Assignment 5

Software Tools and Technologies

```
Name : Ravish Ranjan
Course : MCA
Semester : 1st semester
```

Objective

```
Debug, log, and measure performance of a bash script.
```

Step 1: Create a Sample Script

1. Create a script named process_data.sh:

```
touch proqcess_data.sh
```

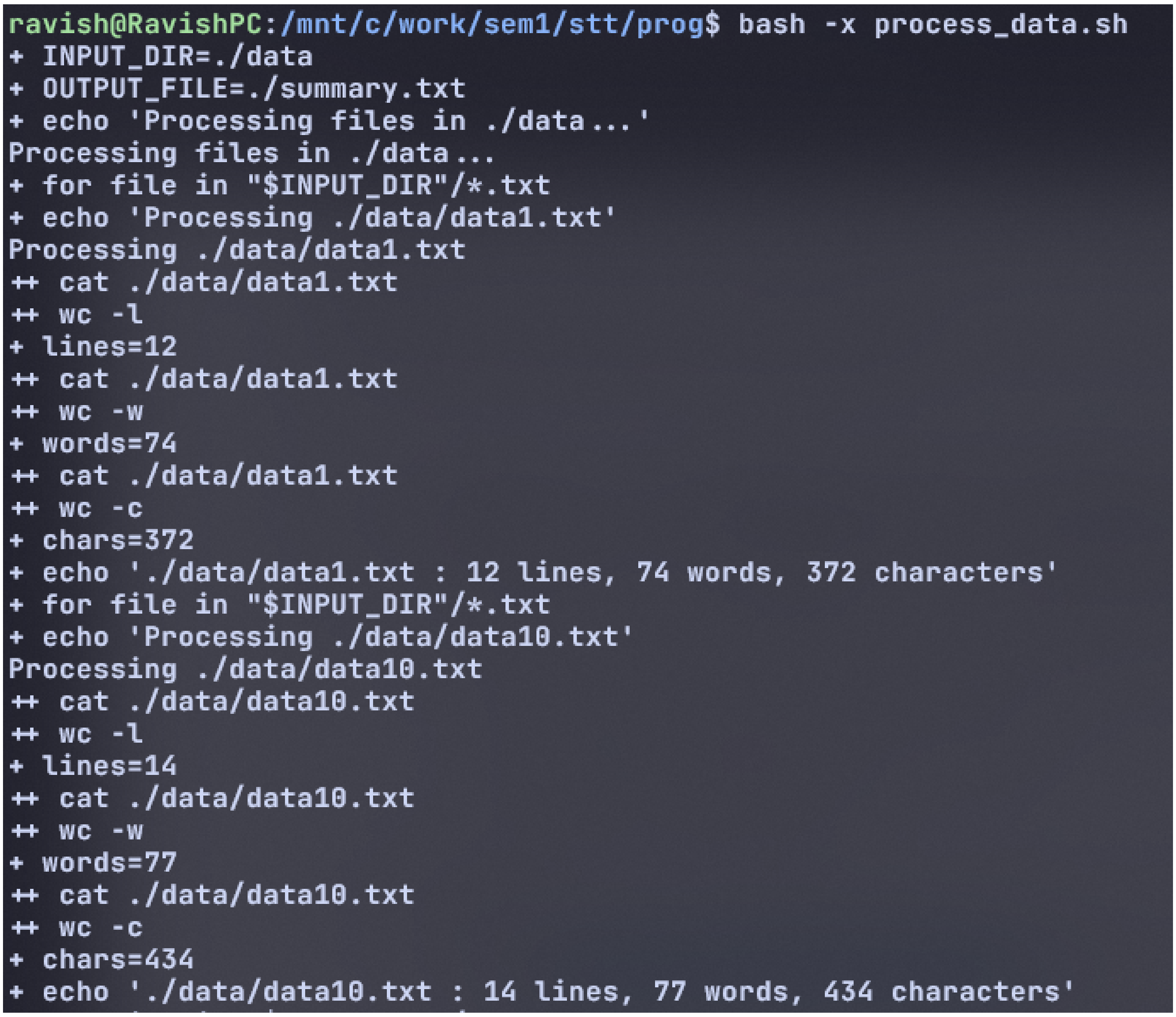
```
cat process_data.sh
```

```
#!/bin/bash

INPUT_DIR="./data"
OUTPUT_FILE="./summary.txt"

echo "Processing files in $INPUT_DIR..."
> "$OUTPUT_FILE"

for file in "$INPUT_DIR"/*.txt; do
        echo "Processing $file"
        lines=$( cat "$file" | wc -l )
        words=$( cat "$file" | wc -w )
        chars=$( cat "$file" | wc -c )
        echo "$file : $lines lines, $words words, $chars characters" >>
"$OUTPUT_FILE"
done

echo "All files processed"
```

