

# CSE 438/598: Embedded Systems Programming

## Device Drivers for Shared Message Queues

### Profiling Report

### Project 1

Author: Ankit Rathi

ASU ID: 1207543476

# 1 TABLE OF CONTENTS

---

2	Introduction .....	3
3	Profiling Report for Dynamically Allocated Message Buffers .....	4
3.1	CPU Cycles .....	4
3.1.1	CPU Cycles in User Space .....	4
3.1.2	CPU Cycles in Kernel Space .....	5
3.1.3	CPU Cycles in User and Kernel Space .....	6
3.2	Number of Instructions .....	7
3.2.1	Number of Instructions in User Space .....	7
3.2.2	Number of Instructions in Kernel Space .....	8
3.2.3	Number of Instructions in User and Kernel Space .....	9
3.3	Memory Usage.....	10
3.3.1	Memory Usage in User Space .....	10
3.3.2	Memory Usage in Kernel Space .....	11
4	Profiling Report for Statically Allocated Message Buffers .....	12
4.1	3.1 CPU Cycles.....	12
4.1.1	CPU Cycles in User Space .....	12
4.1.2	CPU Cycles in Kernel Space .....	13
4.1.3	CPU Cycles in User and Kernel Space .....	14
4.2	Number of Instructions .....	15
4.2.1	Number of Instructions in User Space .....	15
4.2.2	Number of Instructions in Kernel Space .....	16
4.2.3	Number of Instructions in User and Kernel Space .....	17
4.3	Memory Usage.....	18
4.3.1	Memory Usage in User Space .....	18
4.3.2	Memory Usage in Kernel Space .....	19
5	Bibliography .....	20

## 2 INTRODUCTION

---

The purpose of this project is to implement a device driver for shared message queues. In this project I have two programs. First, is the Device Driver, which performs the basic functionality of en-queue and de-queue on a Circular Buffer. It makes use of the helper file (header file) to perform the functionalities. Driver is also used to calculate the time spent by the message in the queues. Second, is a test program to control the threads in the project and generate random messages of random length for sending over the device. The test programs uses 3 sender threads which generate data and add the data into the device queue "*bus\_in\_q*". Bus Daemon thread is used to read messages from "*bus\_in\_q*" device and add it into the respective receiver queues. i.e "*bus\_out\_q1*", "*bus\_out\_q2*", and "*bus\_out\_q3*". The messages that have been queued into the receiver buffers are then picked up by the corresponding receiver threads. The driver implements the functionality of queuing time by making use of the TSC Counter. And in the last part of the project, the code has been analyzed by making use of Linux perf tool to check for number of CPU cycles, number of instructions executed in user and kernel space and memory used by the code during its execution.

Perf is a profiler tool for Linux 2.6+ based systems that abstracts away CPU hardware differences in Linux performance measurements and presents a simple commandline interface. Perf is based on the perf\_events interface exported by recent versions of the Linux kernel. This article demonstrates the perf tool through example runs. Output was obtained on a Ubuntu 11.04 system with kernel 2.6.38-8-generic results running on an HP 6710b with dual-core Intel Core2 T7100 CPU). For readability, some output is abbreviated using ellipsis ([...]).

## 3 PROFILING REPORT FOR DYNAMICALLY ALLOCATED MESSAGE BUFFERS

### 3.1 CPU CYCLES

The instruction cycle of CPU consists of:

- 1) fetch the next instruction to execute from the memory;
- 2) decode it;
- 3) run it.

This is known as the fetch-decode-execute cycle. My test CPU(Laptop) runs at 2.4GHz, meaning that it can run this cycle 2.4 billion times per second. Simpler CPUs, the instruction cycle is executed sequentially, i.e each instruction is processed completely before it begins to start with the next instruction. But, now a days CPUs, instruction cycle is executed simultaneously in parallel. The next instruction is fetched and being processed even before the previous instruction has finished executing. This has been made possible by techniques such as pipelining.

#### 3.1.1 CPU Cycles in User Space

Command: `sudo perf stat -e cycles:u ./main_1.o`

80,316,723 cycles are used by the user program.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
2047      1      2      89619780      224 mS      !"# $%&'()*+,-./
2052      1      1      87603192      219 mS      !"# $%&'()*+,-./01
23456789:;<=>?@
2051      2      1      94564404      236 mS      !"# $%&'()*+,-./01
23456789
2050      3      2      115151184      287 mS      !"# $%&'()*+,-./
0123456789:;<=>?@ABC
2046      2      2      126806016      317 mS      !"# $%&'()*+,-./
0123456789:;<=>?@ABCDEFGH IJ
2053      1      2      117000792      292 mS      !"# $%&'()*+,-./
0123456789:;<=>?@AB
Number of Messages Sent: 2053
Number of Messages Received By Receiver 1: 681
Number of Messages Received By Receiver 2: 682
Number of Messages Received By Receiver 3: 690
Total Number of Messages Received: 2053

Performance counter stats for './main_1.o':

