simpsh	bash	execline

--rdonly /dev/random
User time: 0.000006s
System time: 0.000006s
Total time: 0.000012s
 	
--pipe
User time: 0.000003s
System time: 0.000003s
Total time: 0.000006s
 	
--pipe
User time: 0.000002s
System time: 0.000002s
Total time: 0.000004s
 	
--wronly output.txt
User time: 0.000046s
System time: 0.000046s
Total time: 0.000092s
 	
--wronly error.txt
User time: 0.000031s
System time: 0.000031s
Total time: 0.000062s
 	
--command 0 2 6 head -10000
User time: 0.000049s
System time: 0.000049s
Total time: 0.000098s
 	
--command 1 4 6 sort
User time: 0.000027s
System time: 0.000027s
Total time: 0.000054s
 	
--command 3 5 6 cat
User time: 0.000028s
System time: 0.000028s
Total time: 0.000056s
 	
--wait
0 head -10000
0 sort
0 cat
PARENT TIME:
User time: 0.000097s
System time: 0.000097s
Total time: 0.000194s
CHILDREN TIME:
User time: 0.129075s
System time: 6.357977s
	time head -10000 /dev/random | sort | cat > output.txt
	
	real 4m9.505s
	user 0m0.129s
	sys 0m6.956s
	--- 	
	Attempt to time each command separately:
	
	time head -10000 /dev/random > a
	real 3m50.063s
	user 0m0.016s
	sys 0m6.407s
	
	time sort a > b
	real 0m0.015s
	user 0m0.007s
	sys 0m0.002s
	
	time cat b > output.txt
	real 0m0.008s
	user 0m0.001s
	sys 0m0.002s
		time execlineb exec.sh
		(where the script contains:
		
		`redirfd -r 0 /dev/random`
		`pipeline { head -10000 }`
		`pipeline { sort }`
		`redirfd -w 1 output.txt`
		`cat`
		)
		real 4m3.679s
		user 0m0.001s
		sys 0m0.009s
		
		--- 	
		Attempt to time each command separately, since pipeline time was not included above:
		
		time execlineb exec2.sh
		(where script contains:
		`redirfd -r 0 /dev/random`
		`redirfd -w 1 a`
		`head -10000`
		)
		real 3m20.538s
		user 0m0.009s
		sys 0m6.146s
		
		time execlineb exec3.sh
		(where script contains:
		`redirfd -r 0 a`
		`redirfd -w 1 b`
		`sort`)
		real 0m0.140s
		user 0m0.098s
		sys 0m0.004s
		
		time execlineb exec4.sh
		(where script contains:
		`redirfd -r 0 b`
		`redirfd -w 1 output.txt`
		`cat`)
		real 0m0.044s
		user 0m0.000s
		sys 0m0.006s

Total time: 6.487052s OVERALL TIME FOR MAIN PROCESS, EXCLUDING CHILD COMMANDS: User time: 0.000422s System time: 0.000422s Total time: 0.000844s		

pipel--rdonly poem.txt
User time: 0.000000s
System time: 0.000094s
Total time: 0.000094s
 	
--pipe
User time: 0.000000s
System time: 0.000006s
Total time: 0.000006s
 	
--trunc --wronly c
User time: 0.000000s
System time: 0.000085s
Total time: 0.000085s
 	
--creat --append --wronly d
User time: 0.000000s
System time: 0.000056s
Total time: 0.000056s
 	
--command 0 2 4 tr A-Z a-z
User time: 0.000000s
System time: 0.000090s
Total time: 0.000090s
 	
--command 1 3 4 cat
User time: 0.000000s
System time: 0.000058s
Total time: 0.000058s
 	
--wait
0 tr A-Z a-z
0 cat
PARENT TIME:
User time: 0.000000s
System time: 0.000121s
Total time: 0.000121s
CHILDREN TIME:
User time: 0.000000s
System time: 0.002337s
Total time: 0.002337s
 	
OVERALL TIME FOR MAIN PROCESS, EXCLUDING CHILD COMMANDS:
User time: 0.000000s
	time tr A-Z a-z < poem.txt | cat > c 2> d
	real 0m0.005s
	user 0m0.001s
	sys 0m0.002s
	
