**Homework 1**

1. Briefly describe the semantics of the following Unix system calls:

a) fork( );

fork() is a system call operation for creating a new process inside a process called a child process which would run concurrently with the parent process that made the system call.

b) exit( );

exit() is a system call operation that terminates the calling process and the children processes are inherited by init(1).

c) getpid( );

getpid() is a system call that returns the process id of the calling process.

d) getppid( );

getppid() is a system call that returns the process id of the parent of the calling process.

e) wait( );

wait() is a system call operation that blocks the calling process until there is a state change in the child process. A state change can be a termination or the child process being stopped by a signal.

f) execl( );

execl() is part of the exec() family of system calls which replace the current process image with a new process image.

2. Consider the evolution of computers in general and the increase in processor speed in particular. Suppose we were to overclock a processor to 9GHz. This processor can execute one instruction per cycle. Further let us suppose that the system is accessing a magnetic disk (HD) with an access time of 10ms.

a. Determine the number of instructions the CPU must pause when waiting for a read request for data from the HD to complete.

Clock speed = 9GHz or $9Gs^{-1}$

Clock cycle = $1/9Gs = 1.1111 \times 10^{-10} = 11.111ns$

**Homework 1**

HD access time = 10ms = $1 \times 10^{-5}$

Cycles skipped = HD access time/Clock cycle = $1 \times 10^{-5} / 1.111111 \times 10^{-10} = 90000$

The CPU must pause for 90,000 Clock Cycles for a read request for data from HD to complete.

b. Suppose that you are designing the machine architecture and want to guarantee the CPU can obtain data from memory within 2 CPU cycles. Given that the address has to travel from the CPU to the memory unit (MMU) and that the data has to travel from memory to the CPU, what is the maximum distance between CPU and the MMU if the signal on the memory bus propagates at 80% of the speed of light?

Signal propagation speed = .8 c = .8 * 299 792 458m/s = 239 833 966m/s

Clock cycle = 1/9Gs = $1.1111 \times 10^{-10}$s = 11.111ns

Distance = Signal propagation speed * Clock cycle = $1.1111 \times 10^{-10}$s * 239 833 966m/s = 2.664821844cm

3. For this problem, you will conduct experiments and analyze the results.

a. Write a small C program that copies data from a file A to a file B byte-by-byte. For different file sizes 1MB – 128MB (doubling the file size in each step), record the time your program requires to complete the file copy when using read() and write() system calls. Generate a graph that depicts the program performance.

b. Repeat this experiment, but instead of copying individual bytes, use larger size portions of the file that are copied in each read() and write() system call. For instance, you may choose to copy the file in chunks of 2, 4, 8, ..., 1024 byte units. Generate a performance graph and interpret your experimental results. What do you observe? Why do you think the system shows the observed behavior?

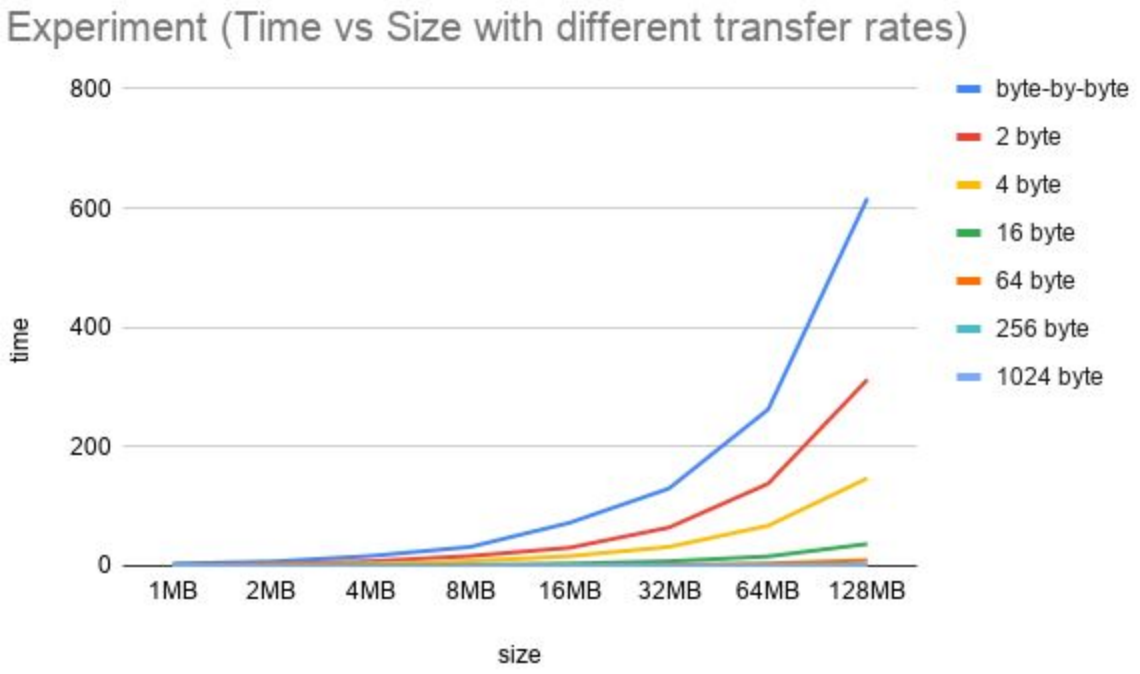
You must submit your programs for parts a) and b); the graphs of the corresponding experiments; and your analysis.

| size | byte-by-byte | 2 byte | 4 byte | 16 byte | 64 byte | 256 byte | 1024 byte |
|---|---|---|---|---|---|---|---|

