Avery Wong

UID:904 582 269

WINTER 2017

CS111 EGGERT

Here are my test cases that I used for benchmarking:

```
------------------------test 1-----------------------------------------------------------
++++simpsh++++
touch k
touch x
./simpsh --creat --rdonly y --pipe --pipe --wronly x --wronly k --command 0 2 6 cat \
--command 1 4 6 tr A-Z a-z --command 3 5 6 tr -sd 'n' 'm' --close 2 --close 4 --profile \
--wait; \
rm -rf k x y

++++bash++++
cat z | tr A-Z a-z | tr -sd 'n' 'm' > y

++++dash++++
cat z | tr A-Z a-z | tr -sd 'n' 'm' > y

------------------------test 2-----------------------------------------------------------
++++simpsh++++
touch a; \
./simpsh \
--rdonly a \
--creat --wronly g \
--creat --wronly c \
--pipe \
--pipe \
--command 0 4 2 head -c 1MB /dev/urandom \
--command 3 6 2 tr -s A-Z a-z \
--command 5 1 2 cat \
--close 4 \
--close 6 \
--profile \
--wait; \
rm -rf a g c

++++bash++++
head -c 1MB /dev/urandom | tr -s A-Z a-z | cat > g
++++dash++++
head -c 1MB /dev/urandom | tr -s A-Z a-z | cat > b

-----------------------test 3--------------------------------------------------------
++++simpsh++++
./simpsh --creat --rdwr e --pipe --pipe --pipe --creat --rdwr out --command 0 2 7 cat
/dev/urandom --command 1 4 7 tr -dc 'a-zA-Z0-9' --command 3 6 7 fold -w 500000 \
--command 5 7 7 head -n 1 --profile --wait; \

rm -rf e out

++++bash++++
cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 500000 | head -n 1 > a

++++dash++++
cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 500000 | head -n 1 > a
```

Data Table

I ran my test script five times and I took the average, for the cpu times of simpsh, I added the parent and child processes together.

Test 1

	Simpsh	Bash	Dash
User CPU Time(s)	.0005112	.00001	0.0
System CPU Time(s)	.0028442	.003	0.0
Max Resident Size(kb)	1186	641	640
Page Reclaims (soft page faults)	908	212	207
Page faults (hard page faults)	0	0	0
Block Inputs	0	0	0
Block Outputs	0	0	0
Voluntary Context Switches	17	2	3
Involuntary Context Switches	4	1	1

Test 2

	Simpsh	Bash	Dash
User CPU Time(s)	.0100706	.0104	.01
System CPU Time(s)	.1145074	.1154	.110

Max Resident Size(kb)	708	708	704
Page Reclaims (soft page faults)	671	226	223
Page faults (hard page faults)	0	0	0
Block Inputs	0	0	0
Block Outputs	1952	0	0
Voluntary Context Switches	297	2	2
Involuntary Context Switches	18	1	1

Test 3

	Simpsh	Bash	Dash
User CPU Time(s)	.0257056	.028	.02
System CPU Time(s)	.431622	.433	.43
Max Resident Size(kb)	1836	732	728
Page Reclaims (soft page faults)	1358	227	227
Page faults (hard page faults)	0	0	0
Block Inputs	0	0	0
Block Outputs	0	0	0
Voluntary Context Switches	690	258	258
Involuntary Context Switches	13	1	1

Conclusion

Looking at my data simpsh performed better than bash but not as good as dash. However dash probably performed better because of the rounding error in dash, as it only give me two significant digits at most. Simpsh, however is inefficient, compared to Bash and Dash in terms of spaxe complexity, as you can see from the Max Resident Size can easily double of that of Dash and Bash. Also, it is worth noting that Simpsh performed 2 to 3 times voluntary context switches as Bash and Dash, which is probably due to the many system calls I have used in my source code. All in all, I think Simpsh was the best in terms of time, but inefficient in terms of memory usage.

Note: I have included my testscripts and data.txt in my submission for verification