      80,316,723 cycles:u

      9.422337929 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

### 3.1.2 CPU Cycles in Kernel Space

Command: `sudo perf stat -e cycles:k ./main_1.o`

274,338,448 cycles are used by the kernel program.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUV
1998      3      1      147596040      368 mS      !"#$%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ[\]^_
1997      2      1      135596724      338 mS      !"#$%&'()*+,-./0
123456789:;<=>
2000      3      2      154393764      385 mS      !"#$%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ
1999      1      1      145626372      364 mS      !"#$%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ[\]^_`abcdefgh
2001      3      3      143244312      358 mS      !"#$%&'()*+,-./
01
Number of Messages Sent: 2001
Number of Messages Received By Receiver 1: 662
Number of Messages Received By Receiver 2: 703
Number of Messages Received By Receiver 3: 636
Total Number of Messages Received: 2001

Performance counter stats for './main_1.o':

    274,338,488 cycles:k

    9.134116708 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

### 3.1.3 CPU Cycles in User and Kernel Space

Command: `sudo perf stat -e cycles:uk ./main_1.o`

386,907,943 cycles are used by the user and kernel program combined.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
0123456789:;<=>?@ABCDEFGHIJKLMNO P
2166      1      3      99042876      247 mS      !"#%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMNO PQRS
2167      2      3      97149000      242 mS      !"#%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMNO PQRS
2168      1      1      113862132      284 mS      !"#%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTU VWXYZ[\]^_`abcdefg
2169      3      2      122325012      305 mS      !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMNO PQRSTU VWXYZ[\]^_`abcdefghijklm
2170      3      2      88337760      220 mS      !"#%&'()*+,-./0
123456789:;<=>?@A
Number of Messages Sent: 2170
Number of Messages Received By Receiver 1: 745
Number of Messages Received By Receiver 2: 744
Number of Messages Received By Receiver 3: 681
Total Number of Messages Received: 2170

Performance counter stats for './main_1.o':

      386,907,943 cycles:uk

      9.978315582 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

## 3.2 NUMBER OF INSTRUCTIONS

Here, I have analyzed how much instructions are executed in total for the program in user and kernel space program.

### 3.2.1 Number of Instructions in User Space

Command: `sudo perf stat -e instructions:u ./main_1.o`

Nearly, 14 million instructions are executed by the user space program.

Number of Instructions / Clock Cycles = 14,198,316 / 80,316,723 = 0.17 instructions per cycle

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
2200      2      2      116425968      291 mS      !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMN0PQRSTUVWXYZ[\
2199      1      1      130510440      326 mS      !"#%&'()*+,-./0
123456789:;<=>?@ABC
2203      2      1      128046851      320 mS      !"#%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMN0PQRSTUVWXYZ[\
2205      1      2      116755944      291 mS      !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMN0PQRSTUVW
2202      3      1      119750568      299 mS      !"#%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLM
2204      2      3      117986052      294 mS      !"#%&'()*+,-.
Number of Messages Sent: 2205
Number of Messages Received By Receiver 1: 715
Number of Messages Received By Receiver 2: 723
Number of Messages Received By Receiver 3: 767
Total Number of Messages Received: 2205

Performance counter stats for './main_1.o':

      14,198,316 instructions:u

      10.041978003 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

### 3.2.2 Number of Instructions in Kernel Space

Command: `sudo perf stat -e instructions:k ./main_1.o`

Nearly, 94 million instructions are executed by the kernel space program.

Number of Instructions / Clock Cycles = 94,612,658 / 274,338,448 = 0.344 instructions per cycle

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
0123456789:;<=>?@ABCDEFGHIJKLMN0PQRSTUVWXYZ[
2055          3          1          126085752          315 mS    !"#%&'()*+,-./0
123456789:;<
2052          1          2          104704644          261 mS    !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMN0PQRSTUVWXYZ[\]^
2060          1          1          90115296          225 mS    !"#%&'()*+,-./01
23456789:
2059          2          1          73485168          183 mS    !"#%&'()*+,-./01
234567
2061          3          3          116958072          292 mS    !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMN0PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz
Number of Messages Sent: 2061
Number of Messages Received By Receiver 1: 676
Number of Messages Received By Receiver 2: 717
Number of Messages Received By Receiver 3: 668
Total Number of Messages Received: 2061

Performance counter stats for './main_1.o':

    94,612,658 instructions:k

    9.522906211 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```



### 3.2.3 Number of Instructions in User and Kernel Space

Command: `sudo perf stat -e instructions:uk ./main_1.o`

Nearly, 105 million instructions are executed by the user and kernel space program combined.

Number of Instructions / Clock Cycles = 105,472,281 / 386,907,943 = 0.272 instructions per cycle

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
2027      3      1      139016484      347 mS      !"#%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWX
2030      3      2      122765460      306 mS      !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ[\]^
2029      2      3      132654516      331 mS      !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ[\]^_`abcdefghijkl
2028      1      2      127409028      318 mS      !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKL
2032      3      3      150538980      376 mS      !"#%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ[\]^_`abcdefghijklmn
2031      2      1      128110068      320 mS      !"#%&'()*+,
Number of Messages Sent: 2032
Number of Messages Received By Receiver 1: 648
Number of Messages Received By Receiver 2: 699
Number of Messages Received By Receiver 3: 685
Total Number of Messages Received: 2032

Performance counter stats for './main_1.o':

