

CSE 4502 (SWE) [Operating Systems Lab]

Lab # 08

Lab Task:

1. Write a C program using the `fork ()` to create eight (8) child processes. In each process, print their process IDs and their parents' process IDs.
2. Write a C program using the `fork ()` to create a child's processes. Take a variable with an initial zero (0) value. Add your student ID to that variable in the child process and print that value. Add the last three (3) digits of your student ID to that variable in the parent process and print that value.
3. Write a C program using the `fork ()` system call that generates the Fibonacci sequence in the child process. For example, if 5 is provided as input, the first five numbers in the Fibonacci sequence will be output by the child process. Because the parent and child processes have their own copies of the data, it will be necessary for the child to output the sequence. Have the parent invoke the `wait ()` call to wait for the child process to complete before exiting the program. Perform necessary error checking to ensure that a non-negative number is passed on the command line.
4. Write a C program that takes a list of command-line arguments. The program should fork a child process. The child process will only print 'Hi, I am a Child Process'. The parent process should wait for the child to complete and then print a message indicating the completion of the child process. After the completion of child process, the parent process should use one of the `exec` family of system calls (e.g., `execl`, `execv`, `execle`, `execve`, etc.) to execute a different program specified by the first command-line argument.

For example, if the program is executed with the command:

```
./myexec ls -l
```

The parent process should use `exec` to run the `ls` command with the specified options, and before that it should wait for the child process to finish.