
CS344: Operating Systems Lab

Lab # 01 (1 Questions, 60 Points)

Held on 22-Aug-2023

Lab Timings: 09:00 to 12:00 Hours Pages: 2

Submission: 12:00 Hrs, 22-Aug-2023

Instructor Dr. V. Vijaya saradhi

Head TAs Adithya Moorthy & Laxita Agrawal

Department of CSE, IIT Guwahati

- a. This assignment is based on chapter 3, Process Management in the book Operating System Principles, Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne.
- b. In order to perform this assignment, understanding of system calls `fork()`, `wait()`, `exit()` is essential.
- c. Example program in chapter 3 covers `fork`, `wait`, `waitpid`, `getpid`, & `exit`.
- d. Carefully read the manual pages for the above system calls.

Question 1: (60 points)

Process creation: Implement the following:

- a. (10 marks) **Process creation** Write a C program using the `fork()` system call that generates sequence of numbers in the child process as per the following equation

$$n = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 3 \times n + 1 & \text{if } n \text{ is odd} \end{cases} \quad (1)$$

The input n will be given through command line argument.

Ensure that parent waits till the child program completes its execution. Terminate parent after child successfully complete the its work.

- b. (10 marks) **Resource sharing - file - between parent and child** Write a second C program identical to the above description. In this program, open a file in append mode before performing `fork()`. In the child process, write all the sequence of numbers generated in this file with a prefix "Child PID:". At the end of writing the sequence of numbers, print the file position indicator using `ftell`.
In the main program, write the parent PID, get the file position indicator.
- c. (10 marks) **Creating more than one child:** Write a third C program that generates 50 children. Each child would generate a sequence of numbers as per the equation (1). The input for the first child is $1 \times n$, second child is $2 \times n$, third child is $3 \times n, \dots$, 50^{th} child is $50 \times n$. Output should be written into a file by each child: as follows:

Child <PID>: Sequence
Child <PID>: Sequence
...
Child <PID>: Sequence

Order of the above lines in the file is not relevant. Parent must wait for all the children to complete execution and then parent should exit normally.

- d. (10 marks) **Parent waiting for a specific child to complete the task.** Write a fourth C program similar to above with the difference that parent should wait only till 1st, 3rd, 5th, 7th and 9th children completed execution. Once these children completes their execution, the parent must exit without waiting for other children.
- e. (10 marks) **One Zombie process:** Write a fifth C program identical to the above description such that it leaves one zombie process in the operating system. The zombie process must remain in the system for at least 10 seconds.
- f. (10 marks) **Several Zombie processes:** Write a sixth C program identical to the above description such that it leaves any 40 processes as zombie process. Each of the zombie process must remain in the system for at least 10 seconds.