	---------- 	
	Attempt to time each command separately:
	
	time tr A-Z a-z < poem.txt > a
	real 0m0.006s
	user 0m0.000s
	sys 0m0.002s
	
	time cat a > c
	real 0m0.003s
	user 0m0.000s
	sys 0m0.001s
		time execlineb exec5.sh
		(where the script contains:
		`redirfd -r 0 poem.txt`
		`pipeline { tr A-Z a-z }`
		`redirfd -w 1 c`
		`redirfd -w 2 d`
		`cat`)
		
		real 0m0.009s
		user 0m0.000s
		sys 0m0.003s
		
		---------- 	
		Attempt to time each command separately:
		
		time execlineb exec6.sh
		(where script contains:
		`redirfd -r 0 poem.txt`
		`redirfd -w 1 a`
		`tr A-Z a-z`)
		real 0m0.008s
		user 0m0.001s
		sys 0m0.002s
		
		time execlineb exec7.sh
		(where script contains:
		`redirfd -r 0 a`
		`redirfd -w 1 c`
		`redirfd –w 2 d`
		`cat`)
		real 0m0.014s
		user 0m0.000s
		sys 0m0.003s

System time: 0.000735s Total time: 0.000735s		

--rdonly output.txt
User time: 0.000000s
System time: 0.000084s
Total time: 0.000084s
 	
--wronly c
User time: 0.000000s
System time: 0.000041s
Total time: 0.000041s
 	
--wronly error.txt
User time: 0.000000s
System time: 0.000035s
Total time: 0.000035s
 	
--command 0 1 2 wc -c
User time: 0.000000s
System time: 0.000101s
Total time: 0.000101s
 	
--wait
0 wc -c
PARENT TIME:
User time: 0.000000s
System time: 0.000029s
Total time: 0.000029s
CHILDREN TIME:
User time: 0.000000s
System time: 0.001147s
Total time: 0.001147s
 	
OVERALL TIME FOR MAIN PROCESS, EXCLUDING CHILD COMMANDS:
User time: 0.000000s
System time: 0.000492s
Total time: 0.000492s
	time wc –c output.txt > output2.txt 2> error.txt
	
	real 0m0.007s
	user 0m0.000s
	sys 0m0.001s
		time execlineb exec8.sh
		(where script contains:
		`redirfd -r 0 output.txt`
		`redirfd -w 1 output2.txt`
		`wc -c`)
		
		real 0m0.019s
		user 0m0.000s
		sys 0m0.003s

The total time to execute for simpsh is the overall time for the main process + total time for child processes.

Comparison of total times for three test cases:

SIMPSH	BASH	EXECLINE
0.000844s + 6.487052s = 6.487896s	7.085s (total), 6.435 (sep)	0.01s (pipeline), 6.263 (separate)
0.000735s + 0.002337s = 0.003072	0.003s (total), (0.003 sep)	0.003s (pipeline), 0.006 (separate)
0.000492s + 0.001147s = 0.001639	0.001s (both)	0.003s (both)

CONCLUSION:

It is difficult to compare, because some things are done in parallel and some are not. So I got the times both for entire commands and for commands separated out. For example, timing execline while using pipelines does not get the time for the pipelines, so I also ran execline with separate commands not using pipelines and added those.

Overall, the times are pretty different for all three test cases. Part of this may be due to slight differences when running commands at different times. However, I did run each test multiple times to make sure the timing results were similar every iteration. There are differences in time, but I think the differences are not enough to show that one program is clearly the fastest.
For the first case, execline is clearly faster.
For the second, they are all about the same, although bash shows the same time for running the commands in one line versus running separately, while execline is slightly slower when running commands separately. It matches bash time though using pipelines.
For the third case, execline is actually slightly slower than both bash and simpsh.

I expected execline to be fast, because it is supposed to reduce overhead. This shows up in the first test case, but not so much in the others. In conclusion, I did not see that one program was significantly faster than the others in all cases.