Step 2: Debuging the bash file

1. Debugging whole file

```
bash -x process_data . sh
```



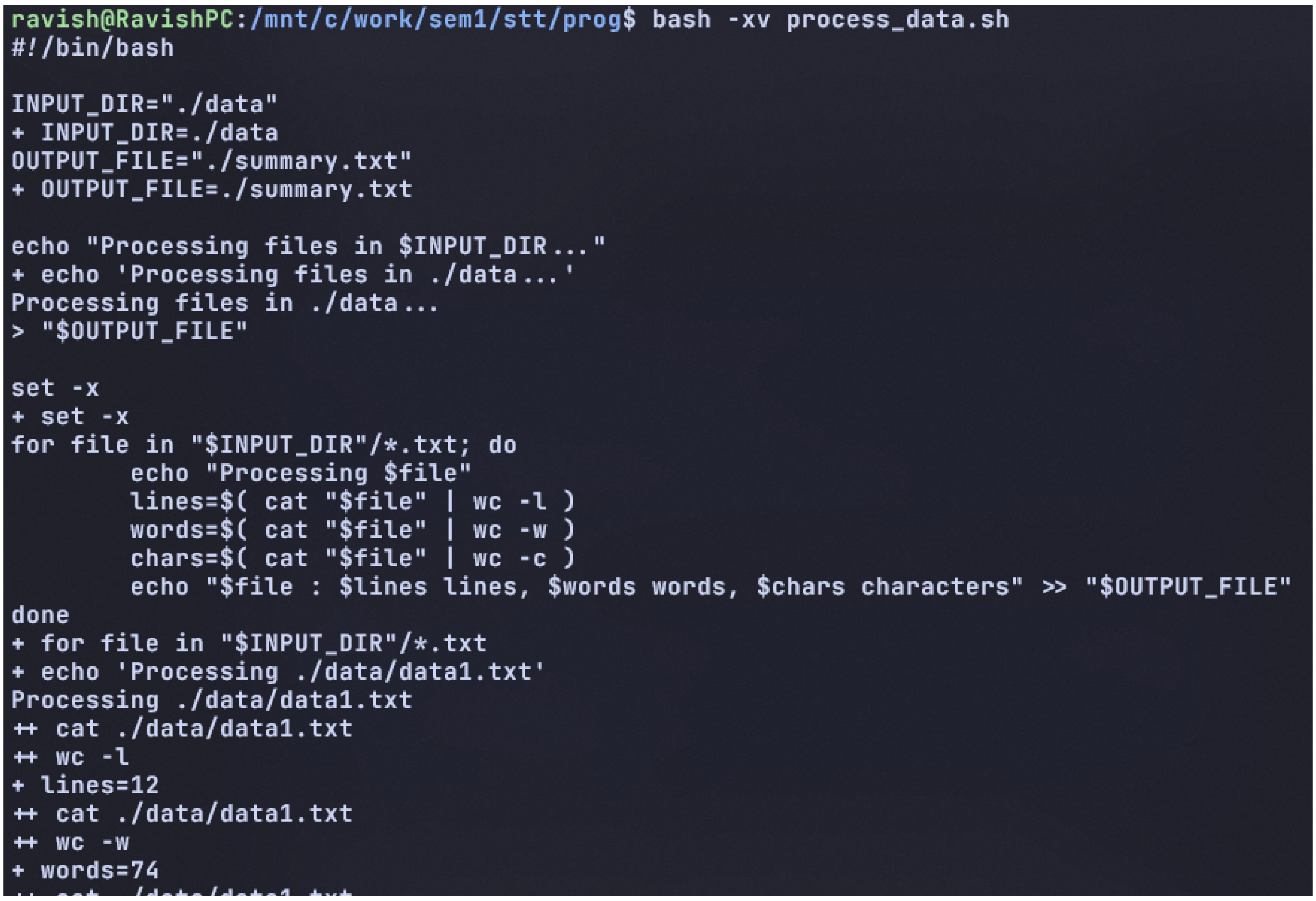
```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ bash -x process_data.sh
+ INPUT_DIR=./data
+ OUTPUT_FILE=./summary.txt
+ echo 'Processing files in ./data...'
Processing files in ./data...
+ for file in "$INPUT_DIR"/*.txt
+ echo 'Processing ./data/data1.txt'
Processing ./data/data1.txt
++ cat ./data/data1.txt
++ wc -l
+ lines=12
++ cat ./data/data1.txt
++ wc -w
+ words=74
++ cat ./data/data1.txt
++ wc -c
+ chars=372
+ echo './data/data1.txt : 12 lines, 74 words, 372 characters'
+ for file in "$INPUT_DIR"/*.txt
+ echo 'Processing ./data/data10.txt'
Processing ./data/data10.txt
++ cat ./data/data10.txt
++ wc -l
+ lines=14
++ cat ./data/data10.txt
++ wc -w
+ words=77
++ cat ./data/data10.txt
++ wc -c
+ chars=434
+ echo './data/data10.txt : 14 lines, 77 words, 434 characters'
```

