Command Name: script

The Unix script command. script is used to take a copy of everything which is output to the terminal and place it in a log file. It should be followed by the name of the file to place the log in, and the exit command should be used to stop logging and close the file.

script is a standard Unix command that records a script of your interaction with the Unix system. Once it's started, it works "in the background", meaning that you continue to work normally, but the script session is dumping everything that shows up on your screen (more or less*) into some file. To start a script session, issue the command ***script*** to the Unix shell; then continue on working normally as long as you like. If you don't provide a file name to the script command, it places its output in a default file named ***typescript***, but for CS125, I recommend you name your script file hwn.txt, where n is the number of the programming assignment you're doing.

Whatever you do, do not use the name of your program's source code file as the filename for the output of the script command. If you type script hello.c the output from the script command will overwrite and destroy whatever used to be in hello.c Anyway, when you decide you don't need to record stuff anymore, exit from the scripting session by issuing the command exit to the Unix shell.

[LINE 1] guest-jBeOIH@geu-SMBIOS:~$ script abc
[Line 2] Script started, file is abc
[Line 3] guest-jBeOIH@geu-SMBIOS:~$ cal
August 2017
Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
guest-jBeOIH@geu-SMBIOS:~$ date
Wed Aug 30 13:03:43 IST 2017
[Line 4] guest-jBeOIH@geu-SMBIOS:~$ exit
[Line 5] exit
[Line 6] Script done, file is abc
[Line 7] guest-jBeOIH@geu-SMBIOS:~$

Line [1]: The % is the standard Unix shell prompt, in response to which I have entered the Unix command script and told it (script) to place all its output in a file named test_script.txt

Line [2]: This line is from the script program telling me that it has started and is recording everything.

Line [3]: This line starts with another standard Unix prompt (%), to which, in this example, I responded by entering the command ls. (When you use script to prepare a file to turn in for CS125, you won't be using ls, just cat and gcc; but Unix and the script command don't care what you do inside a script session.)

Line [4]: This line starts with another Unix prompt, to which I reply exit, meaning "exit from the scripting session". Technically, exit exits the current shell, which you hope is actually a subshell (running script) of your login shell. If you forget where you are (you can be 20 shells deep as far as Linux is concerned; it doesn't care) and inadvertently exit from your login shell, you are logged off the system. What, you say? That's not very user-friendly? Welcome to the real world; Linux is not a toy.

Line [5]: Lines 5 and 6 are the response from script as it exits and stops the scripting session.

Line [6]: This is the standard Unix shell prompt, waiting for me to continue on, but from now on, nothing will be recorded anywhere (since I'm not in a script session anymore)

If I wanted to, at this point (line 8 and forever after) I could use an editor or other Unix tools to take a look at the output file of the script (which I put in test_script.txt, in the example above; but I could have named the output file anything I wanted or left it blank, in which case Unix would have assigned a default name of typescript). In any event, what I would see would be a perfectly ordinary text file, containing, in this particular case, lines 2 through 7, above.