Test Results Erdem Ege Maraşlı - 21602156

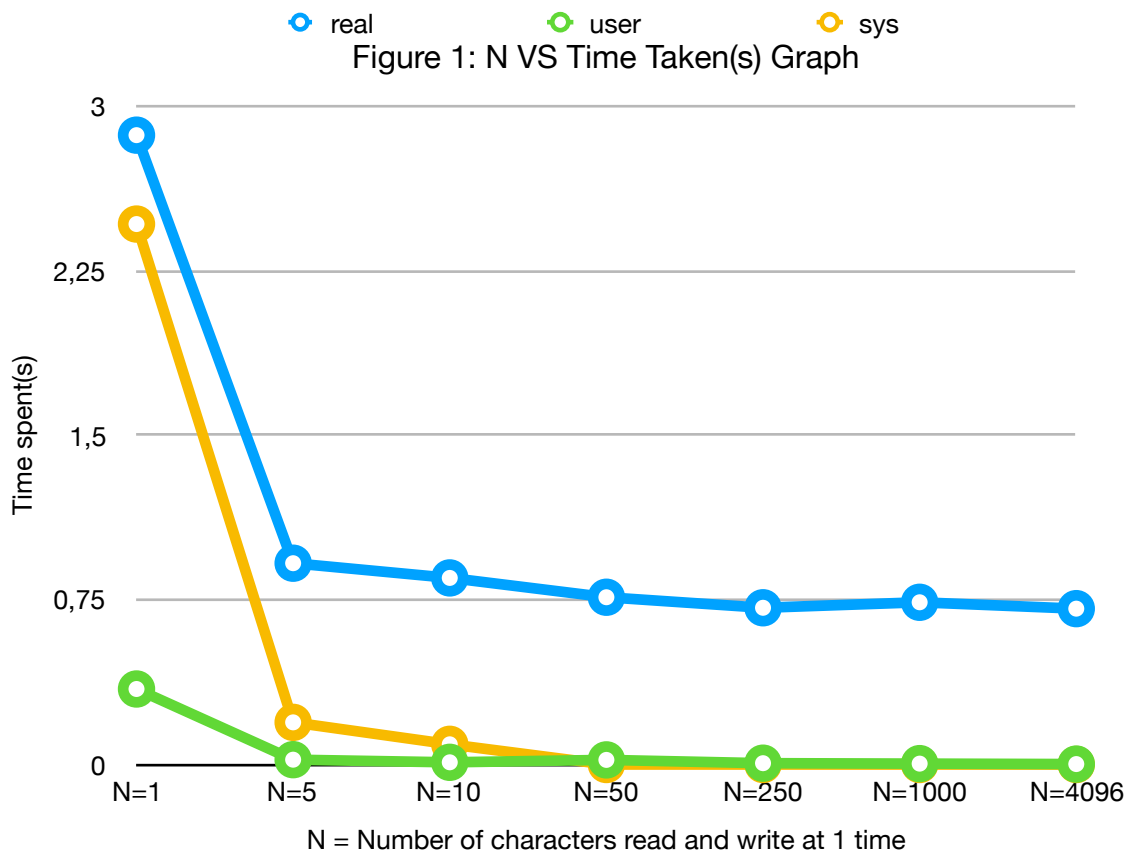

Figure 1: N VS Time Taken(s) Graph

Figure 2: N VS Time Taken(s) Table

N VS time spent(s)	Real	User	Sys
N=1	2,867	0,344	2,462
N=5	0,917	0,22	0,192
N=10	0,850	0,010	0,092
N=50	0,762	0,021	0
N=250	0,714	0,006	0
N=1000	0,739	0,004	0
N=4096	0,710	0,002	0

I do experiment on my code with the given N numbers at the Figure 2. 1000000 character produced and consumed by producer and consumer at each N number. At the beginning of my experiment I expect to see that while N number is getting larger, time spent for all real, user and sys parameters to be smaller. While N is really small(such as 1, 5 and 10) we can observe that time spent is getting smaller, however, with larger N values

time spent is not continue to change significantly. For example, After N = 50 and for larger values of N sys time spent shows 0. Sys refers to the amount of CPU time spent in the kernel within the process. At each time a system call is called kernel takes control of the CPU. In our experiment while N = 1, since we need to transfer 1000000 bytes of information between pipes, 1000000 write() and 1000000 read() system calls are called. While N = 250, 4000 write() and read() system calls are called and this number is getting really smaller with larger N values. 4000 write() and read() system call is not a huge value for CPU. It takes really little time to execute so it is measured as 0. With N gets larger again it does not make a really significant effect on CPU time spent in the kernel. Real refers to time from start to finish of the program. While N = 1, it takes too much time because 1000000 bytes of data is transferring 1 byte per time. But again with larger N values time taken does not change significantly because CPU's are really fast nowadays. User refers to the amount of CPU time spent outside of kernel. While N = 1, user time spent is huge when we compare with the N = 4096. This is caused by the calculation of loops, since while N = 1 loop will be executed 1000000 times to transfer data. For the conclusion, we saw that time spent is generally gets smaller with the less system calls, however, after some point it does not effect significantly.

Producer

```
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    int m = atoi(argv[1]);
    int i;
    for(i = 0; i < m; i++)
    {
        write(1, "a", 1);
    }
    return 0;
}
```

Consumer

```
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    int m = atoi(argv[1]);
    int i;
    char buff[2];
    for(i = 0; i < m; i++)
    {
        read(0, buff, 1);
    }
    return 0;
}
```