Zack Meeks
****
--profile analysis
Lab 1c
****

Below are the average results from four trials of each of three benchmark tests in simpsh, bash, & dash

Simpsh		Bash		Dash	
0m0.001s	0m0.002s	0m0.001s	0m0.002s	0m0.002s	0m0.001s
1m10.692s	0m0.902s	1m13.161s	0m0.908s	1m13.437s	0m0.882s
0m0.002s	0m0.002s	0m0.002s	0m0.001s	0m0.000s	0m0.002s
0m58.619s	0m0.774s	1m1.794s	0m0.808s	1m0.071s	0m0.744s
0m0.001s	0m0.242s	0m0.002s	0m0.002s	0m0.001s	0m0.002s
1m8.822s	0m0.695s	1m8.883s	0m0.936s	1m12.416s	0m1.035s

Please see the pg. 2--5 for full trial datas.

Most of the results between Bash and Dash don't look significant enough to say anything. However, the third test warranted statistical analysis. I ran a 95% confidence interval on the difference of the means using pooled sample variance and found the confidence interval (note: calculations done in R):

```
> (72.416-68.88275) - 3.182*sqrt(0.08035758+1.265205)/2
[1] 1.687716
> (72.416-68.88275) + 3.182*sqrt(0.08035758+1.265205)/2
[1] 5.378784
```

Thus we see that zero is not inside the interval and hence the difference is statistically significant with alpha=.05 and test statistic under T-distribution with three degrees of freedom.

Given that most of dash's benefit over bash is purportedly related to start up, I'm assuming that Bash happens to have a better sort implementation, since as seen in the –profile results in simpsh, that appears to be the most time heavy command and most thus the first place to look to optimize, but since the times call doesn't itemize bash/dash that is purely an assumption.

Simillarly to Bash vs Dash, I can tell that simpsh is statistically significant with respect to difference of the means between simpsh and Bash (and thus also Dash) for the first two tests. The other differences don't look practically significant to warrant further investigation.

This surprised me at first as I would have thought that surely Simpsh would be the slowest since I'm still very new to operating systems. Upon inspection of the profile results of the benchmarks I understand why simpsh is the fastest. The kernel optimizes performance by seeing that, for example in simpsh1 (benchtest 1 in simpsh) that the code is written as sort, tr, then sort, but this is unnecessary busy work and will result in the same result as tr, sort. So the improved performance in Simpsh is due to the kernel more than it is due to my excellent OS skills, since the kernel decided to abstain from running sort, tr, sort in favor of tr, sort. In the one benchmark test in simpsh that the kernel didn't get to rearrange things, we see that it is almost identical to Bash, and is definitely not statistically signifiant to say it is faster or slower. Based on the data we must conclude that simpsh is the best, followed by bash, followed by dash.

```
******
Dash *
******

bashdash1
0.00175             .001
73.4365             0.8815

bashdash2
0.00025             0.0025
60.071      0.7435

bashdash3
0.001       0.002
72.416      1.0345

$ ./bashdash1.sh
0m0.001s 0m0.001s
1m11.597s 0m0.895s
$ ./bashdash1.sh
0m0.002s 0m0.001s
1m14.713s 0m0.858s
$ ./bashdash1.sh
0m0.003s 0m0.000s
1m11.941s 0m0.872s
$ ./bashdash1.sh
0m0.001s 0m0.002s
1m15.495s 0m0.901s

$ ./bashdash2.sh
0m0.000s 0m0.003s
1m1.139s 0m0.745s
$ ./bashdash2.sh
0m0.001s 0m0.001s
0m59.314s 0m0.729s
$ ./bashdash2.sh
0m0.000s 0m0.003s
1m0.404s 0m0.776s
$ ./bashdash2.sh
0m0.000s 0m0.003s
0m59.427s 0m0.724s

$ ./bashdash3.sh
0m0.001s 0m0.002s
1m12.045s 0m1.018s
$ ./bashdash3.sh
0m0.001s 0m0.002s
1m11.489s 0m1.025s
$ ./bashdash3.sh
```

```
0m0.000s 0m0.003s
1m14.054s 0m1.112s
$ ./bashdash3.sh
0m0.002s 0m0.001s
1m12.076s 0m0.983s
```

```
******
Bash *
******

bashdash1
0.001           0.00175
73.1605                 0.9075

bashdash2
0.0015          0.00125
61.79425        0.80775

bashdash3
0.0015          0.0015
68.88275        0.9355
```

```
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash1.sh
0m0.000s 0m0.003s
1m13.853s 0m0.928s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash1.sh
0m0.001s 0m0.002s
1m14.077s 0m0.938s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash1.sh
0m0.001s 0m0.001s
1m13.860s 0m0.923s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash1.sh
0m0.002s 0m0.001s
1m10.852s 0m0.841s

[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash2.sh
0m0.001s 0m0.001s
0m58.767s 0m0.771s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash2.sh
0m0.003s 0m0.000s
0m59.144s 0m0.829s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash2.sh
0m0.001s 0m0.002s
1m1.113s 0m0.807s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash2.sh
0m0.001s 0m0.002s
1m8.153s 0m0.824s
```

```
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash3.sh
0m0.000s 0m0.003s
1m8.745s 0m0.958s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash3.sh
0m0.003s 0m0.000s
1m8.646s 0m0.955s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash3.sh
0m0.001s 0m0.002s
1m9.289s 0m0.899s
[classzac@lnxsrv09 ~/cs111/Lab1/lab1c]$ ./bashdash3.sh
0m0.002s 0m0.001s
1m8.851s 0m0.930s
```

```
********
Simpsh *
********

simpsh1
0.001054            0.00223925
70.69199            0.9024188

simpsh2
0.00159925          0.0017485
58.61897            0.7735428

simpsh3
0.00072725          0.241518
68.82185            0.6953995

~~s1~~
0.001011            0.002755
70.789485           0.857384

0.000000            0.003109
71.110452           0.940780

0.003205            0.000000
70.467001           0.911586

0.000000            0.003093
70.401034           0.899925


~~s2~~
0.000993            0.002439
58.643945           0.811638
```

```
0.003206        0.000000
58.880892       0.755740

0.000000        0.003444
58.686947       0.759673

0.002198        0.001111
58.264088       0.767120

~~s3~~
0.001213        0.002271
68.693721       0.922123

0.001696        0.957064
68.865509       0.001610

0.000000        0.003270
68.644954       0.934925

0.000000        0.003467
69.083208       0.922940
```

NOTE: to see total process data for simpsh, please see simpsh—profile_runs.pdf