

Table of Contents

PART ZERO – Introduction on github	
PART ONE – Finding maximum temperature using script	-
PART TWO – Finding maximum temperature using java program.....	

Part 0:

Introduction is copied from discussion board and pasted into a file called README.md which is cloned from the github repository.

Once photo and information was added the file was pushed to github using the below commands

1.git add README.md

2.git commit -m "Adding introduction to readme"

3.git push

Below is the content which was updated in the README.md file.

##**INTRODUCTION**

I am Aneesh Partha from Tamilnadu,India. I have an software experience of 3 years and i am very happy to pursue my masters since this gives me a different experience and exposure.

In the past I did not get an opportunity to work as a developer but i am passionate about coding. I have a decent knowledge on working with Linux and Unix operating systems.

To tell about myself i love playing keyboard and i have passed examinations with distinction conducted by Trinity musical college , London.

![aneeshp487](<https://cloud.githubusercontent.com/assets/17997235/22183593/67346f9a-e087-11e6-854c-54cdeb01e364.jpg>)

Below picture is the output of the above lines:

INTRODUCTION

I am Aneesh Partha from Tamilnadu, India. I have an software experience of 3 years and i am very happy to pursue my masters since this gives me a different experience and exposure.

In the past I did not get an opportunity to work as a developer but i am passionate about coding. I have a decent knowledge on working with Linux and Unix operating systems.

To tell about myself i love playing keyboard and i have passed examinations with distinction conducted by Trinity musical college , London.



Part 1:

Result of maximum temperature script triggered for 1990.gz

```
vagrant@vagrant-ubuntu-trusty-64:/vagrant$ time ./max_temperature.sh
1990    607

real    0m14.891s
user    0m11.196s
sys     0m3.379s
vagrant@vagrant-ubuntu-trusty-64:/vagrant$
```

Result of maximum temperature script triggered for 1990.gz and 1992.gz

```
vagrant@vagrant-ubuntu-trusty-64:/vagrant$ time ./max_temperature.sh
1990    607
1992    605

real    1m58.945s
user    1m42.570s
sys     0m15.119s
vagrant@vagrant-ubuntu-trusty-64:/vagrant$
```

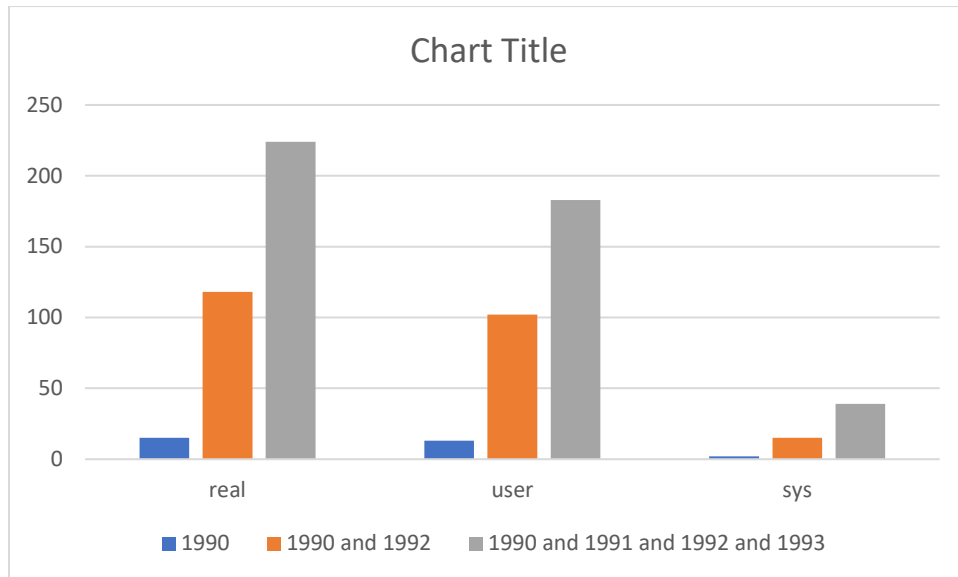
Result of maximum temperature script triggered for 1990.gz , 1991.gz , 1992.gz and 1993.gz

```
vagrant@vagrant-ubuntu-trusty-64:/vagrant$ time ./max_temperature.sh
1990    607
1991    607
1992    605
1993    567

real    3m44.527s
user    3m3.423s
sys     0m39.001s
vagrant@vagrant-ubuntu-trusty-64:/vagrant$
```

Analysis between output of 3 different time

Dataset	real	user	sys
1990	15	13	2
1990 and 1992	118	102	15
1990 , 1991 , 1992 and 1993	224	183	39



The “Time” keyword included with a command gives us the time taken for the action to take place. In this case we have triggered the script `max_temperature.sh` which takes a file as input and gives out maximum value of temperature available in the file.

