Project 2

Amauri Lopez

Darrien Pinkman

Operating Systems (CSC 345-02)

The College of New Jersey

Report Due: 3/11/18

In project 2, all of the required tasked are held within their own functions. The user may call and execute these functions separately by entering “single” and then the corresponding task they would like to execute through the command line. In task 1 the function goes through and tests the strings within the files. The numbers are stripped from the original string and then compared again. If the pointer match then the string is a non-numerical string literal and is added to an array of words. A series of nested loops compares each element in the array of words with the every other element in the array to detect unique words. When one is found, the counter is incremented, giving a number for the amount of unique non-numeric string literals. In task 2 the tokens in each file are split and converted to reach the pure number portion of the string. This is stored into a buffer array and then the pointer to the token is compared to the buffer. Once we see that the token is not in the buffer array, the function gets the length of the word and uses a series of if else statements to find the max and min lengths of the alphanumeric string. These lengths are then stored in a word lengths array. The function keeps a running sum of all the lengths and then divides by the number of total words to reach the average value. When attempting to get the variance the function subtracts the average from each datapoint and squares the value. It then adds then squared values and divides by the total number of datapoints-1. Task 3 has multiple count variables allowing the function to keep track of the min, max, and average values, being int, int, and double respectively. In order to find the number of columns, we will add to the amount when a ‘,’ is detected. And to find the rows, the function will detect everytime fgets is used and will increase the counter counting the numbers of rows, which is counting the number of lines read in. The function then uses a series of if else statements in order to come up with the min and max values, the running sum of the rows and columns are kept and divided by ‘filecount’ to receive the average.The ratio is calculated by multiplying the rows and columns together, subtracting the non zero number count and

multiplying it by 100, then dividing it by the rows multiplied by the columns and making sure the whole thing is casted to a double.

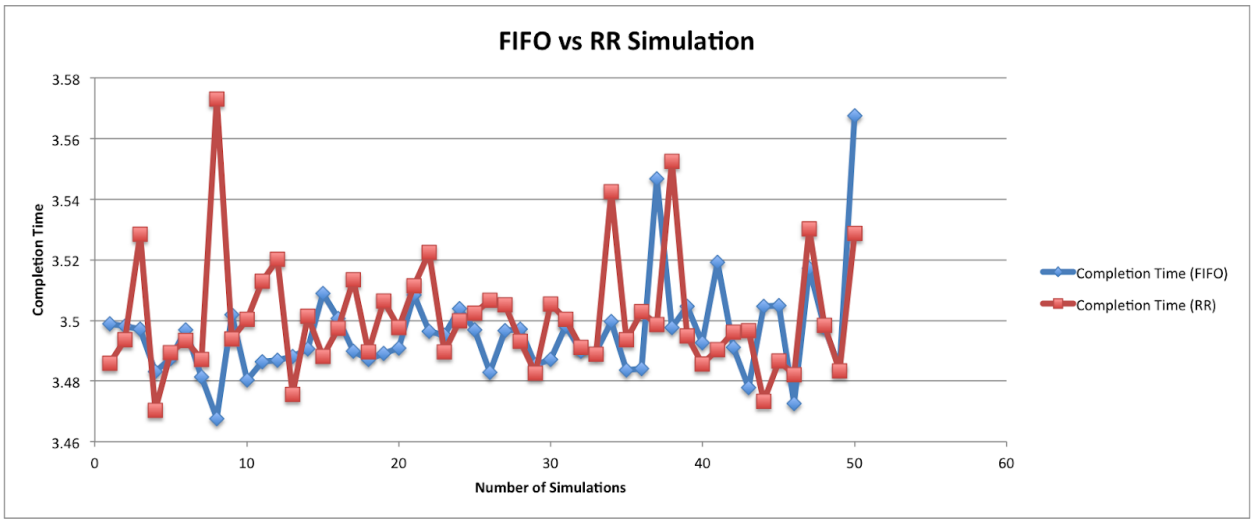

Figure 1: FIFO vs RR Simulation Test Runs

	Min Completion Time	Max Completion Time	Avg Completion Time
RR	3.4704	3.573	3.501056
FIFO	3.4677	3.5675	3.495684

Figure 2: MIN, MAX, AVG Completion for FIFO and RR

Above is a statistical analysis experiment that compares Round Robin and First-In First-Out. The first figure shows the graphical representation of the 50 individual simulations that were run, while the table above shows the minimum, maximum, and average values for both Round Robin and First-In First-Out. In the 50 individual runs each of the 3 tasks were completed and the total completion time was recorded. On average the FIFO scheduling policy recorded a faster completion time, about .005372 seconds. Round Robin can at times be faster than FIFO but as time and simulations increase the difference between the two completion times can be noticed.