Using the page 98 provided in the spec

Test case 1:

```
times bash -c '(cat <pg98.txt | grep -n history | wc) 2>Aerr >Aout'
        parent          user: 0m0.188s    system:  0m0.106s
        children        user: 0m0.683s    system: 0m0.479s
times dash -c '(cat <pg98.txt | grep -n history | wc) 2>Aerr >Aout'
        parent          user: 0m0.190s    system: 0m0.107s
        children        user: 0m0.683s    system: 0m0.479s
./simpsh --profile --rdonly pg98.txt --creat --wronly Aout --creat --wronly Aerr --pipe --pipe --command 0 4 2 cat \
--command 3 6 2 grep -n history --close 3 --close 4 --command 5 1 2 wc --close 5 --close 6 --wait
        parent          user: 0s 0us              system: 0s 1820us
        children        user: 0s 2717us           system: 0s 4310us
```

Test case 2:

```
times bash -c '(head -n 20 2>Berr <pg98.txt | sort 2>>Berr | tail) >Bout'
        parent          user: 0m0.230s    system: 0m0.135s
        children  user: 0m9.137s    system: 0m2.471s
times dash -c '(head -n 20 2>Berr <pg98.txt | sort 2>>Berr | tail) >Bout'
        parent          user: 0m0.224s    system: 0m0.132s
        children        user: 0m9.137s    system: 0m2.471s
./simpsh --profile --rdonly pg98.txt --creat --wronly Bout --creat --wronly Berr --pipe --pipe \
--command 0 4 2 head -n 20 --command 3 6 2 sort --close 3 --close 4 --command 5 1 2 tail --close 5 --close 6 --wait
        parent          user: 0s 0us              system: 0s 1548us
        children        user: 0s 0us              system: 0s 3628us
```

Test case 3:

```
times bash '(cat <pg98.txt | awk '{print $1}' | sed -n '/e/p' | sort -u | wc) >Cout 2>Cerr'
        parent          user: 0m0.217s    system: 0m0.084s
        children        user: 0m0.326s    system: 0m0.204s
times dash '(cat <pg98.txt | awk '{print $1}' | sed -n '/e/p' | sort -u | wc) >Cout 2>Cerr'
        parent          user: 0m0.207s    system: 0m0.084s
        children        user: 0m0.326s    system:0m0.204s
./simpsh --profile --rdonly pg98.txt --creat --wronly Cout --creat --wronly Cerr --pipe --pipe --pipe --pipe \
--command 0 4 2 cat --command 3 6 2 awk '{print $1}' --close 3 --close 4 --command 5 8 2 sed -n '/e/p' \
--close 5 --close 6 --command 7 10 2 sort -u --close 7 --close 8 --command 9 1 2 wc --close 9 --close 10 --wait
        parent          user: 0s 1221us   system: 0s 789us
        children        user: 0s 29898us system: 0s 6667us
```

Conclusion:

According to my data, the simpsh command was significantly faster than the bash and dash implementations while the bash and dash implementations were almost exactly the same speeds. On average, simpsh is at least 10 times faster both, and bash and dash differ only by fractions of seconds. I also noticed that simpsh was very varied in use of user and system modes (it’s harder to tell since I averaged the times), and sometimes it didn’t use user mode at all. dash and bash were consistent in their use of user and system modes.Thus the simpsh command would be faster in most situations (though the API is a pain to use, and is less compact)