2. Debugging only the loop

```
#!/bin/bash

INPUT_DIR="./data"
OUTPUT_FILE="./summary.txt"

echo "Processing files in $INPUT_DIR..."
> "$OUTPUT_FILE"

set -x
for file in "$INPUT_DIR"/*.txt; do
```

```
        echo "Processing $file"
        lines=$( cat "$file" | wc -l )
        words=$( cat "$file" | wc -w )
        chars=$( cat "$file" | wc -c )
        echo "$file : $lines lines, $words words, $chars characters" >>
"$OUTPUT_FILE"
done
set +x

echo "All files processed"
```

```
bash -xv process_data.sh
```



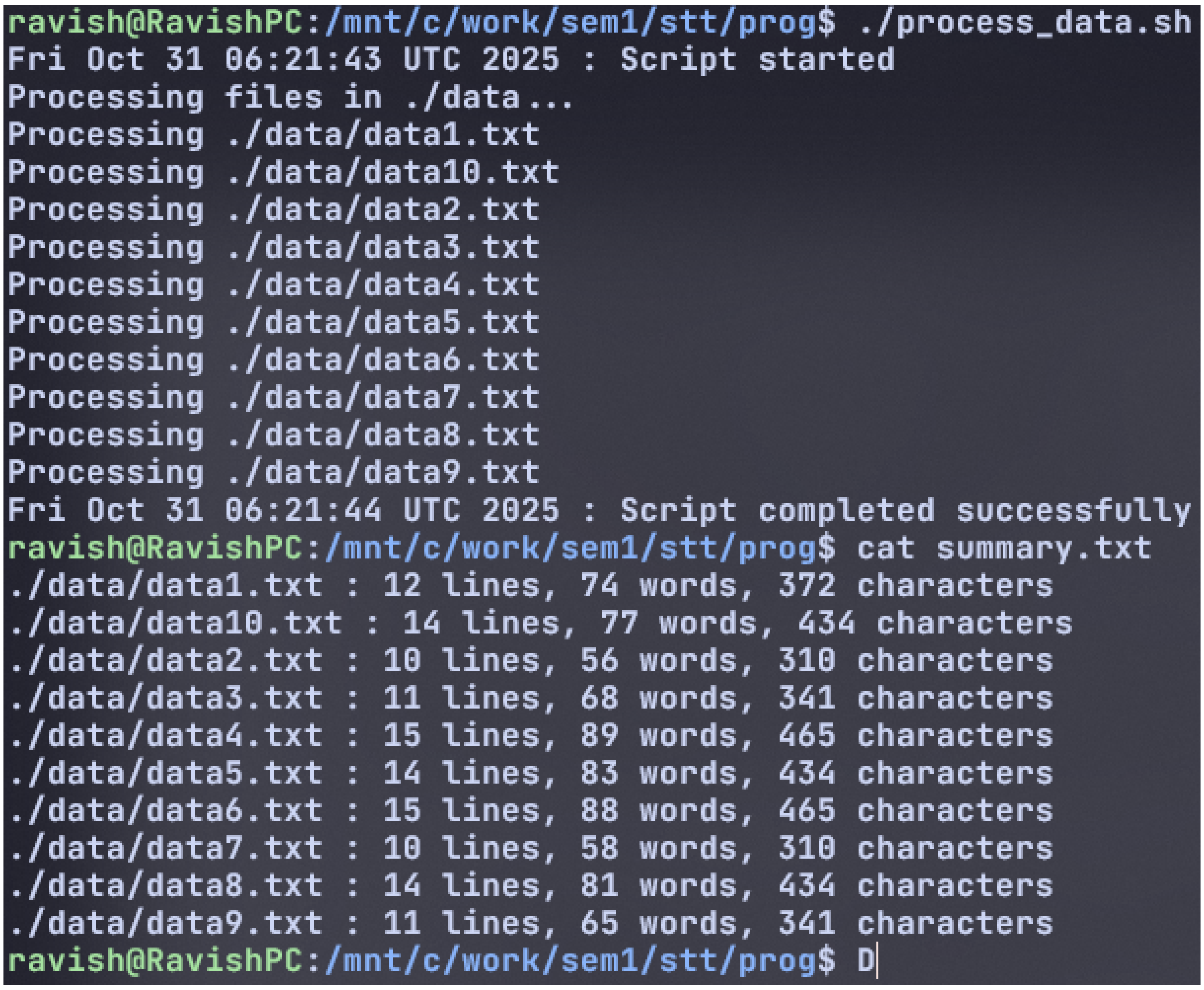
```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ bash -xv process_data.sh
#!/bin/bash

INPUT_DIR="./data"
+ INPUT_DIR=./data
OUTPUT_FILE="./summary.txt"
+ OUTPUT_FILE=./summary.txt

echo "Processing files in $INPUT_DIR..."
+ echo 'Processing files in ./data...'
Processing files in ./data...
> "$OUTPUT_FILE"

set -x
+ set -x
for file in "$INPUT_DIR"/*.txt; do
        echo "Processing $file"
        lines=$( cat "$file" | wc -l )
        words=$( cat "$file" | wc -w )
        chars=$( cat "$file" | wc -c )
        echo "$file : $lines lines, $words words, $chars characters" >> "$OUTPUT_FILE"
done
+ for file in "$INPUT_DIR"/*.txt
+ echo 'Processing ./data/data1.txt'
Processing ./data/data1.txt
++ cat ./data/data1.txt
++ wc -l
+ lines=12
++ cat ./data/data1.txt
++ wc -w
+ words=74
```

