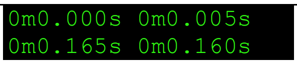
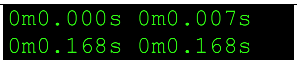
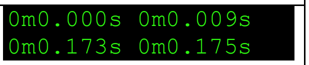
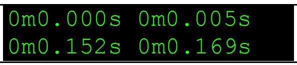
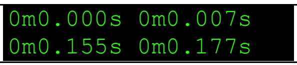
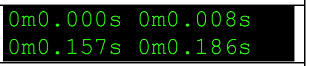
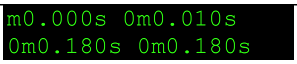
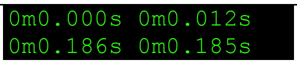
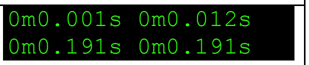
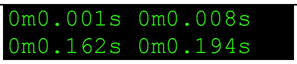
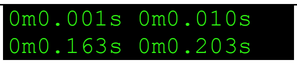
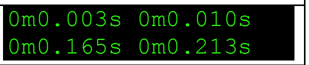
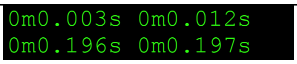
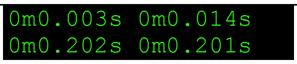
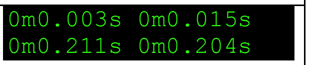
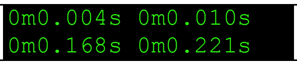
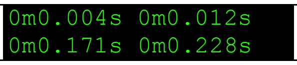
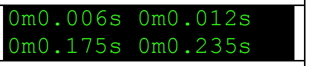
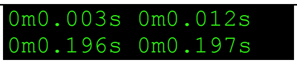
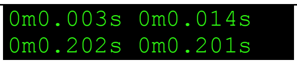
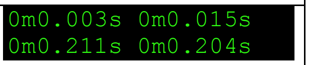
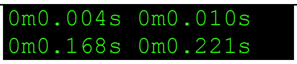
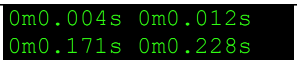
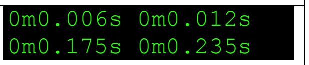
Nathan Tjoar
005081232

Lab 1

Data Gathered:

Test 1:			
	Benchmark 1:	Benchmark 2:	Benchmark 3:
Bash	0m0.000s 0m0.005s 0m0.165s 0m0.160s	0m0.000s 0m0.007s 0m0.168s 0m0.168s	0m0.000s 0m0.009s 0m0.173s 0m0.175s
Dash	0m0.000s 0m0.005s 0m0.152s 0m0.169s	0m0.000s 0m0.007s 0m0.155s 0m0.177s	0m0.000s 0m0.008s 0m0.157s 0m0.186s
simpsh	0.003828s 0.002203s	0.003546s 0.001874s	0.004584s 0.002472s
Test 2:			
	Benchmark 1:	Benchmark 2:	Benchmark 3:
Bash	m0.000s 0m0.010s 0m0.180s 0m0.180s	0m0.000s 0m0.012s 0m0.186s 0m0.185s	0m0.001s 0m0.012s 0m0.191s 0m0.191s
Dash	0m0.001s 0m0.008s 0m0.162s 0m0.194s	0m0.001s 0m0.010s 0m0.163s 0m0.203s	0m0.003s 0m0.010s 0m0.165s 0m0.213s
simpsh	0.004s 0.002297s	0.003597s 0.002017s	0.004331s 0.002571s
Test 3:			
	Benchmark 1:	Benchmark 2:	Benchmark 3:
Bash	0m0.003s 0m0.012s 0m0.196s 0m0.197s	0m0.003s 0m0.014s 0m0.202s 0m0.201s	0m0.003s 0m0.015s 0m0.211s 0m0.204s
Dash	0m0.004s 0m0.010s 0m0.168s 0m0.221s	0m0.004s 0m0.012s 0m0.171s 0m0.228s	0m0.006s 0m0.012s 0m0.175s 0m0.235s
simpsh	0.003787s 0.002177s	0.003651s 0.001961s	0.004117s 0.002615s
Averages:			
	Benchmark 1:	Benchmark 2:	Benchmark 3:
Bash	0m0.003s 0m0.012s 0m0.196s 0m0.197s	0m0.003s 0m0.014s 0m0.202s 0m0.201s	0m0.003s 0m0.015s 0m0.211s 0m0.204s
Dash	0m0.004s 0m0.010s 0m0.168s 0m0.221s	0m0.004s 0m0.012s 0m0.171s 0m0.228s	0m0.006s 0m0.012s 0m0.175s 0m0.235s
simpsh	0.003872s 0.002225s	0.003598s 0.001951s	0.004344s 0.002553s