    105,472,281 instructions:uk

    9.143497218 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

## 3.3 MEMORY USAGE

### 3.3.1 Memory Loads

Command: `sudo perf stat -e mem-loads ./main_1.o`

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
0123456789:;<=>?@ABCDEFGHI
2050      2      1      124717890      311 mS      !"#$%&'()*+,-./0
123456789:;<=>?@ABCDEFGHI
2049      3      2      127289352      318 mS      !"#$%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
2051      1      2      121300692      303 mS      !"#$%&'()*+,-./
0123456789:;<=>?@ABCDEF
2053      2      2      115471692      288 mS      !"#$%&'()*+,-./
0123456789:;<=>?@ABCDEFGHIJ
2052      3      3      130591656      326 mS      !"#$%&'()*+,-./
0123456789:;<=>?@ABC
Number of Messages Sent: 2053
Number of Messages Received By Receiver 1: 674
Number of Messages Received By Receiver 2: 683
Number of Messages Received By Receiver 3: 696
Total Number of Messages Received: 2053

Performance counter stats for './main_1.o':

          0 cpu/mem-loads/

    9.418361994 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

### 3.3.2 Memory Stores

Command: `sudo perf stat -e mem-stores ./main_1.o`

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
2062      2      2      84173916      210 mS      !"#$$%&'()*+,-./
2060      1      3      98053608      245 mS      !"#$$%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[
2064      2      1      77000832      192 mS      !"#$$%&'()*+,-./01
2
2065      3      3      60585201      151 mS      !"#$$%&'()*+,-./0
1234
2059      3      1      84704964      211 mS      !"#$$%&'()*+,-./01
23456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcd
2066      2      1      62600676      156 mS      !"#$$%&'()*+
2063      1      1      58751799      146 mS      !"#$$%&'()*+
Number of Messages Sent: 2066
Number of Messages Received By Receiver 1: 687
Number of Messages Received By Receiver 2: 677
Number of Messages Received By Receiver 3: 702
Total Number of Messages Received: 2066

Performance counter stats for './main_1.o':

      18,443,282 cpu/mem-stores/

      9.580241918 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

## 4 PROFILING REPORT FOR STATICALLY ALLOCATED MESSAGE BUFFERS

---

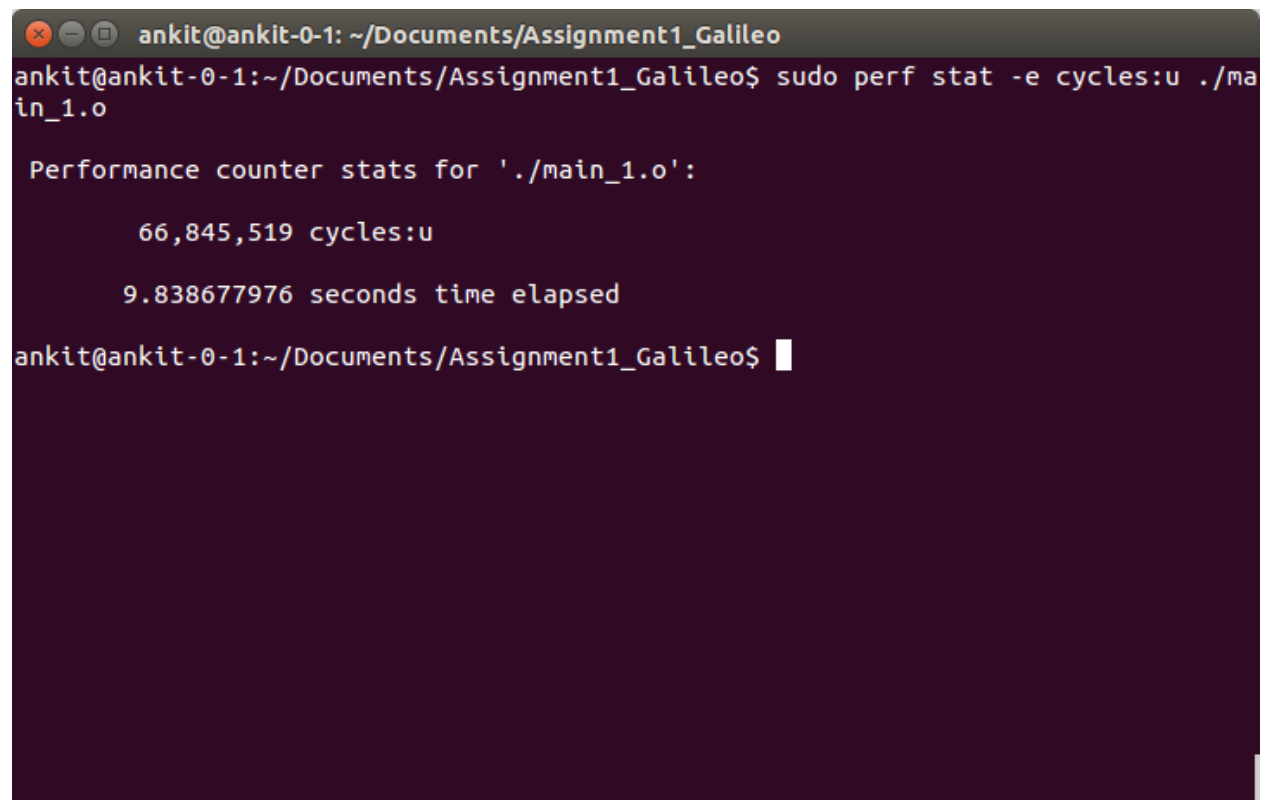
This is the second way in which the code in CircularBuffer.h has been designed. Here the message buffers have been allocated for all 10 buffers at one go.

