System Call Programming

CS 35L
Winter 2018 - Lab 8

`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
- **strace**: intercepts and prints out system calls to stderr or to an output file
  - $ strace –o strace_output ./tr2b ‘AB’ ‘XY’ < input.txt
  - $ strace –o strace_output2 ./tr2u ‘AB’ ‘XY’ < input.txt

Homework 5

- Recall Homework 5!
- 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
  - You can estimate the number of comparisons performed by counting in code (but the estimate required for the assignment will have to be a function, hint: read the documentation for qsort).
- Functions you'll need: `read`, `write`, and `fstat` (read the man pages, e.g. `man -S 2 read`)

Homework 5

- Measure differences in performance between `sfrob` and `sfrobu` using the `time` command
- Estimate the number of comparisons as a function of the number of input lines provided to `sfrobu`
- Write a shell script "`sfrobs`" that uses `tr` and the `sort` utility to perform the same function as `sfrob`
  - Encrypted input
    -> tr (decrypt)
    -> sort (sort decrypted text)
    -> tr (encrypt)
    -> encrypted output

Hints

- The qsort method is **not** implemented as quicksort in GCC. Experiment: how many comparisons are required to sort an already sorted input?
- For the shell scripting exercise, consider awk: https://www.gnu.org/software/gawk/manual/gawk.html#Getting-Started
- Make sure you read Rahul's guidelines for Assignment 5 on Piazza (@214).