Conclusions:

In looking at the data values, it is clear that our Dash script is far less optimized to handle file input-output tan our bash script. Furthermore, we see that our C program greatly surpasses the two other methods in terms of efficiency. For C, this is thanks to gcc optimizations that allow our code—for the most part—to outdo and outrun bash scripts. For bash, this may be due to some slightly less compatible code that wasn't fully translated from its bash equivalent.

Benchmark Code:

Bash/Dash:

```
sort -g a0.txt | tr a-z A-Z | cat > test1_1out.txt times >
bash1_1.txt
```

Simpsh:

```
./simpsh --rdonly a0.txt --creat --wronly test1_1out.txt --creat
--rdwr test1_1err.txt \
--pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2
tr a-z A-Z --command 5 1 2 cat \
--close 3 --close 4 --close 5 --close 6 --wait >c1_1time.txt
2>c1_1err.txt
```

Benchmark 2 Code:

Bash/Dash:

```
cat a0.txt | grep "bottle" | wc -c > test1_2out.txt times >
bash1_2.txt
```

Simpsh:

```
./simpsh --rdonly a0.txt --creat --wronly test1_2out.txt --creat
--rdwr test1_2err.txt \ --pipe --pipe --profile --command 0 4 2
cat - --command 3 6 2 grep "bottle" --command 5 1 2 wc -c \ --
close 3 --close 4 --close 5 --close 6 --wait >c1_2time.txt
2>c1_2err.txt
```

Benchmark 3 Code:

Bash/Dash:

```
sort -g a0.txt | cat | grep "vessel" > test1_3out.txt times >
bash1_3.txt
```

Simpsh:

```
./simpsh --rdonly a0.txt --creat --wronly test1_3out.txt --creat
--rdwr test1_3err.txt \
--pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2
cat - --command 5 1 2 grep "vessel" \
--close 3 --close 4 --close 5 --close 6 --wait >c1_3time.txt
2>c1_3err.txt
```