### 4.1 3.1 CPU CYCLES

#### 4.1.1 CPU Cycles in User Space

Command: `sudo perf stat -e cycles:u ./main_1.o`

66,845,519 cycles are used by the user program.

A terminal window with a dark purple background and light green text. The window title is 'ankit@ankit-0-1: ~/Documents/Assignment1\_Galileo'. The command 'sudo perf stat -e cycles:u ./main\_1.o' has been executed. The output shows performance counter stats for './main\_1.o': 66,845,519 cycles:u and 9.838677976 seconds time elapsed. The prompt 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$' is visible at the bottom.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e cycles:u ./main_1.o

Performance counter stats for './main_1.o':

      66,845,519 cycles:u

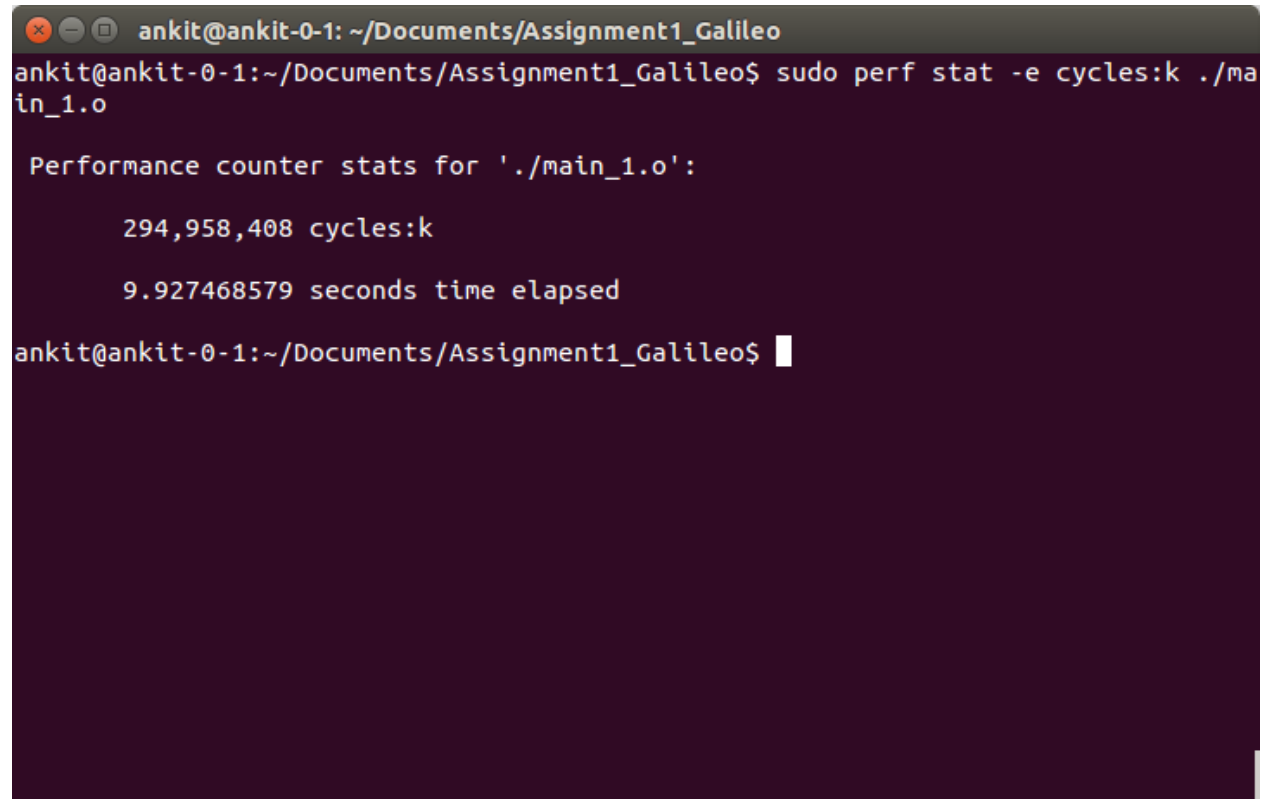
      9.838677976 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

#### 4.1.2 CPU Cycles in Kernel Space

Command: `sudo perf stat -e cycles:k ./main_1.o`

294,958,408 cycles are used by the kernel program.

