make” in your directory should make the program. You should also have a “clean” target that removes all generated files including the final executable. You may need to read about the “make” program to get this working the way you want it.

The following file is a start:

```
# This is a very simple makefile
#
#CFLAGS=-g -Wall

ush: ush.o
	$(CC) $(CFLAGS) -o ush ush.o
```

The “gcc” line must start with the tab character. As this project continues, you will need to add some new .c or .h files to your project. When you add the .c file to your project, you need to update the Makefile so it compiles your ush from all the .c files. When a .h file is added, you may need to add dependency lines.

When you upgrade your Makefile it must do the following:

- compiles each .c file to a .o file.
- links all .o files together into the ush executable.
- uses make Macros so only one line needs to be changed to add a file that needs to be compiled. A depends line would possibly need to be added or changed to get the dependencies correct.

Also, it would best if your Makefile uses the built-in suffix rules for compiling C programs as is the case with the simple makefile above. (Notice, it does not specify how to make “ush.o”, it just knows how to do it.)