bash script:

```
#!/bin/bash
if [ -d bashTimes ]
then
   rm -rf bashTimes
fi
if [ -d cTimes ]
then
   rm -rf cTimes
```

```
fi

if [ "${PATH:0:16}" == "/usr/local/cs/bin" ]
then
  true
else
  PATH=/usr/local/cs/bin:$PATH
fi

if ps | grep "simpsh"
then
  echo "simpsh is running in background."
  echo "Testing cannot continue."
  echo "Kill it and then run the script."
  exit 1
fi

if [ ! -e Makefile ]
then
    echo "No file to make"
    exit 1
fi
make

cat > a0.txt <<'EOF'
minor
crutch
fashionable
extent
study
discriminate
depart
cupboard
ceiling
vessel
ribbon
develop
```

```
brilliance
write
prisoner
emergency
soldier
tourist
circle
bottle
EOF

### Interation I ###
# Test case 1
echo "Test 1.1"
sort -g a0.txt | tr a-z A-Z | cat > test1_1out.txt ; times > bash1_1.txt
./simpsh --rdonly a0.txt --creat --wronly test1_1out.txt --creat --rdwr test1_1err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 tr a-z A-Z --command 5 1 2 cat \
  --close 3 --close 4 --close 5 --close 6 --wait >c1_1time.txt 2>c1_1err.txt
rm -rf test1_1* c1_1err*
echo "File outputs to bash1_1.txt and c1_1time.txt"
echo "---"

# Test case 2
echo "Test 1.2"
cat a0.txt | grep "bottle" | wc -c > test1_2out.txt ; times > bash1_2.txt
./simpsh --rdonly a0.txt --creat --wronly test1_2out.txt --creat --rdwr test1_2err.txt \
  --pipe --pipe --profile --command 0 4 2 cat - --command 3 6 2 grep "bottle" --command 5 1 2 wc -c \
  --close 3 --close 4 --close 5 --close 6 --wait >c1_2time.txt 2>c1_2err.txt
rm -rf test1_2* c1_2err*
echo "File outputs to bash1_2.txt and c1_2time.txt"
echo "---"

# Test case 3
echo "Test 1.3"
sort -g a0.txt | cat | grep "vessel" > test1_3out.txt ; times > bash1_3.txt
./simpsh --rdonly a0.txt --creat --wronly test1_3out.txt --creat --rdwr test1_3err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 cat - --command 5 1 2 grep "vessel" \
  --close 3 --close 4 --close 5 --close 6 --wait >c1_3time.txt 2>c1_3err.txt
```

```
rm -rf test1_3* c1_3err*
echo "File outputs to bash1_3.txt and c1_3time.txt"
echo "---"

### Interation II ###
# Test case 1
echo "Test 2.1"
sort -g a0.txt | tr a-z A-Z | cat > test2_1out.txt ; times bash > bash2_1.txt
./simpsh --rdonly a0.txt --creat --wronly test2_1out.txt --creat --rdwr test2_1err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 tr a-z A-Z --command 5 1 2 cat \
  --close 3 --close 4 --close 5 --close 6 --wait >c2_1time.txt 2>c2_1err.txt
rm -rf test2_1* c2_1err*
echo "File outputs to bash2_1.txt and c2_1time.txt"
echo "---"

# Test case 2
echo "Test 2.2"
cat a0.txt | grep "bottle" | wc -c > test2_2out.txt ; times bash > bash2_2.txt
./simpsh --rdonly a0.txt --creat --wronly test2_2out.txt --creat --rdwr test2_2err.txt \
  --pipe --pipe --profile --command 0 4 2 cat - --command 3 6 2 grep "bottle" --command 5 1 2 wc -c \
  --close 3 --close 4 --close 5 --close 6 --wait >c2_2time.txt 2>c2_2err.txt
rm -rf test2_2* c2_2err*
echo "File outputs to bash2_2.txt and c2_2time.txt"
echo "---"

# Test case 3
echo "Test 2.3"
sort -g a0.txt | cat | grep "vessel" > test2_3out.txt ; times bash > bash2_3.txt
./simpsh --rdonly a0.txt --creat --wronly test2_3out.txt --creat --rdwr test2_3err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 cat - --command 5 1 2 grep "vessel" \
  --close 3 --close 4 --close 5 --close 6 --wait >c2_3time.txt 2>c2_3err.txt
rm -rf test2_3* c2_3err*
echo "File outputs to bash2_3.txt and c2_3time.txt"
echo "---"

### Interation III ###
# Test case 1
# Test case 1
echo "Test 3.1"
sort -g a0.txt | tr a-z A-Z | cat > test3_1out.txt ; times bash > bash3_1.txt
./simpsh --rdonly a0.txt --creat --wronly test3_1out.txt --creat --rdwr test3_1err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 tr a-z A-Z --command 5 1 2 cat \
  --close 3 --close 4 --close 5 --close 6 --wait >c3_1time.txt 2>c3_1err.txt
rm -rf test3_1* c3_1err*
echo "File outputs to bash3_1.txt and c3_1time.txt"
echo "---"

# Test case 2
echo "Test 3.2"
cat a0.txt | grep "bottle" | wc -c > test3_2out.txt ; times > bash3_2.txt
./simpsh --rdonly a0.txt --creat --wronly test3_2out.txt --creat --rdwr test3_2err.txt \
  --pipe --pipe --profile --command 0 4 2 cat - --command 3 6 2 grep "bottle" --command 5 1 2 wc -c \
  --close 3 --close 4 --close 5 --close 6 --wait >c3_2time.txt 2>c3_2err.txt
rm -rf test3_2* c3_2err*
echo "File outputs to bash3_2.txt and c3_2time.txt"
echo "---"

# Test case 3
echo "Test 3.3"
sort -g a0.txt | cat | grep "vessel" > test3_3out.txt ; times > bash3_3.txt
./simpsh --rdonly a0.txt --creat --wronly test3_3out.txt --creat --rdwr test3_3err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 cat - --command 5 1 2 grep "vessel" \
  --close 3 --close 4 --close 5 --close 6 --wait >c3_3time.txt 2>c3_3err.txt
rm -rf test3_3* c3_3err*
echo "File outputs to bash3_3.txt and c3_3time.txt"
echo "---"

rm -f a0.txt

echo "All c files in c; times folder"
echo "All bash files in bashTimes folder"
mkdir cTimes
mkdir bashTimes
mv c*time.txt cTimes
mv bash*.txt bashTimes
```

