Gavin Hayes
CS4513 Project 1 Experimentation
1/24/16

Design

System Conditions:
OS: Linux Mint 17 Qiana with kernel: 3.13.0-24-generic x86_64 o
RAM : 8GB
primary partition: ext4 SSD
secondary partition: fuseblk(ntfs) 7200rpm hard drive

Testing apparatus:
The tests were done using bash scripting with the system commands:
`time - run programs and summarize system resource usage`
`sync - flush file system buffers, a wrapper for the c function.`

My `rm` function was used for usage of `rename()` and copying and `unlink()`. Bash was used in order to keep the testing harness separate from the program.

All data was collected from the output in my terminal and after execution was copied to LibreOffice Calc. Since gettimeofday() was recommended all times were recorded in wallclock.

DUMPSTER directory was on primary partition.

Test 1: test1.sh
`time` was run 5 times on test1.sh which contains 10 calls of `rm` with a `sync` after each call. Each `time` was divided by 10 as 10 operations were performed.

Test 2: test2.sh
File was generated using: `fallocate -l 1G test.img`
`time` was run 5 times on test2.sh which contains 1 call of `rm` with a `sync` on a file of 1 GB on the secondary partition.

Test 3 test3.sh
`time` was run 5 times on test3.sh which contains 1 call of `rm -r` with a `sync` on a downloads directory on the secondary partition with a size of 1,764, 668, 218 bytes over 12,575 files and 1,699 folders.

Results

Test 1:

test type	Wall Clock in ms
Test 1	11.2
Test 2	11.5
Test 3	9.6
Test 4	11.1
Test 5	10
Mean	10.68
Standard Deviation	0.8288546314

Test 2:

test type	Throughput in bytes/sec
Test 1	440238550.225502
Test 2	464220416.774751
Test 3	242489120.144535
Test 4	414732261.104674
Test 5	442597619.126134
Mean	400855593.475119
Standard Deviation	90251061.7656573

Test 3:
Estimation:
From taking the bytes/sec speed and multiplying it by the size and adding it to number of files including folders * average link time, the operation should take:
1764668218 bytes /400855593.475119 bytes/second = 4.4 seconds
12575 files+1699 folders = 14274 files
14274 files * .01068 seconds per file = 152.44632 seconds.
Total time = 156.44632 seconds or 2 minutes 32 seconds.

Actual:

test type	Wall Clock in ms
Test 1	65493
Test 2	27475
Test 3	24554
Test 4	23012
Test 5	28218
Mean	33750.4
Standard Deviation	17870.8404754785

Analysis

The results mean the linux (ext4) file system is very smart at saving itself work. The bash testing apparatus appears to be fine as my call to sync is literally a wrapper for the same call that would be done in a c program. The only data that seems valid is the renaming test(test 1) and possibly test 3. There is no clear reason why the first copy in test 3 takes much longer when the program and disk load on the computer stayed constant. Test 2 is mathematically impossible as I have yet to see the secondary partition (7200rpm drive) in normal use beat 70MB per second which at that rate would take 14.6 seconds to copy which is a lot slower than the 4.4 second longest speed recorded result. These discrepancies are likely due to some sort of caching in RAM or the SSD or file analysis being able to recreate a file without copying every byte, or because of how ntfs partitions are mounted in linux. Linking files can be as expensive as moving the actual data. Further testing with ext4 only partitions, more varied files, and operating system file transfer settings may produce better results.