Lab 1C Writeup

Anjali Pemmaraju

Professor Eggert, Winter 2017

To test my benchmarks, I timed my script, the bash shell, and the dash shell with the same commands each 3 times. I then took the average of the system times and user times for each to compare them to each other and determine which was the most efficient.

Benchmark 1

./simpsh --profile --rdonly a0.txt --creat --append --wronly d --pipe --pipe --pipe --pipe --command 0 4 2 sort --command 3 6 2 tr A-Z a-z --command 5 8 2 tr -d [:digit] --command 7 10 2 sed “s/one/uno/I” --command 9 1 2 sed ‘/.’,’^$/!d’ --wait

=

times ; (sort a0.txt | tr A-Z a-z | tr -d [:digit’ | sed “s/one/uno/I” | sed ‘/./,/^$/!d’ > c) 2>>d ; times

Benchmark 2

./simpsh --profile --rdonly a0.txt --creat --trunc --wronly c --creat --append --wronly d --pipe --pipe --pipe --command 0 6 2 cat --command 5 8 2 grep “the” --command 7 4 2 grep -v “a” --command 3 1 2 wc -l --wait

=

times ; (cat a0.txt | grep “the” | grep -v “a” | wc -l > c) 2>>d ; times

Benchmark 3

./simpsh --profile --rdonly a0.txt --creat --trunc --wronly c --creat --append --wronly d --pipe --pipe --pipe --pipe --command 0 10 2 cat --command 9 6 2 uniq --command 5 4 2 sed -n ‘/^The/p’ --command 3 8 2 grep ‘!$’ --command 7 1 2 wc -l --wait

=

times ; (cat a0.txt | uniq | sed -n ‘/^The/p’ | grep ‘!$’ | wc -l > c) 2>>d ; times

	Benchmark 1		Benchmark 2		Benchmark 3	
Average Time→	User Time	System Time	User Time	System Time	User Time	System Time
./simpsh	13.4746	4.4519	0.3538	0.3856	1.0202	0.7293
bash	10.1207	1.504	0.3323	0.1123	0.989	0.2
dash	10.1333	0.4600	0.3767	0.11	0.98	0.1833

Based on these results, the dash shell is more efficient in terms of total time spent. It was always faster than all other commands in terms of time spent in the kernel and only marginally slower in some of the other cases for user time, which I assume is due to an efficient way of calling multiple system calls at once. After this the bash shell was second faster and then the simpsh call. These two were comparable in terms of user time but the bash shell was much faster in terms of system calls. This once again may be by somehow grouping together system calls (or my implementation is just being inefficient). This means that in my opinion, given the user and system times both, that the dash shell is the fastest, then the bash shell, then my script. I am not surprised by these results because dash was designed to be faster and more efficient than bash and because bash has been refined much more than my script has.