CSE 380: Introduction to Operating Systems

Homework 1: Process Creation using Fork

Due: Friday, 19 September 2003

This assignment requires you to experiment with the creation, termination and handling of processes in UNIX by writing small C programs. First, let's make sure we can write a program using calls such as `fork` and `exec`, and get it to compile and run. Write a program X that creates two child processes Y and Z, and write a program A that creates a child process B such that the child B loads and executes the program X. To test such a program, you should make each of the processes to do some I/O (e.g. write something to the output, or wait for some input) at appropriate places. For each of the questions below, design a test experiment by modifying the programs you have written so that the result of test justifies the answer to the question.

1. Are any of the variables of a parent shared with any of its child processes? Do the children of the same parent share some variables among themselves?

2. Are file descriptors of a parent inherited by child processes? In particular, if a file F is opened by a parent and kept open across a call to `fork`, is F still open in child process? If yes, when a child process reads blocks from F, will the parent remain at the same position in F?

3. Recall the `exit` and `wait` calls that can be used for communication between a parent and its children. Modify the program so that X terminates only after both its children have terminated.

4. Is it allowed for a parent to terminate before one of its child processes terminates? If so, does it affect the children in any way?

5. Is it possible for X to terminate its child process, say Y? Is it possible for A to terminate its grandchild Y? How?
   Hint: Using `exit()` to pass data between child and parent will not work in this case; for details about why, see `man wait`.

For each question, first describe succinctly what you expect and how you have designed the test. Then, include the code. Your code should be well-documented, including the explanations to any sections which might be unclear. Finally, include the output of the test, and explain how it justifies the answer.

Refer to the man pages for more information on UNIX system calls, and also make use of the class newsgroup to post any questions regarding this assignment. Before submitting your code, verify that it compiles and runs as you expect on eniac.

Submission. Submit code and documentation using the **turnin** command to the account **cse380@eniac** by 11:59:59 pm on September 19. Information regarding use of the turnin command has been posted on the cse380 newsgroup.

Reminder. Submitting a solution that is not done by you (such as a copied solution from a fellow student, a website, or some other source) is a violation of the University policies on academic integrity.