The purpose of this program is to create a small linux shell that can operate a few basic commands. At the beginning of my program I am defining a few basic variables to hold the user input and keep track of the history in an array. I have a flag that determines whether the program should continue to run here as well. I then have a function to define redirect and piping abilities. If the program encounters a “|” or a “<” in the input line they are called. I then define an execute function to consolidate the general process for forking to children. The process will consider if the process is supposed to run concurrently with the “&” symbol. When a child process is executed it will print the child ID. It will also indicate if it is running in the background or foreground. There is error checking to see if the child fails, and a kill process if it does. The parent process can be told whether to wait for the child or continue letting the user place input dependent up the inclusion of the “&” character. I then go to the main() function and create a couple more objects for history and arguments. I print out the prompt, flush the system, and take input from the user and put it into an array. I then use string tokenizer to break down the user input into pieces that I can process. I now do checks to see if the user entered any redirection or piping. Then I do checks to see if the user entered the “&” character or “exit” which will exit the program. I then check for “history”,”!n”, and !!. The history command checks my command count variable I defined at the beginning and indicates if any external command has previously been entered. I don’t count the internal commands in this program in the history because they are not executed from “execvp()”. The history command will print the previous history up to the last 10 commands using a while loop. I used the “%” operator a lot to make sure the right function is executed. The !n command removes “!” form the front and checks to make sure it’s not “!!”. Then it uses “%” to check through the history array and execute the specified command. The !! command just takes the most recent from the history array. If none of the mentioned tokens are found in the user input I just run “execute” without special parameters. Once the command has been run I increment the command count and loop the process.