Step 3: Add Logging and Error Handling

```
#!/bin/bash

LOGFILE="run.log"
ERRORLOG="error.log"
exec > >(tee -a "$LOGFILE" ) 2> >(tee -a "$ERRORLOG" >&2)
echo "$( date ) : Script started "

INPUT_DIR="./data"
OUTPUT_FILE="./summary.txt"
```

```
if [ ! -d "$INPUT_DIR" ]; then
        echo "$( date ) : ERROR - Input directory not found : $INPUT_DIR " >&2
        exit 1
fi
> "$OUTPUT_FILE"

echo "Processing files in $INPUT_DIR..."
> "$OUTPUT_FILE"

for file in "$INPUT_DIR"/*.txt; do
        if [ ! -f "$file" ]; then
                echo "$( date ) : WARNING - No file found" >&2
                continue
        fi
        echo "Processing $file"
        lines=$( cat "$file" | wc -l )
        words=$( cat "$file" | wc -w )
        chars=$( cat "$file" | wc -c )
        echo "$file : $lines lines, $words words, $chars characters" >>
"$OUTPUT_FILE"
done

echo "$( date ) : Script completed successfully"
```

```
./process_data.sh
cat summary.txt
```

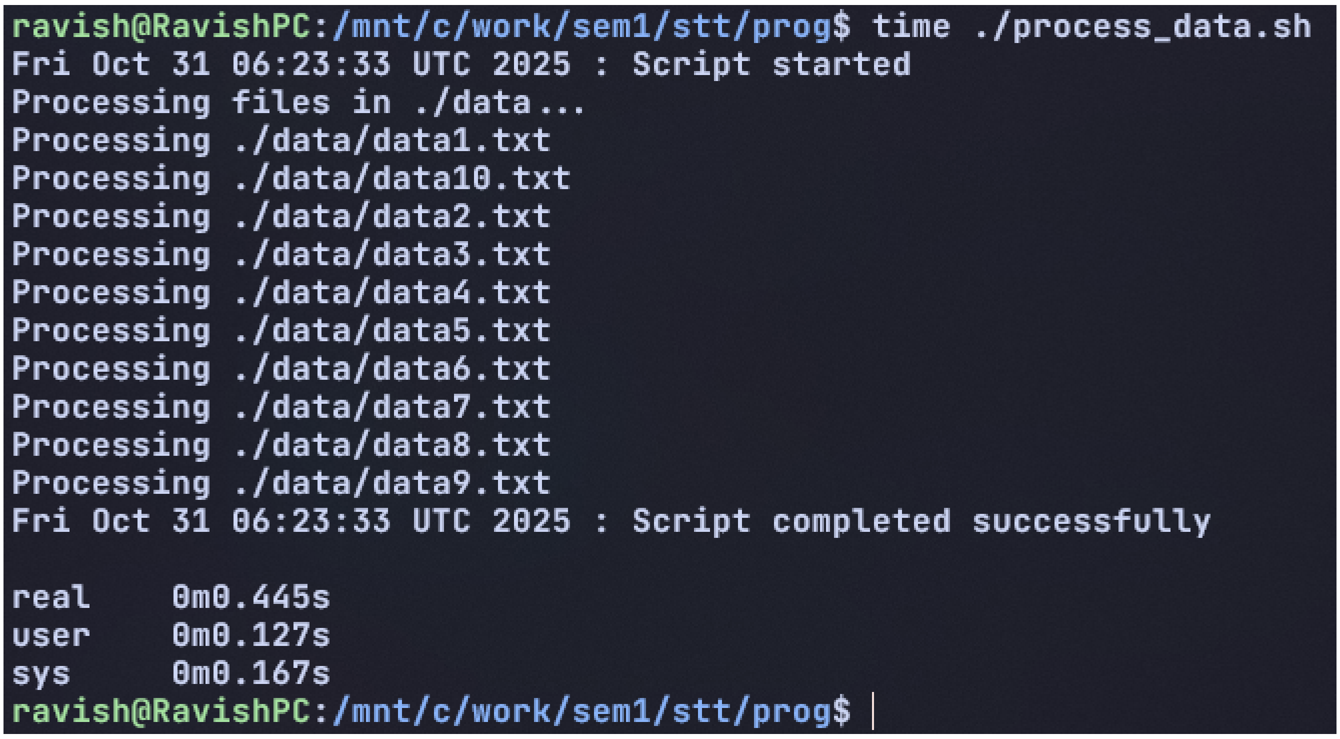


```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./process_data.sh
Fri Oct 31 06:21:43 UTC 2025 : Script started
Processing files in ./data...
Processing ./data/data1.txt
Processing ./data/data10.txt
Processing ./data/data2.txt
Processing ./data/data3.txt
Processing ./data/data4.txt
Processing ./data/data5.txt
Processing ./data/data6.txt
Processing ./data/data7.txt
Processing ./data/data8.txt
Processing ./data/data9.txt
