

Lab 1.1 Report

This laboratory consists of creating our own Cat function and compare the professors (*Cat.c*) time execution with our program time (*MyCat.c*).

The basic functionality of the MyCat.c program was to rewrite the cat.c program by using read, write, open and close instead of their library equivalents as seen on the original *Cat.c* program.

The tools that were used in order to make the comparison was the Linux command *time cat <file name>*.

Time Comparison:

```
loc_close_open_printf_read_writedyld_stub_binder
real    0m0.006s
user    0m0.001s
sys     0m0.003s
Floreths-MacBook-Pro:~ florethgonzalez$
```

Image 1. *MyCat.c* Time execution.

```
__mh_execute_header_rtcopy_main__stderr
pen_fprintf_getc_putcdyld_stub_binder
real    0m0.004s
user    0m0.001s
sys     0m0.002s
Floreths-MacBook-Pro:~ florethgonzalez$
```

Image 2. Original *Cat.c* Time execution.

As seen on the Images above, there are some differences between the programs. We have three variables that measure time execution: *Real*, *User* and *Sys*. These variables are defined as:

- A. **Real:** The actual time spent in running the process from start to finish, as if it was measured by a human with a stopwatch.
- B. **User:** The cumulative time spent by all the CPUs during the computation.
- C. **Sys:** The cumulative time spent by all the CPUs during system-related tasks such as memory allocation.

Graphic Analysis:

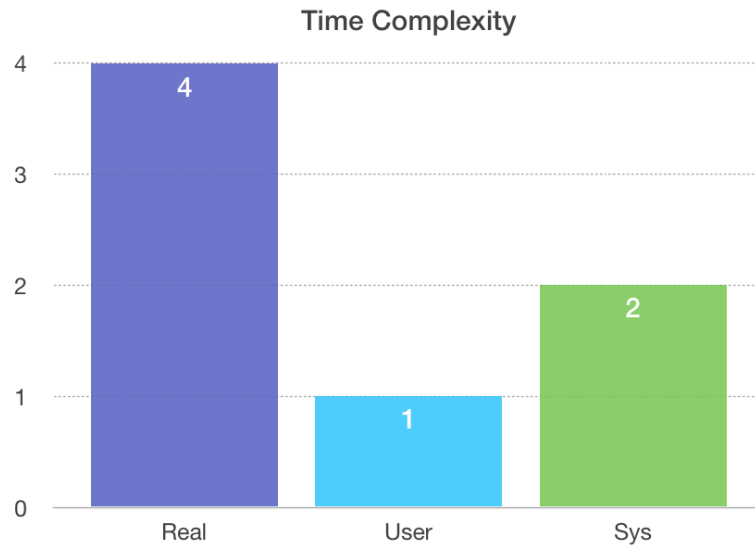


Chart 1. Original *Cat.c*

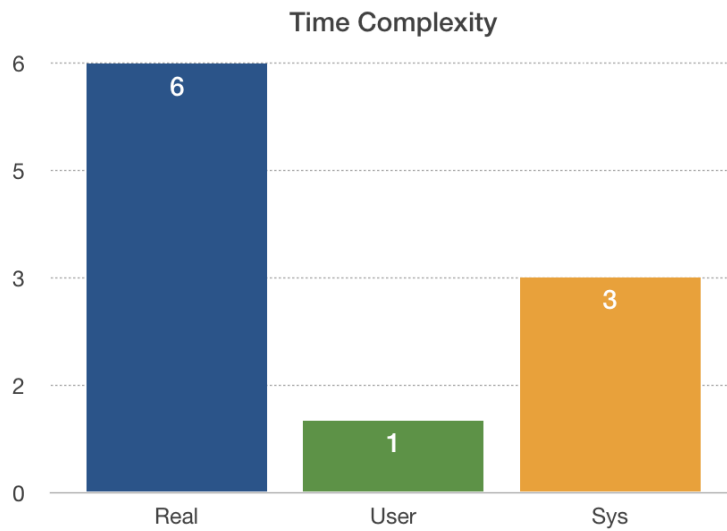


Chart 2. *MyCat.c*

In conclusion, the charts demonstrate that the original *Cat.c* program is faster than the new *MyCat.c* program. The functions implemented on the original program of *Cat* are smoother to run than the ones used on the new program which are slower.