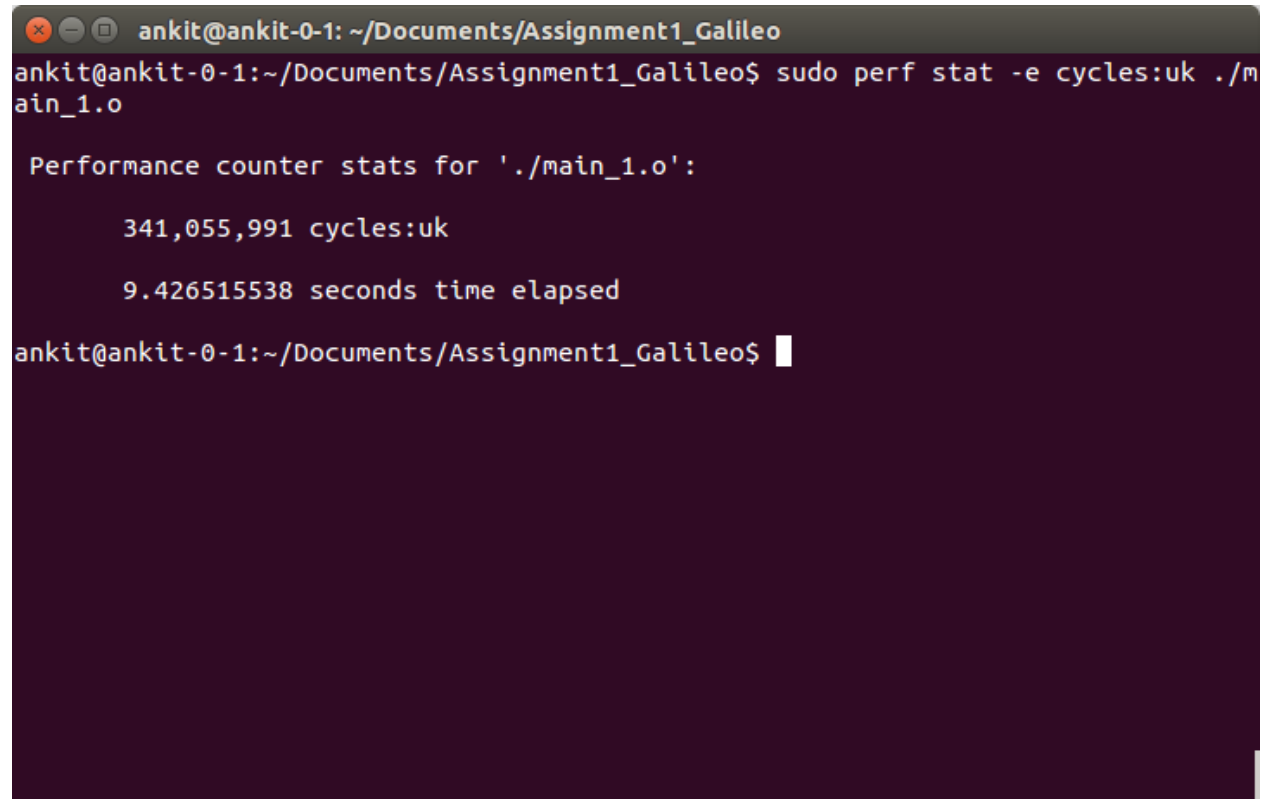
A terminal window with a dark purple background and light green text. The window title is 'ankit@ankit-0-1: ~/Documents/Assignment1\_Galileo'. The prompt is 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$'. The command entered is 'sudo perf stat -e cycles:k ./main\_1.o'. The output shows 'Performance counter stats for './main\_1.o':', followed by '294,958,408 cycles:k' and '9.927468579 seconds time elapsed'. The prompt returns to 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$' with a cursor.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e cycles:k ./main_1.o
Performance counter stats for './main_1.o':
    294,958,408 cycles:k
    9.927468579 seconds time elapsed
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

#### 4.1.3 CPU Cycles in User and Kernel Space

Command: `sudo perf stat -e cycles:uk ./main_1.o`

341,055,991 cycles are used by the user and kernel program combined.

A terminal window with a dark purple background and light green text. The window title is 'ankit@ankit-0-1: ~/Documents/Assignment1\_Galileo'. The prompt is 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$'. The command 'sudo perf stat -e cycles:uk ./main\_1.o' has been executed. The output shows performance counter stats for './main\_1.o': 341,055,991 cycles:uk and 9.426515538 seconds time elapsed. The prompt is now 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$' with a cursor.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e cycles:uk ./main_1.o

Performance counter stats for './main_1.o':