The above graph shows the time consumed for different data sets. Real time is the actual time the script has taken to read all the lines in the file and then identify the maximum temperature. As we can see the time taken for reading 4 different datasets is more than for reading a single dataset or two. This clearly shows that time taken is directly proportional to the size of the data. As data size increases the time taken for reading and manipulating the data also increases.

Real time is always greater than the user and sys time. User time is the amount of CPU time spent on user mode. All the time that is real user and sys is greater for the large datasets and less for small data sets.

RAM – 4GB

CPU - 4

Part 2:

Data set 1 :

Name: Aneesh Partha

CWID- A20376172

```
Inserting record 175
Inserting record 210
Inserting record 69
Inserting record 228
Inserting record 175
Inserting record 188
Inserting record 235
Inserting record -130
Inserting record 65
Inserting record -105
Inserting record 210
Inserting record 44
607
real    58m12.523s
user    2m51.701s
sys     3m3.882s
vagrant@vagrant-ubuntu-trusty-64:/vagrant/compfold$
```

Data set 2 :

```
Maximum temperature for year 1990 is 607
Maximum temperature is 605
real    325m46.189s
user    14m54.133s
sys     18m46.217s
vagrant@vagrant-ubuntu-trusty-64:/vagrant/compfold$
```

Data set 3:

```
vagrant@vagrant-ubuntu-trusty-64:~$ java -Xmx4096M -jar ./DataSet3.jar
Picked up _JAVA_OPTIONS: -Xmx4096M
vagrant@vagrant-ubuntu-trusty-64:~$ java -Xmx4096M -jar ./DataSet3.jar
Picked up _JAVA_OPTIONS: -Xmx4096M
real    514m22.323s
user    21m18.143s
sys     24m33.637s
```

Time taken by data set 1:

```
Data set 1 - 1990
-----
Maximum temperature is : 607
Time taken is 2731 milliseconds
vagrant@vagrant-ubuntu-trusty-64:~$
```

Time taken by data set 2:

Data set 2 - 1990,1992

Success

Maximum temperature of year 1990 is : 607

Maximum temperature of year 1992 is : 605

Total time taken is 3658milliseconds



Aneesh Partha

A20376172

Time taken by data set 3:

Success

Maximum temperature of year 1990 is : 607

Maximum temperature of year 1991 is : 607

Maximum temperature of year 1992 is : 605

Maximum temperature of year 1993 is : 567

5124

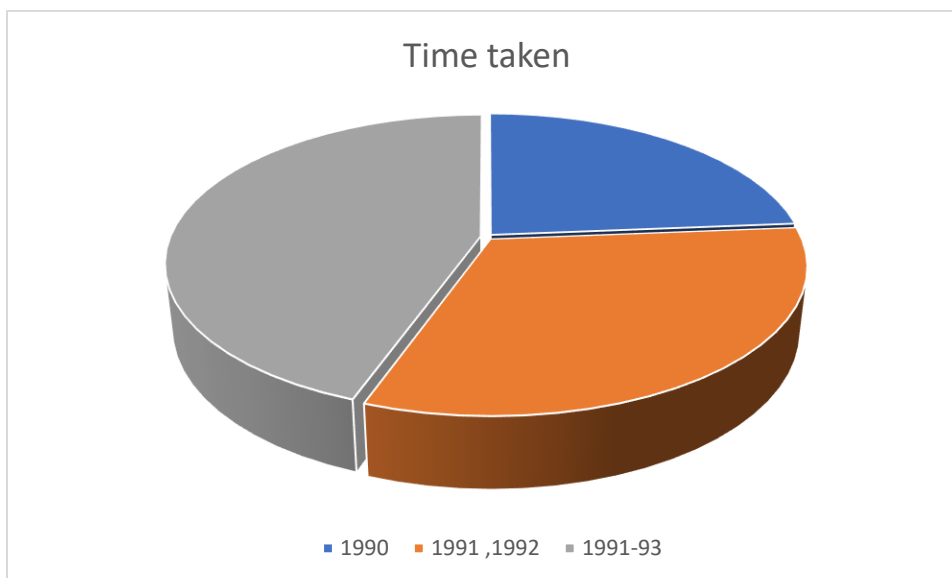
vagrant@vagrant-ubuntu-trusty-64:/vagrant\$

Aneesh Partha

A20376172

Table showing the time taken for querying the maximum temperature in milliseconds.

Year	Time taken
1990	2731
1991,1992	3658
1991-93	5124



From the graph we can conclude that when the dataset is large the entries in database will be more which will result in more time for querying the database. As we can see in the above diagram data set 3 has consumed more time when compared to data set 2 and data set 1. Data set 1 is the least since it involves data of only the year 1990.

RAM – 4GB

CPU - 4