COMP301 - OPERATING SYSTEMS

Fall-2023
ASSIGNMENT - 1
DUE: 16th October, 11:59PM

Task 1

Complete the following on your personal computer/laptop. Both options assume that you are initially using Microsoft Windows PC (although it can also be done on Mac). Warning: Dual boot for Ubuntu is not recommended unless it is a new computer, but if you still want to do it, backup your data first. However, the following method is recommended.

Ubuntu in virtual machine

1. Enable Virtual Machine support in BIOS setup.
2. Download latest Ubuntu iso file from: https://www.ubuntu.com/download/desktop
3. Download Virtual Box "Windows Hosts" from: https://www.virtualbox.org/wiki/Downloads
4. Install Virtual Box and run it.
5. Create a new Linux virtual machine, with default options.
6. Run this VM and provide iso file path.
7. Install Ubuntu on virtual drive.

Note: Submission of this task is not required.

Task 2

- Write a C program that prompts the user to input a series of ASCII characters separated by space or commas. The program then separates the series into individual numbers.
- It then prints the respective word that is made using the ASCII codes. For example, if the user enters the numbers “65 108 111 104 111 109 111 114 97”, then the output should be the word “Alohomora”.
- Name your file with your roll number. For example, your roll number is 12345 then your file would be “12345.c”. Save the file on Desktop.
- Make sure the program runs without errors and warnings.

(contd.)

Task 3

- Write a C program that performs parallel text processing using the fork() system call. The program will read a text file and the number of parts divide the file into specified parts, and process each part in parallel by creating child processes. The parent process will then aggregate (collect and combine) the results from all child processes and present a final output.
- Program inputs:
  - Provide a text file (input.txt) containing a substantial amount of text data for processing. We have provided a sample file for that purpose.
  - The number of child processes (num_processes) to be created for parallel processing.
- Program Details:
  - File Division:
    - The parent process should divide the text file into equal parts based on the number of child processes specified.
    - Ensure that the file is roughly divided equally among all child processes so that all of them handle similar amount of work.
  - Parallel Processing:
    - Each child process should process its assigned portion of the file in parallel.
    - Processing task is only to count the number of Words i.e. word count
  - Aggregation:
    - After all child processes have finished processing their portions, the parent process should aggregate the results (i.e. total words) obtained from each child process.
  - Output:
    - Display the results of each child process as well as the aggregated results.
- Implementation Guidelines:
  - Utilize the $fork()$ system call to create child processes.
  - Utilize inter-process communication (e.g., pipes) to communicate between the parent and child processes.
  - Implement error handling and appropriate cleanup of resources.

- Note: Make sure there is proper division of the file among child processes and parallel processing is correctly implemented. Also that the aggregation of results is accurate. Perform proper error handling and cleanup of resources.
- Hint: To collect the results from the child processes, you can use status signal of the wait system call, or have the child processes write to a file and the parent to read from it.
- You are not allowed to use Shared memory segments or pipes.

SUBMISSION DETAILS:

1. Upload C files named as *YourRollNumberTask1.c*, *YourRollNumberTask2.c* etc.
2. Also Upload the Screenshots of outputs.

3. Always write the following at the start of code in comments:
    - Course Code + Course Name + Section,
    - Your Name,
    - Roll Number
    - Date of Lab
4. Upload the files to Moodle.