Fri Oct 31 06:21:44 UTC 2025 : Script completed successfully
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ cat summary.txt
./data/data1.txt : 12 lines, 74 words, 372 characters
./data/data10.txt : 14 lines, 77 words, 434 characters
./data/data2.txt : 10 lines, 56 words, 310 characters
./data/data3.txt : 11 lines, 68 words, 341 characters
./data/data4.txt : 15 lines, 89 words, 465 characters
./data/data5.txt : 14 lines, 83 words, 434 characters
./data/data6.txt : 15 lines, 88 words, 465 characters
./data/data7.txt : 10 lines, 58 words, 310 characters
./data/data8.txt : 14 lines, 81 words, 434 characters
./data/data9.txt : 11 lines, 65 words, 341 characters
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ D
```

Step 4: Measure Runtime and Performance

1. Measure execution time

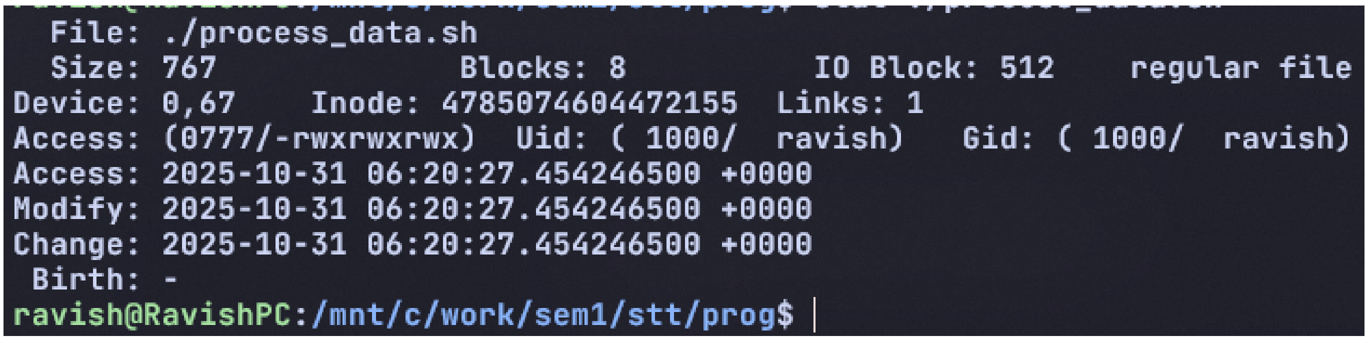
```
time ./process_data.sh
```



```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ time ./process_data.sh
Fri Oct 31 06:23:33 UTC 2025 : Script started
Processing files in ./data...
Processing ./data/data1.txt
Processing ./data/data10.txt
Processing ./data/data2.txt
Processing ./data/data3.txt
Processing ./data/data4.txt
Processing ./data/data5.txt
Processing ./data/data6.txt
Processing ./data/data7.txt
Processing ./data/data8.txt
Processing ./data/data9.txt
Fri Oct 31 06:23:33 UTC 2025 : Script completed successfully

real    0m0.445s
user    0m0.127s
sys     0m0.167s
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$
```

