Jacob Reiser 810-687-011

Assigment 1 for UGA CS4730 Operating Systems

Convert matrix multiplier to use multiple threads to execute multiplication.

The matrix multiplication multi-threaded method is accomplished by taking the m threads sent to parallel_mat_mult and passing each to the child_process_core onceeach thread is confirmed created by the fork process.

Inside of child_process_core, each thread, which is responsible for one row, is passed through linear_mult p times, and each time the resulting value is written to the pipe.

As values are written to the pipe, they are read by the parent process and placed in their correct place in C_parallel. This is how the main multiplication function is handled.

At the time of submission, it seems like my process is able to work correctly as long as m < 10, n < 10, and p < 10. For some reason it stops working past that point.

The wait and crash handling is accomplished by using the infamous goto. I know there was probably a better way to do this but I had to turn to this dark magic in a crunch. If a child process succeeds while the parent is waiting, the process continues normally and adds the child's values to C_parallel. However, if a thread crashes, it enters a loop that first closes the pipe, and then stays within the loop, and jumps back to just before the current thread was created. It is re-piped, re-forked, and the loop is attempted again with the same i value. This handles the recursive issue, as each thread that crashes is retried until it succeeds.