CSCE 351 Operating System Kernels

Fall 2001
Steve Goddard

Team Programming Assignment 3 (PA3), November 12

Due: 6:00pm November 29

Messages, Messages, everywhere!

In this assignment each team will modify the MINIX kernel to keep track of the number of messages sent from process (or task) `i` to process (or task) `j`. Decide what you should do with messages generated by interrupts. That is, decide whether messages sent from interrupts to device drivers (I/O tasks) should be counted in this matrix and then implement your design. Justify your decision.

Print a matrix displaying the message passing count between all processes/tasks (and interrupts if appropriate) when the F4 key is hit.

Does the matrix printed out show all messages sent in the system? If not, which messages are not counted.

How many messages are sent for a file read less than or equal to one block? How many messages are sent for a file read of `k` blocks? Does the storage device affect the number of messages sent?

Grading Policy for Programs

The programs you hand in should work correctly and be documented. When you hand in your programming assignment, you should include:

1. A program listing containing in-line documentation clearly identifying what changes you made. Only turn in files that you changed (or created).

2. A separate (typed) document of approximately two pages describing the overall program design, a verbal description of "how it work,s" and design tradeoffs considered and made (if any). Also include answers to the questions asked above.

3. A separate description of the tests you ran on your program to convince yourself that it is indeed correct and how you timed the changes in performance. Also describe any cases for which your program is known not to work correctly.

4. A make file that compiles your program(s), if any changes were required with the original make file.

Please hand in your source file(s), the Makefile, and your documentation using the Web handin program. If your program is handed in after 6:00 pm on November 21 it will considered late.

The program should be neatly formatted (*i.e.*, easy to read) and structured and documented. Use the handin program to submit your program(s) for grading. This is assignment 3. Your grade will be determined as follows:

```
Program Listing
      works correctly 40%
      in-line documentation 15%
      quality of design/readability 25%
Design Document 15%
Thoroughness of test cases 05%
```