

It is either impossible or exceedingly difficult to get to time specific child processes of execline. Using time execlineb \$SCRIPT\_NAME only reports accurate real time for the whole script, the sys time and user time is questionably accurate, so comparisons will only be made for real time between execlineb and bash and between execlineb and simpsh.

```
./simpsh --profile --rdonly testfiles/foo --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 --wait
cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | head -n 1M | sort > /dev/null
```

```
#!/bin/execlineb
time -p foreground {
redirfd -r 0 testfiles/foo
redirfd -w 1 /dev/null
redirfd -w 2 err
pipeline
cat /dev/urandom
""
pipeline
tr -dc 'a-zA-Z0-9'
""
pipeline
fold -w 32
""
pipeline
head -n 1M
""
sort
}
```

simpsh		bash		execline	
real	0m18.826s	real	0m19.409s	real	0m18.295s
user	0m5.449s	user	0m5.146s	user	0m4.242s
sys	0m15.124s	sys	0m16.080s	sys	0m0.091s
real	0m18.515s	real	0m17.596s	real	0m17.390s
user	0m5.300s	user	0m5.098s	user	0m4.253s
sys	0m14.879s	sys	0m14.158s	sys	0m0.073s
real	0m17.572s	real	0m17.503s	real	0m17.500s
user	0m5.120s	user	0m5.024s	user	0m4.296s
sys	0m14.126s	sys	0m14.088s	sys	0m0.081s
real	0m17.443s	real	0m18.049s	real	0m17.498s
user	0m5.032s	user	0m5.378s	user	0m4.293s
sys	0m14.082s	sys	0m14.526s	sys	0m0.079s
real	0m17.671s	real	0m17.749s	real	0m19.002s
user	0m5.163s	user	0m5.250s	user	0m4.528s
sys	0m14.187s	sys	0m14.293s	sys	0m0.079s
Average					
Real - 18.005s		Real - 18.061s		Real - 17.937s	
User - 5.213s		User - 5.180s		User - N/A	
Sys - 14.480s		Sys - 14.629		Sys - N/A	

In this case, execline wins if we compare the real time to bash and simpsh real times. Otherwise, simpsh beats bash in terms of sys time but loses for user time.

```
./simpsh --profile --rdonly testfiles/foo --creat --wronly err --pipe --pipe --pipe --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 'James' --wait
```

```
cat /dev/urandom | tr -dc 'a-zA-Z0-9' | head -c 10M | grep 'James' > /dev/null
```

```
#!/bin/execlineb
time -p foreground {
redirfd -r 0 testfiles/foo
redirfd -w 2 err
pipeline
cat /dev/urandom
""

pipeline
tr -dc 'a-zA-Z0-9'
""

pipeline
head -c 10M
""

grep 'James'
}
```

simpsh		bash		execline	
real	0m4.398s	real	0m4.726s	real	0m4.391s
user	0m0.223s	user	0m0.244s	user	0m0.015s
sys	0m4.344s	sys	0m4.664s	sys	0m0.022s
real	0m4.415s	real	0m4.779s	real	0m4.348s
user	0m0.218s	user	0m0.232s	user	0m0.015s
sys	0m4.365s	sys	0m4.752s	sys	0m0.020s
real	0m4.411s	real	0m4.632s	real	0m4.227s
user	0m0.219s	user	0m0.211s	user	0m0.020s
sys	0m4.362s	sys	0m4.571s	sys	0m0.014s
real	0m4.416s	real	0m4.762s	real	0m4.359s
user	0m0.222s	user	0m0.229s	user	0m0.017s
sys	0m4.364s	sys	0m4.714s	sys	0m0.022s
real	0m4.419s	real	0m6.122s	real	0m4.369s
user	0m0.220s	user	0m0.224s	user	0m0.022s
sys	0m4.370s	sys	0m6.074s	sys	0m0.013s
Average					
Real	- 4.412s	Real	- 5.004s	Real	- 4.339s
User	- 0.220s	User	- 0.228s	User	- N/A
Sys	- 4.361s	Sys	- 4.955s	Sys	- N/A

execline is faster than bash and simpsh when compared using real times. Otherwise, simpsh is faster than bash in both user and sys time metrics.

```
./simpsh --profile --rdonly testfiles/foo --creat --wronly err2 --pipe --wronly /dev/null --command 0 3 1
od -ta -f -N 2000000 /dev/urandom --command 2 4 1 sort --wait
```

```
od -ta -f -N 2000000 /dev/urandom | sort > /dev/null
```

```
#!/bin/execlineb
time -p foreground {
redirfd -r 0 testfiles/foo
redirfd -w 1 /dev/null
redirfd -w 2 err2
pipeline
od -ta -f -N 2000000 /dev/urandom
""
sort
}
```

simpsh		bash		execline	
real	0m3.928s	real	0m3.605s	real	0m3.695s
user	0m3.324s	user	0m3.371s	user	0m1.596s
sys	0m0.437s	sys	0m0.426s	sys	0m0.059s
real	0m3.901s	real	0m3.565s	real	0m3.563s
user	0m3.589s	user	0m3.299s	user	0m1.495s
sys	0m0.467s	sys	0m0.460s	sys	0m0.070s
real	0m3.659s	real	0m3.744s	real	0m3.666s
user	0m3.422s	user	0m3.456s	user	0m1.542s
sys	0m0.426s	sys	0m0.452s	sys	0m0.091s
real	0m3.844s	real	0m3.683s	real	0m3.598s
user	0m3.565s	user	0m3.374s	user	0m1.523s
sys	0m0.431s	sys	0m0.487s	sys	0m0.065s
real	0m3.610s	real	0m3.965s	real	0m3.560s
user	0m3.345s	user	0m3.695s	user	0m1.513s
sys	0m0.455s	sys	0m0.447s	sys	0m0.062s
Average					
Real	- 3.788s	Real	- 3.712s	Real	- 3.616s
User	- 3.449s	User	- 3.439s	User	- N/A
Sys	- 0.443s	Sys	- 0.377s	Sys	- N/A

simpsh is slower than bash for both user and sys time metrics. execline is yet again faster than the other two.

## Conclusion

Bash has had more man hours put into its development, many of which have presumably been used to optimize the code. Our simpsh code was focused on correctness; we did not worry much about performance, so potential optimizations were ignored. In addition, simpsh commands take up a lot more space than equivalent bash commands; more time could have been spent parsing these commands as a result.

In summary, execline is faster than simpsh and bash when comparing real times. Bash and simpsh trade blows, and appear to be equally as fast, depending on the workload.