UID: 905116878

Summary:

		User: Own	Sys: Own	User: Child	Sys: Child
Test 1:	Simpsh	1365 usec	356 usec	23089 usec	2526 usec
	Bash	2200 usec	2400 usec	23400 usec	5000 usec
	Dash	0 usec	0 usec	20000 usec	0 usec
Test 2:	Simpsh	1365 usec	318 usec	27397 usec	1085 usec
	Bash	600 usec	3400 usec	28000 usec	2800 usec
	Dash	0 usec	0 usec	20000 usec	0 usec
Test 3:	Simpsh	1354 usec	329 usec	16608 usec	2332 usec
	Bash	2800 usec	11200 usec	25000 usec	26600 usec
	Dash	0 usec	0 usec	20000 usec	0 usec

Simpsh:

(1) `./simpsh --rdonly input.txt --creat --rdwr test1out.txt --creat --wronly test1err.txt --pipe --pipe --creat --wronly translate.txt --command 0 4 2 sed 's/ //g' --command 3 6 2 sort --command 5 7 2 tr '[a-z]' '[A-Z]' --close 4 --close 5 --close 6 --wait`

(2) `./simpsh --rdonly input.txt --creat --append --rdwr test2out.txt --creat --excl --wronly test2err.txt --pipe --pipe --command 0 1 2 sleep 1 --command 0 4 2 cat --command 3 6 2 sort -f --command 5 1 2 wc --close 4 --close 6 --wait`

(3) `./simpsh --pipe --rdonly input.txt --creat --wronly out.txt --creat --wronly err.txt --pipe --pipe --command 2 1 4 tr ' ' '\n' --command 0 6 4 sort -d --command 5 8 4 tail -n 100 --command 7 3 4 grep -c y --close 0 --close 1 --close 6 --close 8 --wait`

Bash/Dash:

(1)
```
sed 's/ //g' <input.txt 2> test1err.txt | sort 2>test1err.txt
| tr '[A-Z]''[a-z]' >translate.txt 2> test1err.txt
```
(2)
```
sleep 1 < input.txt > test1out.txt 2> test1err.txt
cat <input.txt 2> test1err.txt | sort -f 2>err.txt | wc 2>
test1err.txt >test1out.txt
```
(3)
```
tr " " "\n" < input.txt 2>err.txt | sort -d 2> err.txt |tail
-n 100 2> err.txt |grep -c y 2> err.txt
```

Conclusion:

Based off of the experimental data, dash was by far the "fastest", with my implementation of simpsh coming in second and bash last. It was very surprising that every single test I did in dash had the same result with extremely fast user/system time of the shell, but user time for the child processes was around the same as it was for bash and simpsh. I had to play around with my input file and commands for there even to be a time for dash to show up because previously all the values were 0. I noticed the larger my input file became, the less of a difference there was between bash and dash time. Additionally bash had substantially longer system times than dash and simpsh.

Trials run to calculate averages (optional):

SIMPSH:

Test 1
```
./simpsh --rdonly input.txt --creat --rdwr test1out.txt
--creat --wronly test1err.txt --pipe --pipe --creat --wronly
translate.txt --command 0 4 2 sed 's/ //g' --command 3 6 2 sort
--command 5 7 2 tr '[a-z]' '[A-Z]' --close 4 --close 5 --close 6
--wait
```

	User: Own	Sys: Own	User: Child	Sys: Child
1	1939 usec	0 usec	23323 usec	2284 usec
2	1738 usec	0 usec	22742 usec	3338 usec
3	1601 usec	0 usec	22678 usec	2265 usec
4	0 usec	1781 usec	23936 usec	2481 usec
5	1549 usec	0 usec	22765 usec	2262 usec
Average:	1365 usec	356 usec	23089 usec	2526 usec

Test 2: ./simpsh --rdonly input.txt --creat --append --rdwr test2out.txt --creat --excl --wronly test2err.txt --pipe --pipe --command 0 1 2 sleep 1 --command 0 4 2 cat --command 3 6 2 sort -f --command 5 1 2 wc --close 4 --close 6 --wait

