Brendon Ng
304-925-492

CS 111: Lab 1C Report

Benchmark 1:

```
(tr A-Z a-z < pg98_100.txt | sort -f | tr x z | cat >> test.txt)
2>test1err.txt
```

```
simpsh:
./simpsh --profile \
--rdonly pg98_100.txt \
--creat --append --wronly test.txt \
--creat --trunc --wronly test1err.txt \
--pipe --pipe --pipe \
--command 0 4 2 tr A-Z a-z \
--command 3 6 2 sort -f \
--command 5 8 2 tr x z \
--command 7 1 2 cat \
--close 4 --close 6 --close 8 \
--wait
```

Method for testing bash and dash:
Made a shell script with the following lines

```
(tr A-Z a-z < pg98_100.txt | sort -f | tr x z | cat >> test.txt)
2>test1err.txt
times
```

Ran this script using bash and dash
bash test1.sh
dash test1.sh
I then totaled the times outputted to find the run time of the script.

	bash	dash	simpsh
Trial 1	11.588	11.40	13.370620
Trial 2	11.623	11.41	13.245087
Trial 3	11.941	11.58	13.140255
Average	11.7173333333333	11.4633333333333	13.2519873333333

Benchmark 2:

```
(sed 's/the/thy/' < pg98_100.txt | tr 123 abc | grep "thou" | wc -l >
out.txt) 2>test2err.txt

simpsh:
./simpsh --profile \
--rdonly pg98_100.txt \
--creat --trunc --wronly out.txt \
--creat --trunc --wronly test2err.txt \
--pipe --pipe --pipe \
--command 0 4 2 sed 's/the/thy/' \
--command 3 6 2 tr 123 abc \
--command 5 8 2 grep "thou" \
--command 7 1 2 wc -l \
--close 4 --close 6 --close 8 \
--wait
```

	bash	dash	simpsh
Trial 1	1.646	1.57	1.594639
Trial 2	1.588	1.59	1.679438
Trial 3	1.585	1.58	1.694859
Average	1.60633333333333	1.58	1.656312

Benchmark 3:

```
(cat pg98_100.txt | tr a-z A-Z | cat | sort -f | egrep -o "*AND" >
out.txt | sleep 5) 2>test3.err.txt

simpsh:
./simpsh --profile \
--rdonly pg98_100.txt \
--creat --trunc --wronly out.txt \
--creat --trunc --wronly test3err.txt \
--pipe --pipe --pipe \
--command 0 4 2 tr a-z A-Z \
--command 3 6 2 cat \
--command 5 8 2 sort -f \
--command 7 1 2 egrep -o "*AND" \
--command 2 2 2 sleep 5 \
--close 4 --close 6 --close 8 \
--wait
```

	bash	dash	simpsh
Trial 1	13.284	12.342	24.625635
Trial 2	11.938	11.742	24.234512
Trial 3	11.968	11.842	23.938424
Average	12.3966666666667	11.9753333333333	24.2661903333333

Conclusion:

From the trials with the three different implementations, we can see that the fastest of the three is dash (hence the name dash). Coming in close second however is bash and the slowest implementation is simpsh.