Mehar Nallamalli
SID: 804769644
2/4/18
Lab 1c Report

### Benchmarks to compare bash, dash, and simpsh

> This test will help to understand which shell is the best, and will also help to understand if my simpsh is inefficient.

> The bash command is very simple, just need to encapsulate around a time() to record the user and sys time.

> With this bash command, the first step is to convert to simpsh command.

----

note: used example from the spec sheet

<u>Benchmark 1:</u>

```
(sort < a | cat b - | tr a-z A-Z > c) 2>> d
```

simpsh:

```
./simpsh --profile --rdonly -- a --pipe --pipe --creat --trunc --wronly -- c --creat --append --wronly d --command 0 2 6 sort --command 3 5 6 tr a-z A-Z --command 1 4 6 cat b - --wait
```

bash/dash:

```
time((sort < a | cat b - | tr a-z A-Z > c) 2>> d)
```

<u>Benchmark 2:</u>

```
(cat pg98.txt | sort | tr a-z A-Z > sorted.txt 2> err.txt)
```

simpsh:

```
./simpsh --profile --rdonly pg98.txt --pipe --pipe --wronly sorted.txt --wronly err.txt --command 3 5 6 tr a-z A-Z --command 0 2 6 cat --command 1 4 6 sort
```

bash/dash:

```
time((cat pg98.txt | sort | tr a-z A-Z > sorted.txt 2> err.txt))
```

Similar to benchmark 2, but now simpsh has the wait command

Benchmark 3 (wait):

(cat pg98.txt | sort | tr a-z A-Z > sorted.txt 2> err.txt)

simpsh:

```
./simpsh --profile --rdonly pg98.txt --pipe --pipe --wronly sorted.txt --
wronly err.txt --command 3 5 6 tr a-z A-Z --command 0 2 6 cat --command 1 4
6 sort --wait
```

bash/dash:

```
time((cat pg98.txt | sort | tr a-z A-Z > sorted.txt 2> err.txt))
```

### Results for the shell

	simpsh	bash	dash
Benchmark 1:	usr: 4.888 sys: 0.00	usr: 4.011 sys: 0.005	usr: 4.000 sys: 0.001
Benchmark 2:	usr: 7.560 sys: 0.990	usr: 0.002 sys: 0.444	usr: 0.000 sys: 0.001
Benchmark 3:	usr: 0.000 sys: 0.001	usr: 0.546 sys: 0.004	usr: 0.000 sys: 0.001

### Conclusions

> From my results, it is evident that dash is faster than simpsh and bash. It seemed like the shell time was always 0.000 which makes that the fastest.

> Sometimes, the Sys dash wasn't precise enough so it looked like the the value was always 0, but this is because it is so fast it did not record the millisec time. The dash was faster I believe because it has less overhead, but shell the fastest since no overhead.