ELEC 377 Operating Systems F22

Documentation Lab 4

Kevin Yu: 20203451

Raatik Sharma: 20120770

Problem Statement

The objective of this lab is to split traces generated by lab2 into separate files to summarize the time span properties for each of the traces using shell script commands.

Step 1

The first part of the script starts off by checking that only one argument is passed in through the terminal by checking the shell variable "$#" which stores the number of arguments passed in by the terminal. After validating the number of arguments, we go and check if the file passed in as an argument is a file and is readable using the "test -r" command which indicates that the file exists and read permissions is granted. Otherwise, the terminal echo's an error message. Next, the Reader traces are extracted from the input file using the command "grep -E". The Reader traces are extracted from the input file using the following regex pattern: ".\[0;[0-9][0-9]mR". This regex pattern isolates the Reader traces in the file by looking for patterns that match the specified pattern, and those traces are written to another file with the same name but with a ".Reader" suffix. The last part in Step 1 includes a similar process where the Printer traces are extracted from the input file and are written to a file with the same name but with a ".Printer" suffix.

Step 2

In Step 2, we start by finding the number of machines. To find the number of machines, the "grep -E" command is used with the following regex pattern ".\[0;[0-9][0-9]m[0-5]" and the traces are extracted from "$1" which is the first argument from the command line. The grep command is then piped using the pipe character '|' to the "cut -b 8" command which selects the 8th byte from each line trace from the grep command. The 8th byte extracted from the "cut -b 8" command extracts the machine number from each line trace retrieved from the grep command. Once the cut command is completed the output is piped again, to the "sort -u" command which sorts the output from the "cut" command in ascending order and only stores unique elements. All three of these commands piped one after the other and then the final result is stored in a variable called "MACHINES". The next step is to use a for loop that loops through the number of unique machine numbers in "MACHINES". Inside the for loop, the "grep -E" command is called again with the following regex expression ".\[0;[0-9][0-9]+m${i}" to extract all of the machine traces in the input file. Then the output from "grep" is outputted into a file with the same name as the input file, but with machine number as the suffix.

Step 3

In step 3, we execute the "head -n 1 $1.Reader" command which outputs the first line from the ".Reader" file generated in Step 1. This output is then piped to the "cut -b 11-20" command which extracts the 11-20 bytes from the output from first command. By cutting the 11-20 bytes we extract the seconds for the specified time stamp of the reader and store it into a variable called "headVal1". Next we run the exact same command but we cut 22-27 bytes to extract the milliseconds from the time stamp and store that in a variable called "headVal2". These two commands above are then repeated, but the "tail" command is used instead of "head" and the seconds are milliseconds are stored in "tailVal1" and tailVal2" respectively. After those values are stored in the variables, the "headVal1" and "headVal2" are concatenated into "head" and"tailVal1" and tailVal2" are concatenated into "tail".

Afterwards, a printf command is called (which works similarly to printf in C), which does the calculations to subtract the “head” value from “tail” value and then raise it to the 10^6 power to get the time difference and display it to the terminal. Lastly, the printf output is appended to the end of the “.Reader” file. This whole process is also repeated for the “.Printer” file.