CS35L – Winter ‘19

Slide set:	5.2
Slide topics:	System call programming
Assignment:	5

Laboratory

- Write `tr2b` and `tr2u` programs in 'C' that transliterate bytes. They take two arguments *from* and *to*. The programs will transliterate every byte in *from* to the corresponding byte in *to*

```
./tr2b 'abcd' 'wxyz' < bigfile.txt
./tr2b 'mno' 'pqr' < bigfile.txt
```

- `tr2u` uses **getchar** and **putchar** to read from STDIN and write to STDOUT.
- `tr2b` uses **read** and **write** to read and write each byte, instead of using getchar and putchar. The nbyte argument should be 1 so it reads/writes a single byte at a time.
- Test it on a big file with 5,000,000 bytes

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

time and strace

- **`time`** `[`*`options`*`]` *`command`* `[`*`arguments...`*`]`
- Output:
  - `real 0m4.866s`: elapsed time as read from a wall clock
  - `user 0m0.001s`: the CPU time used by your process
  - `sys 0m0.021s`: the CPU time used by the system on behalf of your process (to make system calls etc)
- **`strace`**: intercepts and prints out system calls to stderr or to an output file
  - `$ strace -c ./tr2b ‘AB’ ‘XY’ < input.txt`
  - `$ strace -c ./tr2u ‘AB’ ‘XY’ < input.txt`

Homework 5 (sfrobu.c)

- Rewrite `sfrob` using system calls (`sfrobu`)
- `sfrobu` should behave like `sfrob` except:
  - If stdin is a regular file, it should initially allocate enough memory to hold all data in the file all at once
  - `-f` option, your program should ignore case while sorting (use the standard `toupper` function to upper-case each byte after decrypting and before comparing the byte)
- Functions you'll need: `read`, `write`, and `fstat` (read the man pages)
- Measure differences in performance between `sfrob` and `sfrobu` using the `time` command

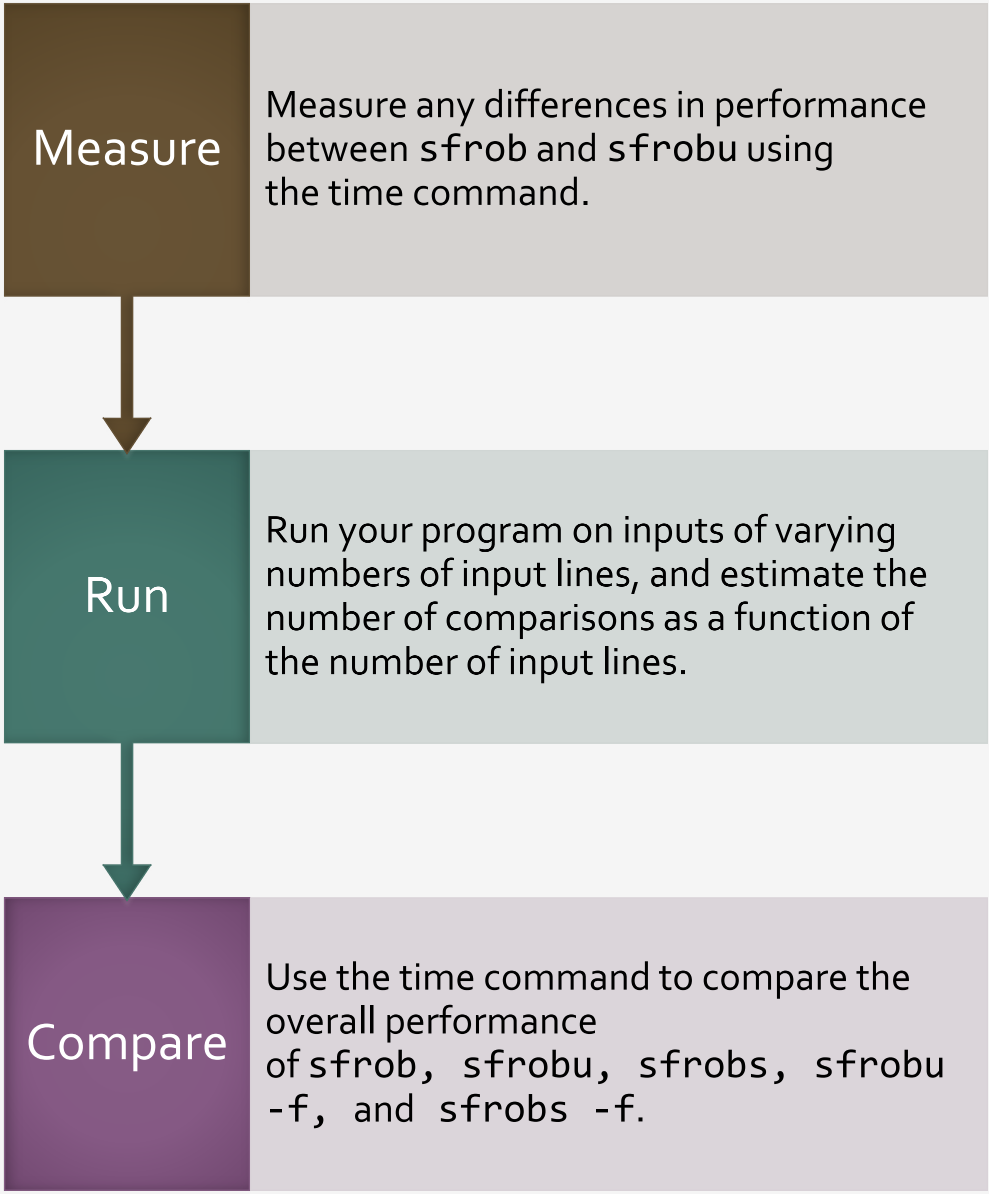
Homework 5(sfrobs)

Write a shell script "`sfrobs`" that uses `tr` and the `sort` utility to perform the same overall operation as `sfrobu` (support –f option as well)

Use pipelines (create no temporary files)

Encrypted input -> `tr` (decrypt) -> `sort` (sort decrypted text) -> `tr` (encrypt) -> encrypted output

Homework 5(sfrob.txt)



Measure

Measure any differences in performance between `sfrob` and `sfrobu` using the time command.

Run

Run your program on inputs of varying numbers of input lines, and estimate the number of comparisons as a function of the number of input lines.

Compare

Use the time command to compare the overall performance of `sfrob, sfrobu, sfrobs, sfrobu -f, and sfrobs -f`.

01

Run your program on inputs of varying numbers of input lines, and estimate the number of comparisons as a function of the number of input lines.

02

Varying number of input lines => number of words

03

Number of comparisons => keep a counter in the `frobcmp()` function to check how many times it is being called

Homework 5