LOYOLA UNIVERSITY CHICAGO
COMP 410-001 - OPERATING SYSTEMS - FALL 2014

Tiago de Almeida - tdealmeida@luc.edu
1394611

Homework #2

Writing to and reading from a file is done through system calls. The process lets the operating system know it wants to read/write a specific amount of data from/to a specific file. This is done calling the read/write system call. The process then waits until the operating system returns the call saying it has read/written the data. System calls have an overhead related to managing the reading and writing process to memory. This experiment writes data to a file in different combinations so that we can notice the overhead of write system calls.

The program implemented for this experiment writes 256MB of data to a file through varying buffer sizes. The buffer size starts as 1 byte and then is doubled for each run until it reaches 128MB. From the measures we can notice that the overhead of the system call is really significant when we run 2^28 system calls, each asking the system to write one byte to the file. It takes in average 238 seconds to complete the job. Once we run 2^27 system calls, each writing 2 bytes, the job is done in 123 seconds, in average. The time keeps getting shorter as we increase the buffer size and decrease the number of system calls, reaching less than one second to complete the job for buffers of 4MB and bigger. There is actually an interesting drop in needed time when we go from a 2MB buffer to a 4MB buffer, going from an average of 8.3 seconds to less than 1 second. The significance of the difference, however, gets lower as we move to bigger buffers. From buffers of 4MB to 128MB the measured time doesn’t show a significant difference.

When running thousands of systems calls, their overhead becomes so significant that most of the time necessary to complete the job is spent on system calls. Once we begin sending bigger buffers to write, the system calls overhead becomes insignificant and doesn’t influence in the total time of the job.