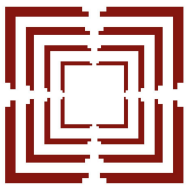


Ahmedabad University

CSE332 Operating Systems

Midsem Exam (Practical)

Section 1

Submitted to faculty: Dhruvil Dave

Date of Submission: Oct 14, 2021

Student Details

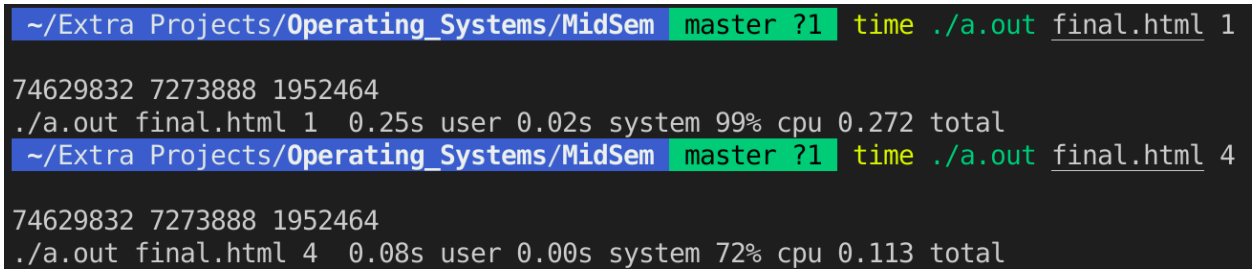
Roll No.	Name of the Student	Name of the Program
AU1940049	Sameep Vani	BTech CSE

2021-2022 (Monsoon Semester)

Code

[Link]

Output Image



Observations

The time and CPU usage reduces drastically when parallelly processes are running. This is an obvious result because now instead of one process, there are 4 processes which are calculating the values through different sections of the file. This happens because the code works in such a way that it first gets the character count through fseek and ftell and then we get the file length per process which creates sections in the test file.

How the code works

The user inputs 2 arguments fromt the terminal and is used by the main function using argc and argv[]. We need to calculate how much length is to be given to each child process. For that, I calculate the total file length using ftell() and then divide the output by the total number of child processes.

Using the fork() system call, I further create the child processes. There is appropriate opening and closing of the file and the pipe files. Pipe here helps to set up the communication between the parent and the child process.

The child process has process id==0. Thus we check for this condition and then implement the logic of WC within it. We store the values of the variables word, line and character count in an object of struct which is globally declared

This data is then written into the pipe by the child process. We use wait(NULL) to wait for all child processes to terminate. Later the parent function reads all the data from the pipe and then calculates the total of all the required variables.

The answer is later printed along with the time function to compare the time taken by and all the errors are checked.

References

1. ANSI C by E Balagursamy
2. https://www.geeksforgeeks.org/create-n-child-process-parent-process-using-fork-c/
3. https://www.geeksforgeeks.org/fseek-in-c-with-example/