Lauren Fromm
404751250

Lab 5 Log

First, I need to create the tr2b.c file

```
$ touch tr2b.c
```

Inside this file, I first check that there is a correct amount of arguments inputted (3). Then, I make sure that the 'from' argument and the 'to' argument are the same length. Then, I need to make sure that none of the characters in the 'from' argument are repeated, so I go through every character and check that it has not already been listed. After all my checks, I can go through every character of the input file using getchar(), and if it matches one of the characters in the 'from' argument, I replace it with the character that it correlates to in the 'to' argument. I output the everything with putchar().

Next, I create my tr2u.c file:

```
$ touch tr2u.c
```

I use the same logic when creating this file, except I need to use read and write. I go through every character of the input file using a while loop that reads a character every time. When I go through every character of the input file, I put each character in an array and then test it to see if it needs to be replaced. If it does neeed to be replaced, I use write to write it out to output.

To get a big file, I use:

```
$ head --bytes=5000000 /dev/urandom > output.txt
```

which puts a 5,000,000 byte file into output.txt

I want to use the strace command to find out what system call each file uses. I use the -c flag to show time, calls, and errors for each system call and report a summary on program exit. I first copy one file to another by outputting the final result into files called outputtr2u and outputtr2b.

My strace command then looks like:

```
$ strace -c ./tr2b 'ab' 'yz' < output.txt > outputtr2b
```

which outputs:

```
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0         2           read
  0.00    0.000000           0         1           write
  0.00    0.000000           0         2           open
  0.00    0.000000           0         2           close
  0.00    0.000000           0         4           fstat
  0.00    0.000000           0         9           mmap
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         1           brk
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                    29         1 total
```

I then run the command with the unbuffered file:

```
$ strace -c ./tr2u 'ab'   'yz' < output.txt > outputtr2u
```

which outputs:

```
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 54.96   21.183129           4   5000000           write
 45.03   17.356050           3   5000002           read
  0.00    0.000059           8         7           mmap
  0.00    0.000027           7         4           mprotect
  0.00    0.000014           7         2           open
  0.00    0.000010          10         1           munmap
  0.00    0.000007           4         2           close
  0.00    0.000007           7         1         1 access
  0.00    0.000007           7         1           execve
  0.00    0.000006           3         2           fstat
  0.00    0.000004           4         1           brk
  0.00    0.000003           3         1           arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00   38.539323              10000024         1 total
```

The unbuffered version that uses read and write has a lot more system calls than the buffered putchar and getchar file. Because of this, tr2u uses a lot more time.

If i want the output to be directly copied into terminal, I delete the > outputtr2b and > outputtr2u:

```
$ strace -c ./tr2b 'ab' 'yz' < output.txt
```

Which gives:

```
 time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0         2           read
  0.00    0.000000           0         2           write
  0.00    0.000000           0         2           open
  0.00    0.000000           0         2           close
  0.00    0.000000           0         4           fstat
  0.00    0.000000           0         9           mmap
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         1           brk
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                    30         1 total
```

Then I do the same for the tr2u:

```
$ strace -c ./tr2u 'ab'    'yz' < output.txt
```

There is a very long list of random characters outputted and then the final summary:

```
 time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 54.41   22.161857           4   5000000           write
 45.59   18.569249           4   5000002           read
  0.00    0.000000           0         2           open
  0.00    0.000000           0         2           close
  0.00    0.000000           0         2           fstat
  0.00    0.000000           0         7           mmap
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         1           brk
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00   40.731106              10000024         1 total
```

Again, the unbuffered file has a significantly more amount of system calls than the buffered file. The reason for this is that the unbuffered file is making a system call everytime it has to read or write a byte, and since the file is so huge, theres an obvious discrepency between the two files.

I then use the time command to see the different times each file has.

```
$ time ./tr2b 'ab' 'yz' < output.txt
```

which outputs:

```
real  0m0.002s
user  0m0.000s
sys   0m0.001s
```

```
$time ./tr2u 'ab' 'yz' < output.txt
```

which outputs a long list of random characters and then:

```
real  0m5.167s
user  0m0.317s
sys   0m4.814s
```

The unbuffered version again takes a lot longer than the buffered version to execute.