**Homework 1**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1MB | 3.8128 | 1.8745 | 0.93054 | 0.24091 | 0.06214 | 0.02065 | 0.0064 |
| 2MB | 7.40536 | 3.86037 | 2.4593 | 0.47029 | 0.12432 | 0.03548 | 0.01295 |
| 4MB | 16.87136 | 7.84952 | 4.06191 | 0.95184 | 0.24633 | 0.07017 | 0.02551 |
| 8MB | 32.05114 | 16.78114 | 8.29543 | 1.96125 | 0.52955 | 0.1375 | 0.05009 |
| 16MB | 72.52249 | 30.65706 | 16.50348 | 4.01997 | 0.9826 | 0.2766 | 0.10002 |
| 32MB | 129.92466 | 64.48463 | 32.11637 | 7.93545 | 2.07804 | 0.55797 | 0.20266 |
| 64MB | 262.58222 | 138.10362 | 67.54584 | 16.05841 | 4.26777 | 1.2089 | 0.40499 |
| 128MB | 616.95359 | 312.68834 | 147.1502 | 37.02698 | 9.66092 | 2.51123 | 0.85808 |

The chart above is the raw experiment data for how long each experiment took at different chunks of data.

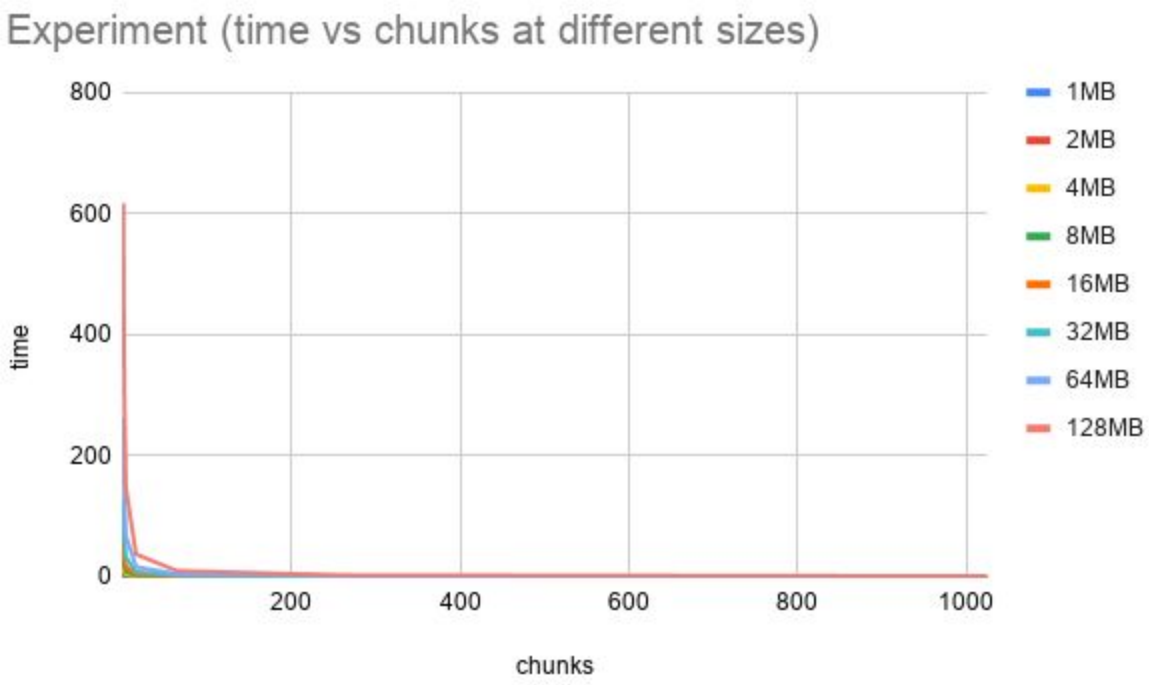



This Graph shows an exponential increase in time as the data grows, I think the observed behavior is exactly as expected as the size of file being copied is doubling and there is an exponential increase in size each step where the exponent is 2, the time it takes to copy also increases exponentially because the system has to make twice as many system calls to copy the data. The trend also shows how that larger and larger chunks of data transfer time is significantly reduced.

**Homework 1**

| chunks | 1MB | 2MB | 4MB | 8MB | 16MB | 32MB | 64MB | 128MB |
|---|---|---|---|---|---|---|---|---|
| 1 | 3.8128 | 7.40536 | 16.87136 | 32.05114 | 72.52249 | 129.92466 | 262.58222 | 616.95359 |
| 2 | 1.8745 | 3.86037 | 7.84952 | 16.78114 | 30.65706 | 64.48463 | 138.10362 | 312.68834 |
| 4 | 0.93054 | 2.4593 | 4.06191 | 8.29543 | 16.50348 | 32.11637 | 67.54584 | 147.1502 |
| 16 | 0.24091 | 0.47029 | 0.95184 | 1.96125 | 4.01997 | 7.93545 | 16.05841 | 37.02698 |
| 64 | 0.06214 | 0.12432 | 0.24633 | 0.52955 | 0.9826 | 2.07804 | 4.26777 | 9.66092 |
| 256 | 0.02065 | 0.03548 | 0.07017 | 0.1375 | 0.2766 | 0.55797 | 1.2089 | 2.51123 |
| 1024 | 0.0064 | 0.01295 | 0.02551 | 0.05009 | 0.10002 | 0.20266 | 0.40499 | 0.85808 |

This chart is a different representation of the same data for analyzing how time in relation to chunks increases decreases with all the sizes in the experiment.




The chart shows how at larger and larger chunks of data transfer, the transfer of each data size decays exponentially, this is to be expected the graph shows the inverse relationship from the previous graph. This graph shows how when the transfer chunk size is increased the time it takes to transfer the file of a particular size is also reduced.

**Homework 1**