dash script:

```
#!/bin/sh
if [ -d dashTimes ]
then
    rm -rf dashTimes
fi
if [ -d cTimes ]
then
    rm -rf cTimes
fi

if [ "${PATH:0:16}" == "/usr/local/cs/bin" ]
then
  true
else
  PATH=/usr/local/cs/bin:$PATH
fi

if ps | grep "simpsh"
then
  echo "simpsh is running in background."
  echo "Testing cannot continue."
  echo "Kill it and then run the script."
  exit 1
fi

if [ ! -e Makefile ]
then
    echo "No file to make"
    exit 1
fi
make

cat > a0.txt <<'EOF'
minor
```

```
crutch
fashionable
extent
study
discriminate
depart
cupboard
ceiling
vessel
ribbon
develop
brilliance
write
prisoner
emergency
soldier
tourist
circle
bottle
EOF

### Interation I ###
# Test case 1
echo "Test 1.1"
sort -g a0.txt | tr a-z A-Z | cat > test1_1out.txt ; times > dash1_1.txt
./simpsh --rdonly a0.txt --creat --wronly test1_1out.txt --creat --rdwr test1_1err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 tr a-z A-Z --command 5 1 2 cat \
  --close 3 --close 4 --close 5 --close 6 --wait >c1_1time.txt 2>c1_1err.txt
rm -rf test1_1* c1_1err*
echo "File outputs to dash1_1.txt and c1_1time.txt"
echo "---"

# Test case 2
echo "Test 1.2"
cat a0.txt | grep "bottle" | wc -c > test1_2out.txt ; times > dash1_2.txt
./simpsh --rdonly a0.txt --creat --wronly test1_2out.txt --creat --rdwr test1_2err.txt \
  --pipe --pipe --profile --command 0 4 2 cat - --command 3 6 2 grep "bottle" --command 5 1 2 wc -c \
  --close 3 --close 4 --close 5 --close 6 --wait >c1_2time.txt 2>c1_2err.txt
rm -rf test1_2* c1_2err*
echo "File outputs to dash1_2.txt and c1_2time.txt"
echo "---"

# Test case 3
echo "Test 1.3"
sort -g a0.txt | cat | grep "vessel" > test1_3out.txt ; times > dash1_3.txt
./simpsh --rdonly a0.txt --creat --wronly test1_3out.txt --creat --rdwr test1_3err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 cat - --command 5 1 2 grep "vessel" \
  --close 3 --close 4 --close 5 --close 6 --wait >c1_3time.txt 2>c1_3err.txt
rm -rf test1_3* c1_3err*
echo "File outputs to dash1_3.txt and c1_3time.txt"
echo "---"

### Interation II ###
# Test case 1
echo "Test 2.1"
sort -g a0.txt | tr a-z A-Z | cat > test2_1out.txt ; times bash > dash2_1.txt
./simpsh --rdonly a0.txt --creat --wronly test2_1out.txt --creat --rdwr test2_1err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 tr a-z A-Z --command 5 1 2 cat \
  --close 3 --close 4 --close 5 --close 6 --wait >c2_1time.txt 2>c2_1err.txt
rm -rf test2_1* c2_1err*
echo "File outputs to dash2_1.txt and c2_1time.txt"
echo "---"

# Test case 2
echo "Test 2.2"
