

N	Real Time (s)	User Time (s)	Sys Time (s)
5	2.209	0.429	1.761
10	2.040	0.369	1.623
50	1.891	0.422	1.464
100	1.875	0.359	1.512
500	1.887	0.340	1.529
1000	1.870	0.387	1.423
5000	1.830	0.401	1.417
10000	1.832	0.350	1.473
50000	1.834	0.358	1.471
100000	1.854	0.398	1.445

The results show that it is difficult for the system to execute the process with small amount of Ns. It is reasonable because the number of read and write system calls are more than others. As the N value gets bigger and bigger, the execution time doesn't vary very much because the size of the read at once also gets bigger so its time neutralizes the small amount time that is spent for read and write calls.

Producer.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>

int main(int argc, char** argv)
{
    if (argc != 2)
    {
        return 1;
    }
    char* alphnums = "abcdefghijklmnopqrstuvwxyz0123456789";
    int m = atoi(argv[1]);

    srand(time(NULL));
    int index;
    for (int i = 0; i < m; i++)
    {
        index = rand() % strlen(alphnums);
        char chosen[2] = "\0";
        chosen[0] = alphnums[index];
        write(1, chosen, strlen(chosen));
    }
    return 0;
}
```

Consumer.c

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

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

IMPORTANT NOTE

Due to my last time changes, the consumer and producer calls may not work. I had made some changes at last minutes and this could have broken something.