Test cases used (bash version):

1. `cat /dev/urandom | tr -dc 'a-zA-Z0-9' | head -c 10M | grep "Anthony"`
2. `cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | head -n 1M | sort`
3. `cat /dev/urandom | tr -dc 'a-zA-Z0-9' | tr A-Z a-z | fold -w 32 | head -n 5M | tail -n 2M | head -n 1M | sed 's/abcdef/hello/g'`

Test cases used (simpsh version):

1. `./simpsh --profile --rdonly dummyRead --creat --wronly err --pipe --pipe --pipe --command 0 3 1 cat /dev/urandom --command 2 5 1 tr -dc 'a-zA-Z0-9' --command 4 7 1 head -c 10M --command 6 1 1 grep 'Anthony'`
2. `./simpsh --profile --rdonly dummyRead --wronly /dev/null --wronly err --pipe --pipe --pipe --pipe --command 0 4 2 cat /dev/urandom --command 3 6 2 tr -dc 'a-zA-Z0-9' --command 5 8 2 fold -w 32 --command 7 10 2 head -n 1M --command 9 1 2 sort`
3. `./simpsh --profile --rdonly dummyRead --wronly /dev/null --wronly err --pipe --pipe --pipe --pipe --pipe --pipe --pipe --command 0 4 2 cat /dev/urandom --command 3 6 2 tr -dc 'a-zA-Z0-9' --command 5 8 2 tr A-Z a-z --command 7 10 2 fold -w 32 --command 9 12 2 head -n 5M --command 11 14 2 tail -n 2M --command 13 16 2 head -n 1M --command 15 1 2 grep "abcdef"`

Test cases used (execline version):

1. `pipeline cat /dev/urandom "" pipeline tr -dc 'a-zA-Z0-9' "" pipeline head -c 10M "" grep 'Anthony'`
2. `pipeline cat /dev/urandom "" pipeline tr -dc 'a-zA-Z0-9' "" pipeline fold -w 32 "" pipeline head -n 1M "" sort`
3. `pipeline cat /dev/urandom "" pipeline tr -dc 'a-zA-Z0-9' "" pipeline tr A-Z a-z "" pipeline fold -w 32 "" pipeline head -n 5M "" pipeline tail -n 2M "" pipeline head -n 1M "" grep "abcdef"`

Average (5 trials) real time (wall clock time)

Test cases	bash	simpsh	execline
1	4.4162s	N/A	4.3756s
2	17.587s	N/A	17.4826s
3	1m 10.959s	N/A	1m 9.8842

Average (5 trials) user time

Test cases	bash	simpsh	execline
1	0.2224s	0.2201s	N/A
2	5.0596s	5.4003s	N/A
3	5.0156s	5.1723s	N/A

Average (5 trials) system time

Test cases	bash	simpsh	execline
1	4.359s	5.4153s	N/A
2	14.1634s	14.7449s	N/A
3	1m 10.6122s	1m 19.9821s	N/A

After 7+ hours of trying, it turns out there's simply no way of getting the child processes time on execline. So in this case, we can only compare bash and execline by comparing their real time (wall clock time), and compare bash and simpsh by their user and system time.

As the data from the first table suggest, bash and execline have pretty much the same performance. Although execline seems to be a little bit faster in terms of performance, real time (wall clock time) is not really a good indicator. We therefore conclude that the comparison between bash and execline is inconclusive.

Comparing the user time between bash and simpsh (second table), we can see that there really isn't much of a difference either. In terms of user time, the comparison between bash and simpsh is inconclusive.

Comparing the system time between bash and simpsh (third table), we can clearly see that simpsh is slower than bash. Because these data are generated from the average of five trials, it is a good indicator that time performance wise, bash is definitely better than simpsh. Drawing data from system time, we can conclude that bash has better performance than simpsh.