In this meeting, we will measure the performance of different tasks on our computers.

- We will work as a team to write file system *microbenchmarks*.
- We will use the seekwatcher tool to see visualize these benchmarks on our actual system.
- We will control for important parameters when evaluating a system.
- We will use filebench to implement these same benchmarks.

Microbenchmarks

A microbenchmark is a test that measures a specific operation. We will write a few microbenchmarks as a group to measure the performance of three operations on our file systems:

- Sequential writes
- Sequential Reads
- Random Writes

Logically, these tasks are very simple. However, there are a surprising number of variables to consider:

- Working set size
- Cache (hot or cold)
- Transfer size
- sync frequency
- Variance

Some of these things we should control at the system level, but some should be incorporated as parameters in our microbenchmarks.

Seekwatcher

Now that we have written these benchmarks, we will run them on our systems. We will also use the seekwatcher tool to visualize their performance.

In particular, we will generate graphs to show the:

- I/O patterns (disk offset vs. time)
- Throughput

(And because it is really cool, you may want to create an animation as well.)

Filebench

Filebench is a powerful benchmarking utility. It provides a framework for defining benchmarks using a Workload Model Language (WML). This makes our tests repeatable and comparable, which is good science. We want to implement our microbenchmarks in filebench using the WML. Please read this article and the filebench wiki for more information on filebench, the WML, and the best practices for file system benchmarking. (When you install filebench, it comes with some *pre-defined personalities*, which you can use as a starting point.)

atop

One of the problems with many benchmarks is that they don't actually measure the thing we want. The atop program is a very powerful resource monitor. By reading the manual page, we can see how to display the disk utilization in real time.

The *bottleneck resource* is the resource in our system that actually determines its performance. Use atop to determine the bottleneck resource in the microbenchmarks we wrote, a predefined filebench personality, and a kernel compile. Is it disk, memory, or CPU?