Lab 1C: Simpleton Shell Report

Devam Shroff
504923307
2/1/2019

Benchmark Tests:

The input file used is pg98_100.txt, which is generated by the TAs Lab 1C grading script using pg98.txt, an text file that was provided to us.
The --profile option was implemented at the beginning of the simpsh commands and all the

	simpsh	bash / dash
Test Case 1	./simpsh --profile --rdonly pg98_100.txt \ --pipe --pipe --wronly outb1.txt --wronly errb1.txt \ --command 3 5 6 head 10 --command 0 2 6 cat --command 1 4 6 sort \ --close 1 --close 2 --close 3 --close 4 --wait \	cat pg98_100.txt | \ sort | \ head 10 > outb1.txt 2>errb1.txt \ times \
Test Case 2	./simpsh --profile --rdonly pg98_100.txt \ --pipe --pipe --wronly outb1.txt --wronly errb1.txt \ --command 3 5 6 tr A-Z a-z --command 0 2 6 tac --command 1 4 6 sort \ --close 1 --close 2 --close 3 --close 4 --wait \	tac pg98_100.txt | \ sort | \ tr A-Z a-z > outb1.txt 2>errb1.txt \ times \
Test Case 3	./simpsh --profile --rdonly pg98_100.txt \ --pipe --pipe --wronly outb1.txt --wronly errb1.txt \ --command 3 5 6 uniq --command 0 2 6 cat --command 1 4 6 sort \ --close 1 --close 2 --close 3 --close 4 --wait \	cat pg98_100.txt | \ sort | \ uniq > outb1.txt 2>errb1.txt \ times \

Performance

benchmark			simpsh	bash	dash
1	user	child	5.2138 s	5.293 s	5.365 s
		shell	0.000615 s	0.002 s	0.001 s
	kernel	child	0.3411 s	0.286 s	0.351 s
		shell	0.000516 s	0.003 s	0.001 s
2	user	child	5.321 s	5.361 s	5.524 s
		shell	0.00005 s	0.000 s	0.001 s
	kernel	child	0.4982 s	0.644 s	0.501 s
		shell	0.000569 s	0.001 s	0.001 s
3	user	child	5.627 s	5.942 s	5.712 s
		shell	0.0003397 s	0.001 s	0.000 s
	kernel	child	0.3397 s	1.224 s	0.526 s
		shell	0.000545 s	0.001 s	0.001 s

Conclusion

- It is evident that our simpsh has the slightly performance by a landslide when compared to bash and dash for all 3 benchmarks.
- Child processes take about the same time in all 3 but the Simpleton Shells spends the least amount of time in the shell.
- However, the difference in performance is in the order of milliseconds and therefore can be accounted to background processes running in the background at the time.
- The performance difference between benchmark tests is also negligible: different individual commands don’t seem to have different run times depending on the shell.