2. Profile system performance

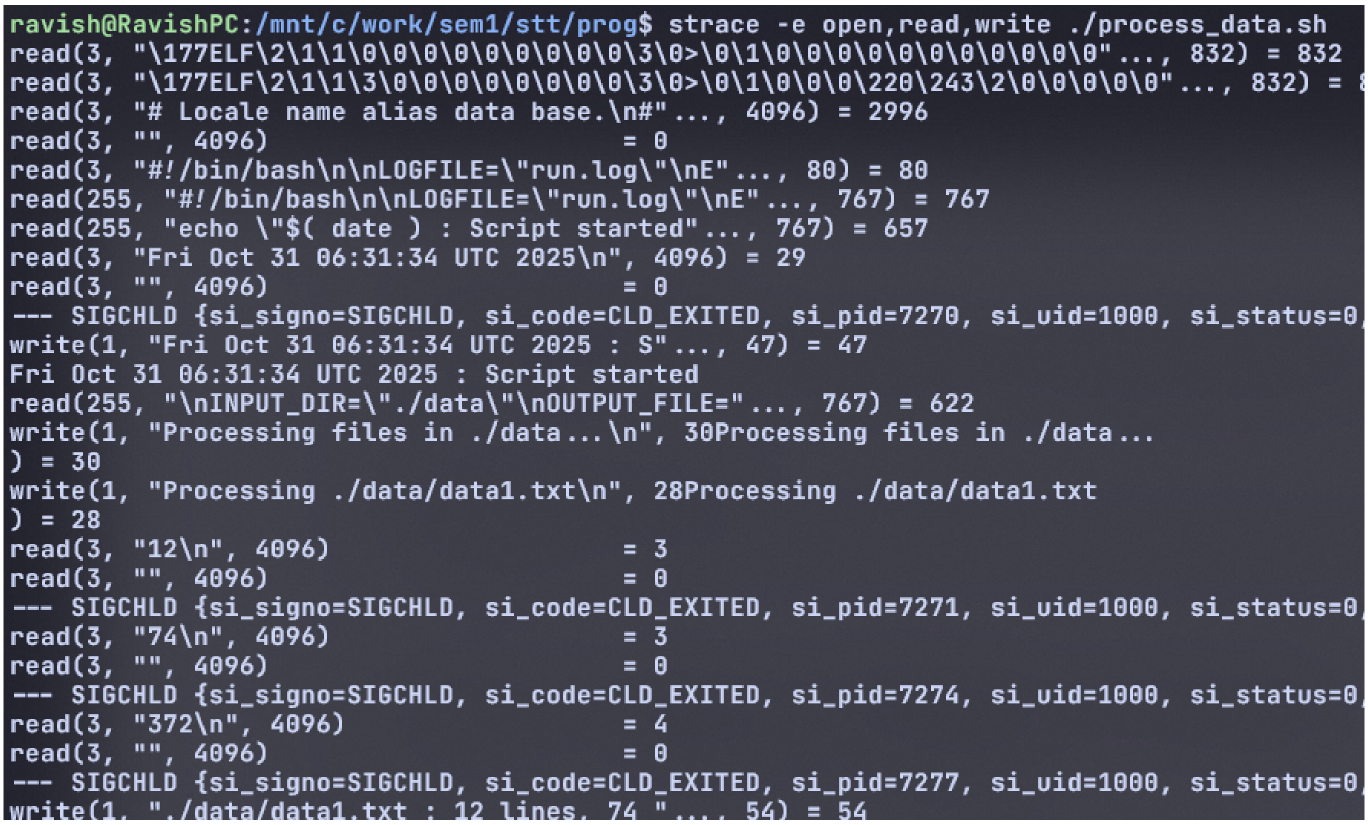
```
perf stat ./process_data.sh
```



```
  File: ./process_data.sh
  Size: 767             Blocks: 8          IO Block: 512    regular file
Device: 0,67    Inode: 4785074604472155  Links: 1
Access: (0777/-rwxrwxrwx)  Uid: ( 1000/  ravish)   Gid: ( 1000/  ravish)
Access: 2025-10-31 06:20:27.454246500 +0000
Modify: 2025-10-31 06:20:27.454246500 +0000
Change: 2025-10-31 06:20:27.454246500 +0000
 Birth: -
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$
```

