

CS351 Lab #EC4

memcpy & memset

Instructions:

- Assigned date: Thursday November 19th, 2020
- Due date: 11:59PM on Sunday December 13th, 2020 [firm deadline, no extensions; no submissions after this deadline will be accepted]
- Extra Credit: 30 points
- This lab must be done individually
- Please post your questions to the Piazza forum
- Only a softcopy submission is required; it will automatically be collected through GIT at the deadline; email confirmation will be sent to your HAWK email address; submissions will be graded based on the collected submissions at the deadline, unless an email to the TAs at cs351-ta-group@iit.edu with the subject “[CS351] homework submission is delayed” is received; when a student is ready to have their late assignment graded, they must send another email to the TAs at cs351-ta-group@iit.edu with the subject “[CS351] late homework submission is ready”; late submission will be penalized at 5% per day

1 Your Assignment

Memcpy and memset are system functions in the C language. The C library function `void *memcpy(void *dest, const void *src, size_t n)` copies `n` characters from memory area `src` to memory area `dest`. The C library function `void *memset(void *str, int c, size_t n)` copies the character `c` (an unsigned char) to the first `n` characters of the string pointed to, by the argument `str`. Memset is defined in `#include <string.h>`.

Your assignment is to benchmark `memcpy()` and `memset()` on small (4KB), medium (4MB), and large (4GB) buffer sizes. You are also to implement your own `my_memcpy()` and `my_memset()` to potentially improve the performance of these systems calls. You may want to look into SSE and AVX instructions to help with improving performance. You may want to try different compilers, such as GCC, LLVM or ICC; you may also explore different optimization flags to see what might work better. You may also want to parallelize your `memcpy` and `memset` implementations using PThreads. You might also use Valgrind to profile your code to better understand where time is being spent. You will measure the throughput in bytes/second of each operation on different buffer sizes. You may want to perform `memcpy` and `memset` on multiple buffers so that your benchmarks take a few seconds to complete; if only a single buffer will be evaluated, you will need to use cycle accurate timing mechanisms as some of these operations will be extremely short.

Write a report with your findings, and present your data comparing your implementation to the existing `memcpy` and `memset`. You will get 10 points for the benchmark code for `memset` and `memcpy`, 10 points for your own implementation of `my_memset` and `my_memcpy` that can outperform the included Linux `memset` and `memcpy`, and 10 points for the written report. Note that the majority of the 10 points for `my_memset` and `my_memcpy` will only be awarded if your version of `my_memset` and `my_memcpy` will outperform the native `memset` and native `memcpy`, so do your best to be creative!

2 What you will submit

When you have finished implementing the complete assignment as described above, you should submit your solution to your private git repository. Each program must work correctly and be detailed in-line documented. You should hand in:

1. **Source code and compilation (70%):** All of the source code in C and Bash; in order to get full credit for the source code, your code must have in-line documents, must compile (with a Makefile), and must be able to run a variety of benchmarks through command line arguments. Must have working code that compiles and runs on fourier.
2. **Report / Performance (30%):** A separate (typed) design document (named labEC4-report.pdf) describing the results in a table format. You must evaluate the performance of the various parameters outlined to showcase the results. You must summarize your findings and explain why you achieve the performance you achieve, and how the results compare between the various approaches.

To submit your work, simply commit all your changes to the [Makefile](#), [my_memset.c](#), [my_memcpy](#), [runbench.sh](#), and push your work to Github. You can find a git cheat sheet here: <https://www.git-tower.com/blog/git-cheat-sheet/>. Your solution will be collected automatically at the deadline. If you want to submit your homework later, you will have to push your final version to your GIT repository and you will have let the TA know of it through email. There is no need to submit anything on BB for this assignment. If you cannot access your repository contact the TAs.