cat a0.txt | grep "bottle" | wc -c > test2_2out.txt ; times bash > dash2_2.txt
./simpsh --rdonly a0.txt --creat --wronly test2_2out.txt --creat --rdwr test2_2err.txt \
  --pipe --pipe --profile --command 0 4 2 cat - --command 3 6 2 grep "bottle" --command 5 1 2 wc -c \
  --close 3 --close 4 --close 5 --close 6 --wait >c2_2time.txt 2>c2_2err.txt
rm -rf test2_2* c2_2err*
echo "File outputs to dash2_2.txt and c2_2time.txt"
echo "---"

# Test case 3
```

```
echo "Test 2.3"
sort -g a0.txt | cat | grep "vessel" > test2_3out.txt ; times bash > dash2_3.txt
./simpsh --rdonly a0.txt --creat --wronly test2_3out.txt --creat --rdwr test2_3err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 cat - --command 5 1 2 grep "vessel" \
  --close 3 --close 4 --close 5 --close 6 --wait >c2_3time.txt 2>c2_3err.txt
rm -rf test2_3* c2_3err*
echo "File outputs to dash2_3.txt and c2_3time.txt"
echo "---"

### Interation III ###
# Test case 1
# Test case 1
echo "Test 3.1"
sort -g a0.txt | tr a-z A-Z | cat > test3_1out.txt ; times bash > dash3_1.txt
./simpsh --rdonly a0.txt --creat --wronly test3_1out.txt --creat --rdwr test3_1err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 tr a-z A-Z --command 5 1 2 cat \
  --close 3 --close 4 --close 5 --close 6 --wait >c3_1time.txt 2>c3_1err.txt
rm -rf test3_1* c3_1err*
echo "File outputs to dash3_1.txt and c3_1time.txt"
echo "---"

# Test case 2
echo "Test 3.2"
cat a0.txt | grep "bottle" | wc -c > test3_2out.txt ; times > dash3_2.txt
./simpsh --rdonly a0.txt --creat --wronly test3_2out.txt --creat --rdwr test3_2err.txt \
  --pipe --pipe --profile --command 0 4 2 cat - --command 3 6 2 grep "bottle" --command 5 1 2 wc -c \
  --close 3 --close 4 --close 5 --close 6 --wait >c3_2time.txt 2>c3_2err.txt
rm -rf test3_2* c3_2err*
echo "File outputs to dash3_2.txt and c3_2time.txt"
echo "---"

# Test case 3
echo "Test 3.3"
sort -g a0.txt | cat | grep "vessel" > test3_3out.txt ; times > dash3_3.txt
./simpsh --rdonly a0.txt --creat --wronly test3_3out.txt --creat --rdwr test3_3err.txt \
  --pipe --pipe --profile --command 0 4 2 sort -g --command 3 6 2 cat - --command 5 1 2 grep "vessel" \
  --close 3 --close 4 --close 5 --close 6 --wait >c3_3time.txt 2>c3_3err.txt
```

```
rm -rf test3_3* c3_3err*
echo "File outputs to dash3_3.txt and c3_3time.txt"
echo "---"

rm -f a0.txt

echo "All c files in c; times folder"
echo "All dash files in dashTimes folder"
mkdir cTimes
mkdir dashTimes
mv c*time.txt cTimes
mv dash*.txt dashTimes
```