    341,055,991 cycles:uk

    9.426515538 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

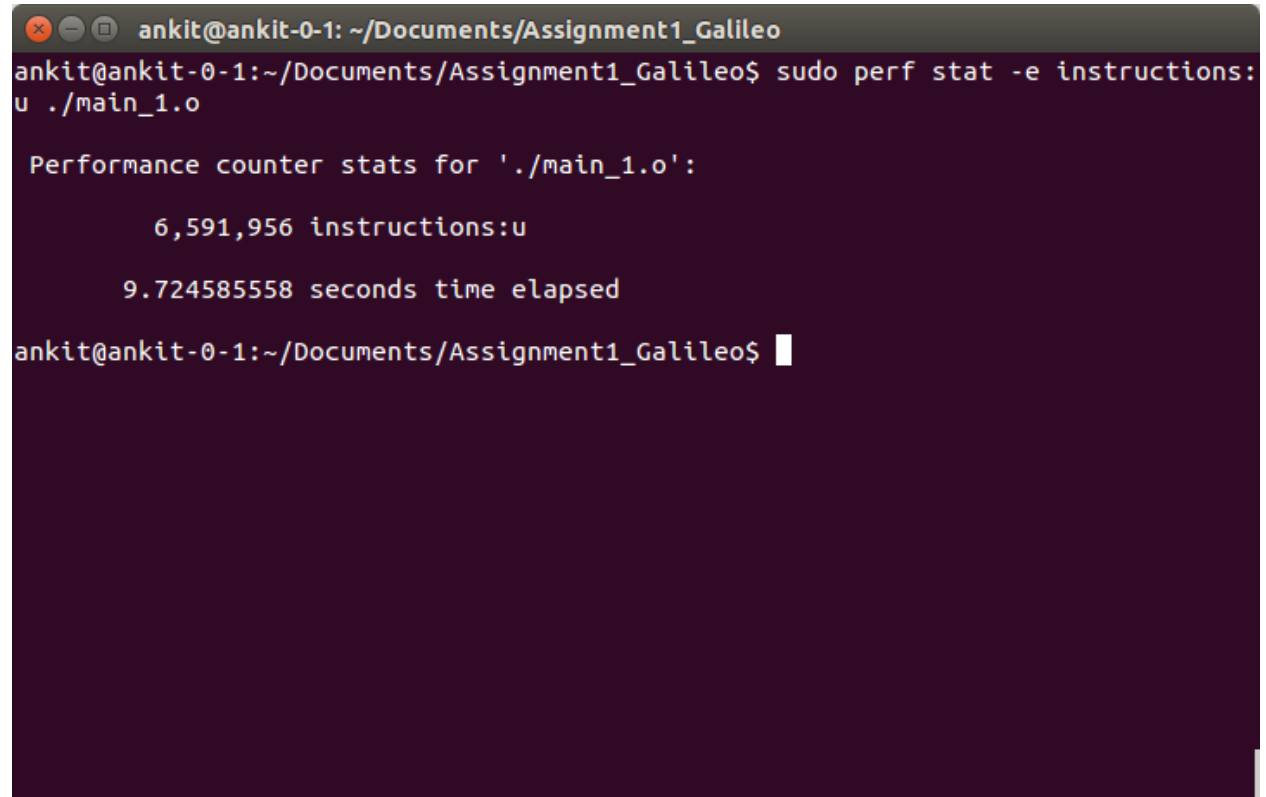
## 4.2 NUMBER OF INSTRUCTIONS

### 4.2.1 Number of Instructions in User Space

Command: `sudo perf stat -e instructions:u ./main_1.o`

Nearly, 6.5 million instructions are executed by the space program.