	User: Own	Sys: Own	User: Child	Sys: Child
1	1744 usec	0 usec	28418 usec	0 usec
2	1598 usec	0 usec	27523 usec	1062 usec
3	0 usec	1589 usec	27196 usec	1049 usec
4	1650 usec	0 usec	27381 usec	1052 usec
5	1832 usec	0 usec	26466 usec	2261 usec
Average:	1365 usec	318 usec	27397 usec	1085 usec

Test case 3:
./simpsh --pipe --rdonly input.txt --creat --wronly out.txt --creat --wronly err.txt --pipe --pipe --command 2 1 4 tr ' ' '\n' --command 0 6 4 sort -d --command 5 8 4 tail -n 100 --command 7 3 4 grep -c y --close 0 --close 1 --close 6 --close 8 --wait

	User: Own	Sys: Own	User: Child	Sys: Child
1	1785 usec	0 usec	26669 usec	3112 usec
2	1543 usec	0 usec	25896 usec	2778 usec
3	1725 usec	0 usec	26826 usec	1977 usec
4	0 usec	1645 usec	25861 usec	2729 usec
5	1715 usec	0 usec	27787 usec	1064 usec
Average:	1354 usec	329 usec	26608 usec	2332 usec

BASH:

```
#test 1
sed 's/ //g' <input.txt 2> test1err.txt | sort 2>test1err.txt | tr
'[A-Z]''[a-z]' >translate.txt 2> test1err.txt
```

	User: Own	Sys: Own	User: Child	Sys: Child
1	.001s	.003s	.025s	.005s
2	.004s	.000s	.021s	.007s
3	.003s	.002s	.024s	.005s
4	.003s	.002s	.022s	.005s
5	.000s	.005s	.025s	.003s
Average:	2200 usec	2400 usec	23400 usec	5000usec

```
#test 2
sleep 1 < input.txt > test1out.txt 2> test1err.txt
cat <input.txt 2> test1err.txt | sort -f 2>err.txt | wc 2>
test1err.txt >test1out.txt
```

	User: Own	Sys: Own	User: Child	Sys: Child
1	.000s	.004s	.030s	.001s
2	.000s	.004s	.030s	.001s
3	.001s	.003s	.026s	.004s
4	.002s	.002s	.027s	.004s
5	.000s	.004s	.027s	.004s
Average:	600 usec	3400 usec	28000 usec	2800 usec

```
#test3
tr " " "\n" < input.txt 2>err.txt | sort -d 2> err.txt |tail -n 100
2> err.txt |grep -c y 2> err.txt
```

	User: Own	Sys: Own	User: Child	Sys: Child
1	.002s	.012s	.027s	.026s
2	.001s	.013s	.025s	.028s

3	.003s	.011s	.026s	.026s
4	.003s	.012s	.025s	.030s
5	.005s	.008s	.022s	.023s
Average:	2800 usec	11200 usec	25000 usec	26600 usec

DASH:

```
#test 1
sed 's/ //g' <input.txt 2> test1err.txt | sort 2>test1err.txt | tr
'[A-Z]''[a-z]' >translate.txt 2> test1err.txt
```

	User: Own	Sys: Own	User: Child	Sys: Child
1	0s	0s	.02s	0s
2	0s	0s	.02s	0s
3	0s	0s	.02s	0s
4	0s	0s	.02s	0s
5	0s	0s	.02s	0s
Average:	0 usec	0 usec	20000 usec	0 usec

```
#test 2
sleep 1 < input.txt > test1out.txt 2> test1err.txt
cat <input.txt 2> test1err.txt | sort -f 2>err.txt | wc 2>
test1err.txt >test1out.txt
```

	User: Own	Sys: Own	User: Child	Sys: Child
1	0s	0s	.02s	0s
2	0s	0s	.02s	0s
3	0s	0s	.02s	0s
4	0s	0s	.02s	0s
5	0s	0s	.02s	0s

Average:	0 usec	0 usec	20000 usec	0 usec

```
#test3
tr " " "\n" < input.txt 2>err.txt | sort -d 2> err.txt |tail -n 100
2> err.txt |grep -c y 2> err.txt
```

	User: Own	Sys: Own	User: Child	Sys: Child
1	.000s	.000s	.02s	.000s
2	.000s	.000s	.02s	.000s
3	.000s	.000s	.02s	.000s
4	.000s	.000s	.02s	.000s
5	.000s	.000s	.02s	.000s
Average:	0 usec	0 usec	20000 usec	0 usec