Using the CS1 Computing Science Linux Server

Capturing Program Interaction (*user input* and *program output*)

Introduction

Capturing the interaction from a CLI (Command Line Interface) program is a useful way of showing that the program works, or obtaining samples of the program running for a manual or user's guide. Whether the program is written in C, C++, Java, etc., UNIX/Linux allows for a simple manner of capturing the program's input and outputs to a file that can be edited and printed.

Method

After verifying that the program works correctly, and sample outputs are ready to be captured, use the **script** and **strings** UNIX/Linux operating system commands to store and clean the interaction.

script – captures all program output and user input that occurs between the **script** and **exit** commands (see example) and sends it to a destination text file; unless a destination filename is provided, the capture is sent to the default file "typescript"

*(**Note**: If more than one run session is required from a program, such as running it three times, capture all the program runs in a single session to one file, rather than 3 separate files.)*

strings – extracts only printable (standard ASCII) characters from a file, which means it can be used to remove the special control and *escape* ASCII characters from a text file

Example

Assume a program executable called **tester.out** has been created, which asks for three values and returns the sum. The interaction is captured to the default file (typescript), followed by the command to clean the output and store it to a final output file called: **output_tester.txt**

*(**Note**: Comments for the example are provided as italics, these are not to be typed.)*

```
[user@server]$script                    start capture to file: typescript
[user@server]$./tester.out              run program
Please enter 3 value: 3 40 2
The sum of 3,40,2 is: 45
[user@server]$./tester.out              run program a second time
Please enter 3 value: 1 2 3
The sum of 1,2,3 is: 6
[user@server]$ exit           stop the capture and close file typescript
[user@server]$ strings typescript > output_tester.txt   clean file
[user@server]$ pico output_tester.txt   clean up formatting in file
```

Programmer's Caution

Having the ability of editing the screen capture file (see the last command in the example), the temptation is to clean up the output to match the intent, not the true output of the program—called "fudging the output."

The clean up of the formatting should not change the actual output of the program. The captured output must exactly match the program that created it—if the output format needs to change, change the program!