3. Strace Command

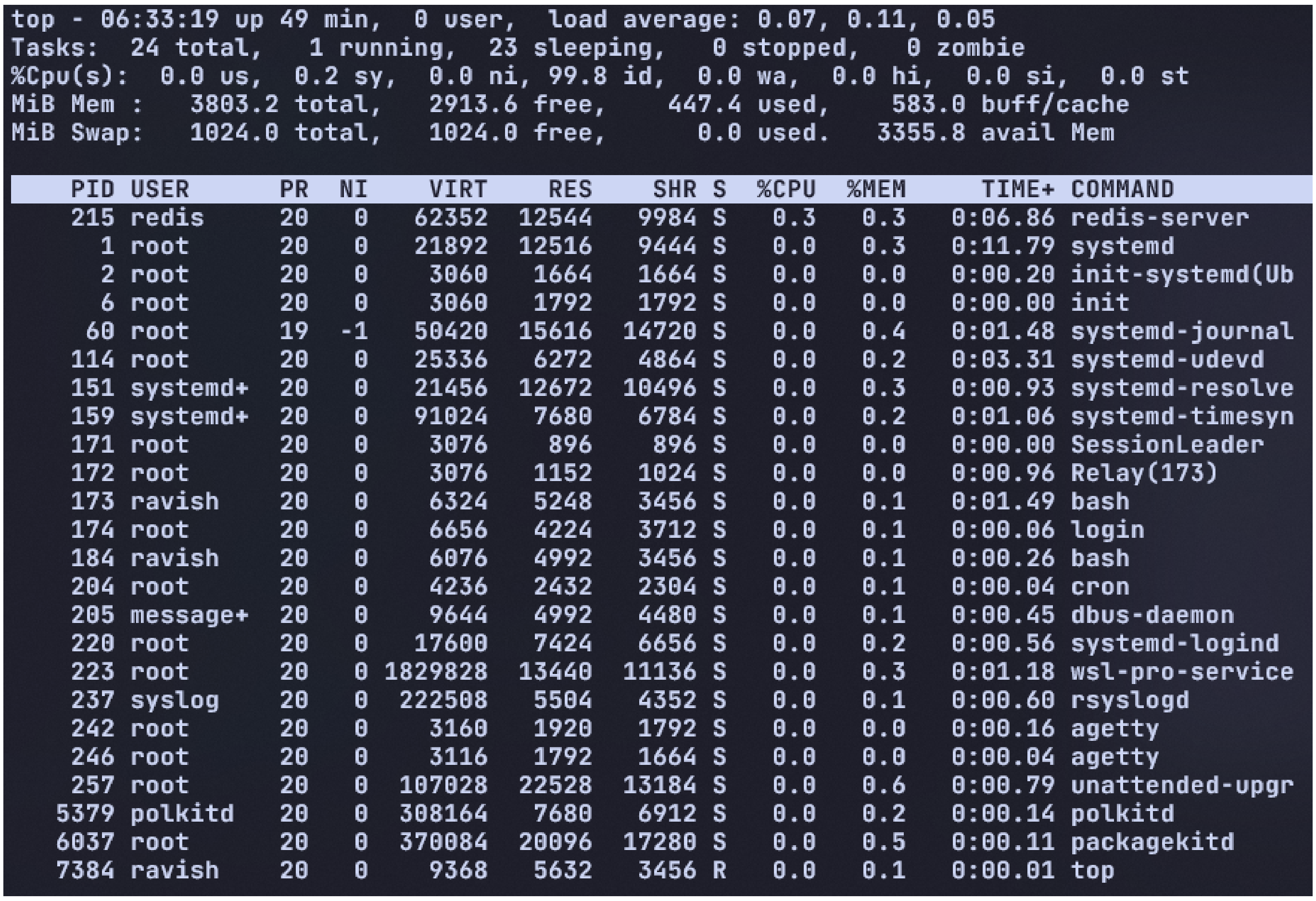
```
strace -e open,read,write ./process_data.sh
```



```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ strace -e open,read,write ./process_data.sh
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 8
read(3, "# Locale name alias data base.\n#"..., 4096) = 2996
read(3, "", 4096)                       = 0
read(3, "#!/bin/bash\n\nLOGFILE=\"run.log\"\nE"..., 80) = 80
read(255, "#!/bin/bash\n\nLOGFILE=\"run.log\"\nE"..., 767) = 767
read(255, "echo \"$( date ) : Script started"..., 767) = 657
read(3, "Fri Oct 31 06:31:34 UTC 2025\n", 4096) = 29
read(3, "", 4096)                       = 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7270, si_uid=1000, si_status=0,
write(1, "Fri Oct 31 06:31:34 UTC 2025 : S"..., 47) = 47
Fri Oct 31 06:31:34 UTC 2025 : Script started
read(255, "\nINPUT_DIR=\"./data\"\nOUTPUT_FILE="..., 767) = 622
write(1, "Processing files in ./data...\n", 30Processing files in ./data...
) = 30
write(1, "Processing ./data/data1.txt\n", 28Processing ./data/data1.txt
) = 28
read(3, "12\n", 4096)                   = 3
read(3, "", 4096)                       = 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7271, si_uid=1000, si_status=0,
read(3, "74\n", 4096)                   = 3
read(3, "", 4096)                       = 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7274, si_uid=1000, si_status=0,
read(3, "372\n", 4096)                  = 4
read(3, "", 4096)                       = 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7277, si_uid=1000, si_status=0,
write(1, "./data/data1.txt : 12 lines, 74 "..., 54) = 54
```

