|  |  |  |  |
| --- | --- | --- | --- |
|  | Benchmark 1 | Benchmark 2 | Benchmark 3 |
| Bash | Average user time: 0.469s  Average sys time: 0.052s | Average user time: 0.4208s  Average sys time: 0.0336s | Average user time: 0.4038s  Average sys time: 0.032s |
| Simpsh | Average user time: 0.4345548s  Average sys time: 0.65594s | Average user time: 1.2481216s  Average sys time: 0.810476s | Average user time: 1.2763528s  Average sys time: 0.910886s |
| Execline | Average user time: 0.01s  Average sys time: 0.01s | Average user time: 0.204s  Average sys time: 0.068s | Average user time: 0.346s  Average sys time: 0.048s |
| Dash | Average user time: 0.434s  Average sys time: 0.0522s | Average user time: 0.4026s  Average sys time: 0.0342s | Average user time: 0.4036s  Average sys time: 0.0348s |
| Tcsh | Average user time: 0.00s  Average sys time: 0.00s | Average user time: 0.402s  Average sys time: 0.008s | Average user time: 0.398s  Average sys time: 0.01s |

Fastest program for benchmark 1: execline

Fastest program for benchmark 2: execline

Fastest program for benchmark 3: execline

Design problem considered:

Fastest program for benchmark 1: tcsh

Fastest program for benchmark 2: execline

Fastest program for benchmark 3: execline

Conclusions:

My simpsh program is the slowest out of all the shells/interpreters. This makes sense because I am not as good of a programmer as the people who wrote these shells. Dash seems to be very similar to bash in terms of both system time and user time. Execline is faster than bash in all cases, and tcsh is the fastest of all three.

It is well known that dash is faster than bash due to its smaller size and less dependency on shared libraries. It is also well known that execline can be faster than bash under certain situations because it is simpler.