#### *Group 13*

#### *Charles Alan Macon*

#### *Garrett Wescott*

#### *Jordan Sanders*

#### *Julie Quiroz*

#### *Introduction*

Since its inception SSSH has been a labor of $love for the team. With it, we hope to implement a basic replacement shell for Linux systems that can be used for even the most basic of commands.

Thank you,

-*Group 13*

*[SSSH ] love = frustration*

#### *Usage*

SSSH is a replacement shell for Unix-based operating systems.

In your standard shell, compile SSSH in gcc by typing gcc \*.c in the SSSH temp directory. To run the shell at that point type in the name of the program (if you gave it a specific name) or type ./a.out (by default).

#### *Implemented* *Commands*

* echo()
  + Usage: *echo [STRING(S)] … [$VAR]*
  + Echo echos the STRING(s) to standard output.
  + Echo also supports the use of variables
  + Example:
  + [SSSH ] variable = 4
  + [SSSH ] echo variable equals $variable
  + *Variable equals 4*
* cd()
  + Usage: *cd [SHORT-OPTION]…[DIRECTORY]*
  + *THIS FEATURE IS CURRENTLY NOT IMPLEMENTED.*
* Clear
  + Usage: *clear*
  + Properly clears the screen by outputting a sequence of escape codes.
  + The standard Linux version would cause issues with certain functions.
* SuperBash()
  + Usage: *superBash[INPUT] … [OPTIONAL OUTPUT]*
  + superBash allows the user to translate very basic (and very specific) c programs into a format that is usable by bash.
  + Example:
  + [SSSH ] superBash input frank
  + *Reads in file input and outputs to file frank*
* man()
  + Usage: *man [COMMAND]*
  + Used to display the man page for a given command.
  + Example:
  + [SSSH ] man –XOR
  + *Returns the manual page for the program XOR.*
* cpusage()
  + Usage: *cpusage*
  + Used to get the average CPU Load since SSSH has started running (15 minute intervals over a 24-hour maximum period).
* cpuAverage()
  + INTERNAL FUNCTION
  + Usage: Used by cpusage() to calculate average CPU Load.
* cpuRead()
  + INTERNAL FUNCTION
  + Usage: Used by cpusage() to output average CPU Load.
* cpuReadFloat()
  + INTERNAL FUNTION
  + Usage: Used by cpusage() to output average CPU Load in a format that could be used in I/O redirection.
* strToBinary()
  + Usage: *strToBinary [STRING]*
  + Returns a binary string for each character of STRING
  + Example:
* charToBinary()
  + INTERNAL FUNCTION
  + Usage: Used by strToBinary() to calculate binary value of each character
* xorBinary()
  + Usage: *xor [STRING 1]…[STRING 2]…[STRING 1 LENGTH]…[STRING 2 LENGTH]*
  + XORs two strings character by character.
  + Returns a binary string with a length based on the shorter of the two strings.
  + Example:
  + [SSSH ] XOR frank franz
  + *Returns 0 0 0 0 1*
* Quit()
  + Usage: *quit*
  + Properly exits the shell (and frees all memory) without the use of a CTRL-D key sequence.
* printBinary()
  + INTERNAL FUNCTION
  + Usage: Used by strToBinary() to output binary value of a string

#### *Implemented Features*

* Piping
  + To pipe output from one command to another, input a pipe between the two.
  + Example: ls | wc
  + SSSH supports the piping of two commands.
  + As the only commands that would really use this function are Linux commands, piping using our functions is currently disabled. Any attempt to do so will result in effectively “chaining” commands by running one right after another.
    - This is done because none of our functions (man, cpusage, etc.) have any need for the output of our other functions.
* Background Processes
  + To run a process in the background, input an & as the last character in the command line
  + Example: ls &
* I/O Redirection
  + To redirect input to a function, use < (Example “man < input”).
    - This is only implemented for functions that would make sense.
  + To redirect output from a function, use > (Example “man –man > output”).
  + These commands can be chained together (Example “man < input > output”).

A picture of my cat for good measure