Number of Instructions / Clock Cycles =  $6,591,956 / 66,845,519 = 0.098$  instructions per cycle



```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e instructions:u ./main_1.o

Performance counter stats for './main_1.o':

      6,591,956 instructions:u

      9.724585558 seconds time elapsed

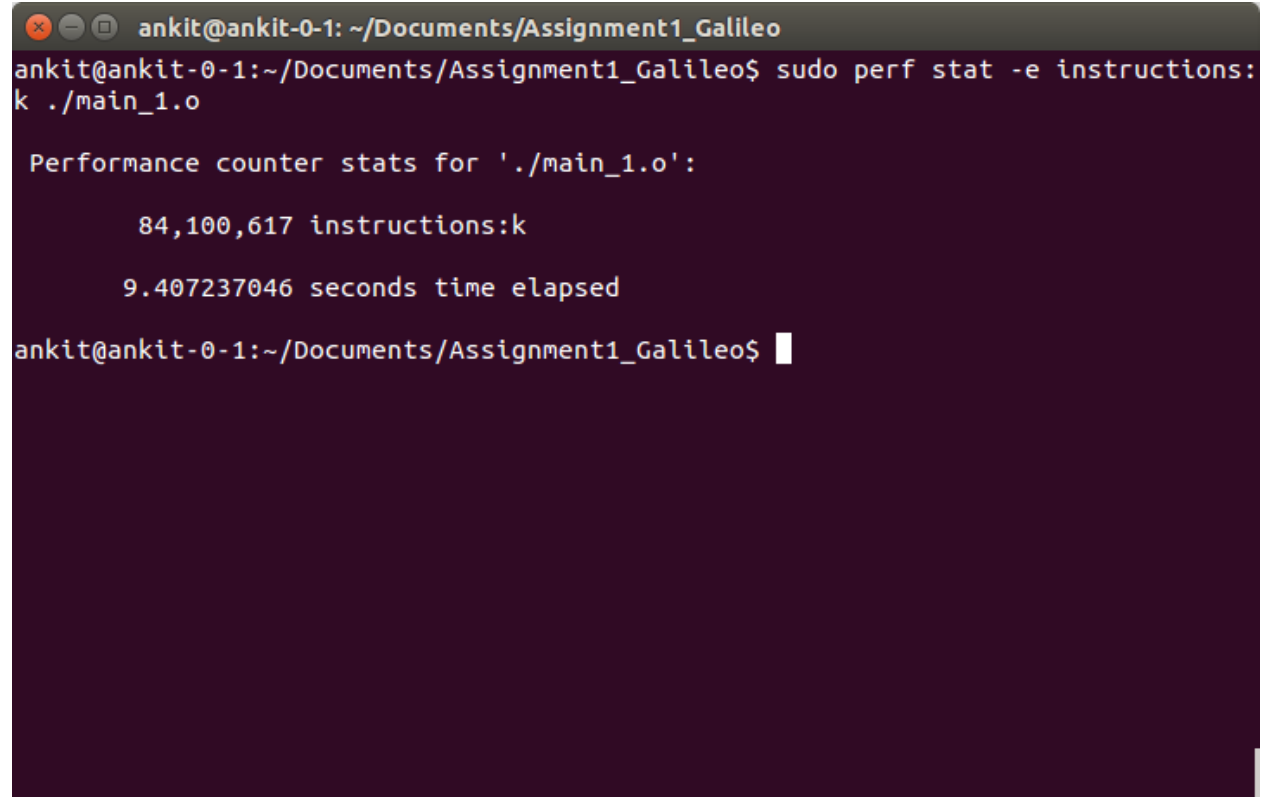
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

#### 4.2.2 Number of Instructions in Kernel Space

Command: `sudo perf stat -e instructions:k ./main_1.o`

Nearly, 84 million instructions are executed by the kernel space program.

Number of Instructions / Clock Cycles = 84,100,617 / 294,958,408 = 0.285 instructions per cycle

A terminal window with a dark purple background. The title bar shows 'ankit@ankit-0-1: ~/Documents/Assignment1\_Galileo'. The prompt is 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$'. The command entered is 'sudo perf stat -e instructions:k ./main\_1.o'. The output shows 'Performance counter stats for './main\_1.o':', followed by '84,100,617 instructions:k' and '9.407237046 seconds time elapsed'. The prompt returns to 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$' with a cursor.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e instructions:
k ./main_1.o

Performance counter stats for './main_1.o':

      84,100,617 instructions:k

      9.407237046 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

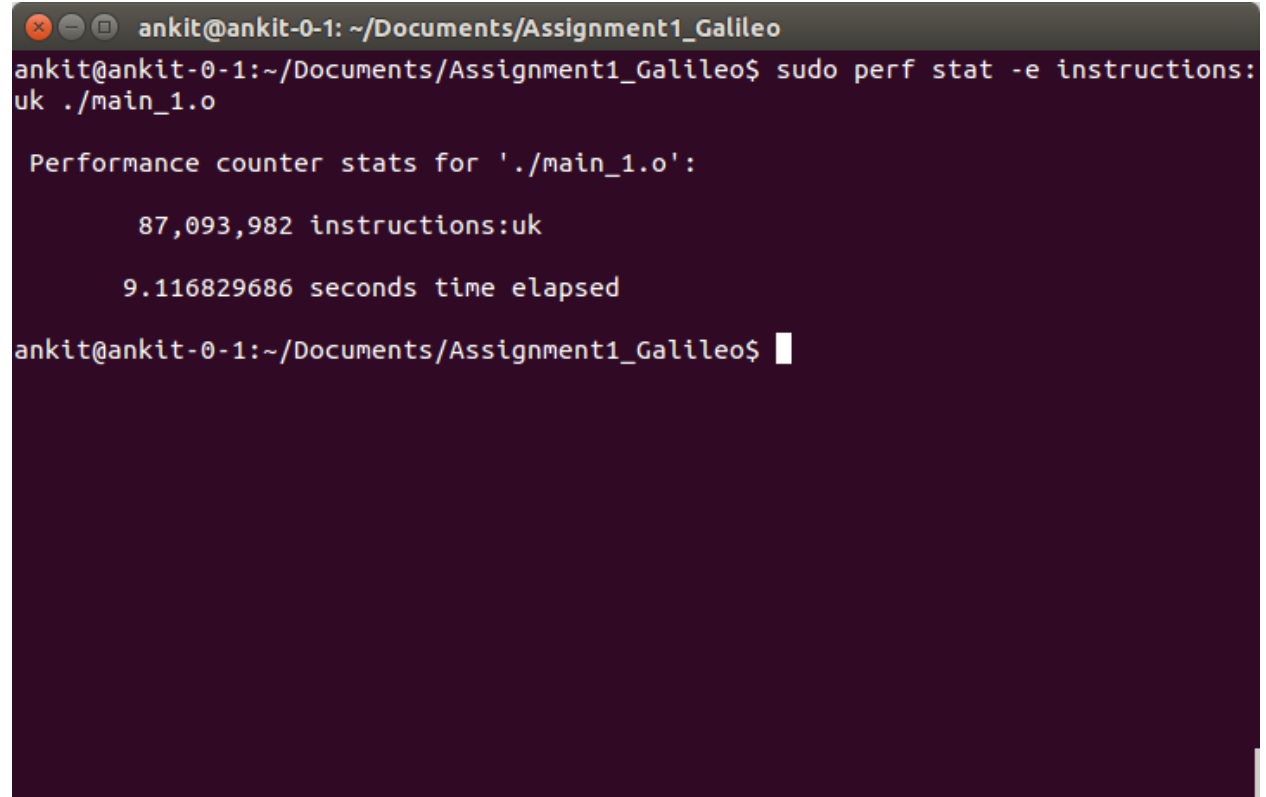


#### 4.2.3 Number of Instructions in User and Kernel Space

Command: `sudo perf stat -e instructions:uk ./main_1.o`

Nearly, 87 million instructions are executed by the user and kernel space program combined.

Number of Instructions / Clock Cycles = 87,093,982 / 341,055,991 = 0.255 instructions per cycle

A terminal window with a dark purple background and light green text. The window title is 'ankit@ankit-0-1: ~/Documents/Assignment1\_Galileo'. The prompt is 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$'. The command entered is 'sudo perf stat -e instructions:uk ./main\_1.o'. The output shows performance counter stats for './main\_1.o': 87,093,982 instructions:uk and 9.116829686 seconds time elapsed. The prompt returns to 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$' with a cursor.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e instructions:
uk ./main_1.o

Performance counter stats for './main_1.o':

      87,093,982 instructions:uk

      9.116829686 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

## 4.3 MEMORY USAGE

### 4.3.1 Memory Loads

Command: `sudo perf stat -e mem-loads ./main_1.o`

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ ls
CircularBuffer.h  Makefile      ReadMe.txt    Squeue.ko     Squeue.o
file.log          modules.order  run.sh        Squeue.mod.c
main_1.c          Module.symvers Squeue.c      Squeue.mod.o
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ cc -o main_1.o main_1.c -lpthread
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e mem-loads ./main_1.o

Performance counter stats for './main_1.o':

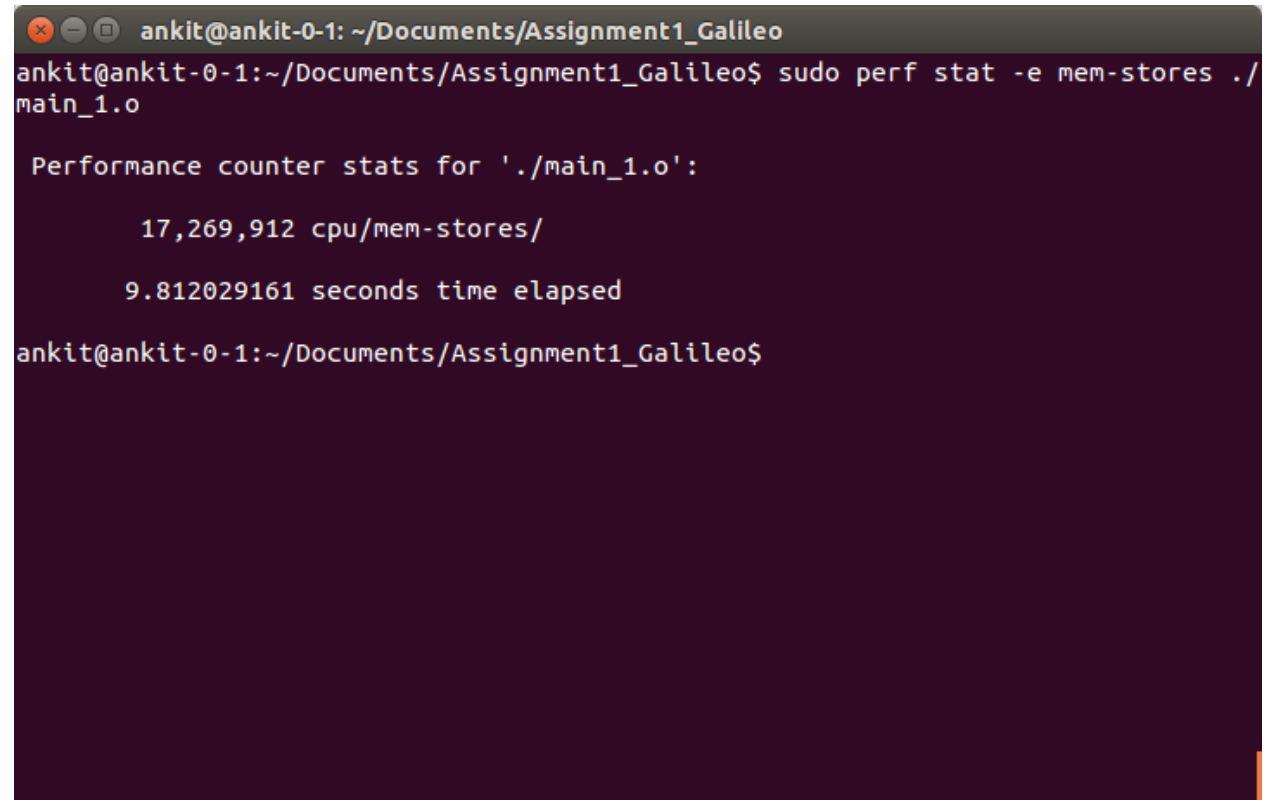
          0 cpu/mem-loads/

    9.952318928 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

### 4.3.2 Memory Stores

Command: `sudo perf stat -e mem-stores ./main_1.o`

A terminal window with a dark purple background and light green text. The window title is 'ankit@ankit-0-1: ~/Documents/Assignment1\_Galileo'. The prompt is 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$'. The command 'sudo perf stat -e mem-stores ./main\_1.o' has been entered. The output shows performance counter stats for './main\_1.o': '17,269,912 cpu/mem-stores/' and '9.812029161 seconds time elapsed'. The prompt is now 'ankit@ankit-0-1:~/Documents/Assignment1\_Galileo\$'.

```
ankit@ankit-0-1: ~/Documents/Assignment1_Galileo
ankit@ankit-0-1:~/Documents/Assignment1_Galileo$ sudo perf stat -e mem-stores ./
main_1.o

Performance counter stats for './main_1.o':

    17,269,912 cpu/mem-stores/

    9.812029161 seconds time elapsed

ankit@ankit-0-1:~/Documents/Assignment1_Galileo$
```

## 5 BIBLIOGRAPHY

---

1. <https://perf.wiki.kernel.org/index.php/Tutorial>
2. <http://paolobernardi.wordpress.com/2012/08/07/playing-around-with-perf/>
3. [http://en.wikipedia.org/wiki/Instruction\\_cycle](http://en.wikipedia.org/wiki/Instruction_cycle)