4. CPU observation

```
top
```



```
top - 06:33:19 up 49 min,  0 user,  load average: 0.07, 0.11, 0.05
Tasks:  24 total,   1 running,  23 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.2 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3803.2 total,   2913.6 free,    447.4 used,    583.0 buff/cache
MiB Swap:   1024.0 total,   1024.0 free,      0.0 used.   3355.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    215 redis     20   0   62352  12544   9984 S   0.3   0.3   0:06.86 redis-server
      1 root      20   0   21892  12516   9444 S   0.0   0.3   0:11.79 systemd
      2 root      20   0    3060   1664   1664 S   0.0   0.0   0:00.20 init-systemd(Ub
      6 root      20   0    3060   1792   1792 S   0.0   0.0   0:00.00 init
     60 root      19  -1   50420  15616  14720 S   0.0   0.4   0:01.48 systemd-journal
    114 root      20   0   25336   6272   4864 S   0.0   0.2   0:03.31 systemd-udevd
    151 systemd+  20   0   21456  12672  10496 S   0.0   0.3   0:00.93 systemd-resolve
    159 systemd+  20   0   91024   7680   6784 S   0.0   0.2   0:01.06 systemd-timesyn
    171 root      20   0    3076    896    896 S   0.0   0.0   0:00.00 SessionLeader
    172 root      20   0    3076   1152   1024 S   0.0   0.0   0:00.96 Relay(173)
    173 ravish    20   0    6324   5248   3456 S   0.0   0.1   0:01.49 bash
    174 root      20   0    6656   4224   3712 S   0.0   0.1   0:00.06 login
    184 ravish    20   0    6076   4992   3456 S   0.0   0.1   0:00.26 bash
    204 root      20   0    4236   2432   2304 S   0.0   0.1   0:00.04 cron
    205 message+  20   0    9644   4992   4480 S   0.0   0.1   0:00.45 dbus-daemon
    220 root      20   0   17600   7424   6656 S   0.0   0.2   0:00.56 systemd-logind
    223 root      20   0 1829828  13440  11136 S   0.0   0.3   0:01.18 wsl-pro-service
    237 syslog    20   0  222508   5504   4352 S   0.0   0.1   0:00.60 rsyslogd
    242 root      20   0    3160   1920   1792 S   0.0   0.0   0:00.16 agetty
    246 root      20   0    3116   1792   1664 S   0.0   0.0   0:00.04 agetty
    257 root      20   0  107028  22528  13184 S   0.0   0.6   0:00.79 unattended-upgr
   5379 polkitd   20   0  308164   7680   6912 S   0.0   0.2   0:00.14 polkitd
   6037 root      20   0  370084  20096  17280 S   0.0   0.5   0:00.11 packagekitd
   7384 ravish    20   0    9368   5632   3456 R   0.0   0.1   0:00.01 top
```

Step 5: Report Findings (10 mins)

1. Write a short report (5–10 lines) answering:
   - What debugging method helped you find issues?

```
 During this debugging exercise, careful inspection of Bash syntax and
variable usage helped identify issues, such as extra spaces inside quotes
that caused directory-not-found errors.
```

   - How did logging improve script visibility?

```
 Using logging with tee significantly improved script visibility, allowing
both standard output and error messages to be captured in run.log and
error.log, making it easier to trace execution flow and errors.
```

   - Which performance tool gave the most useful insights?

```
 For performance analysis, time and strace -c were the most useful tools
under WSL, since perf could not fully access hardware counters. These tools
revealed that most of the script’s runtime is spent on file I/O operations
rather than CPU-bound computations.
```

   - What would you optimize in the script?

```
 the script could be improved by avoiding repeated calls to cat and wc, and
instead using built-in Bash redirections or a single read loop, which would
reduce unnecessary process spawning and improve efficiency. Additionally,
trimming and formatting lines more efficiently could make the script faster
on larger datasets.
```