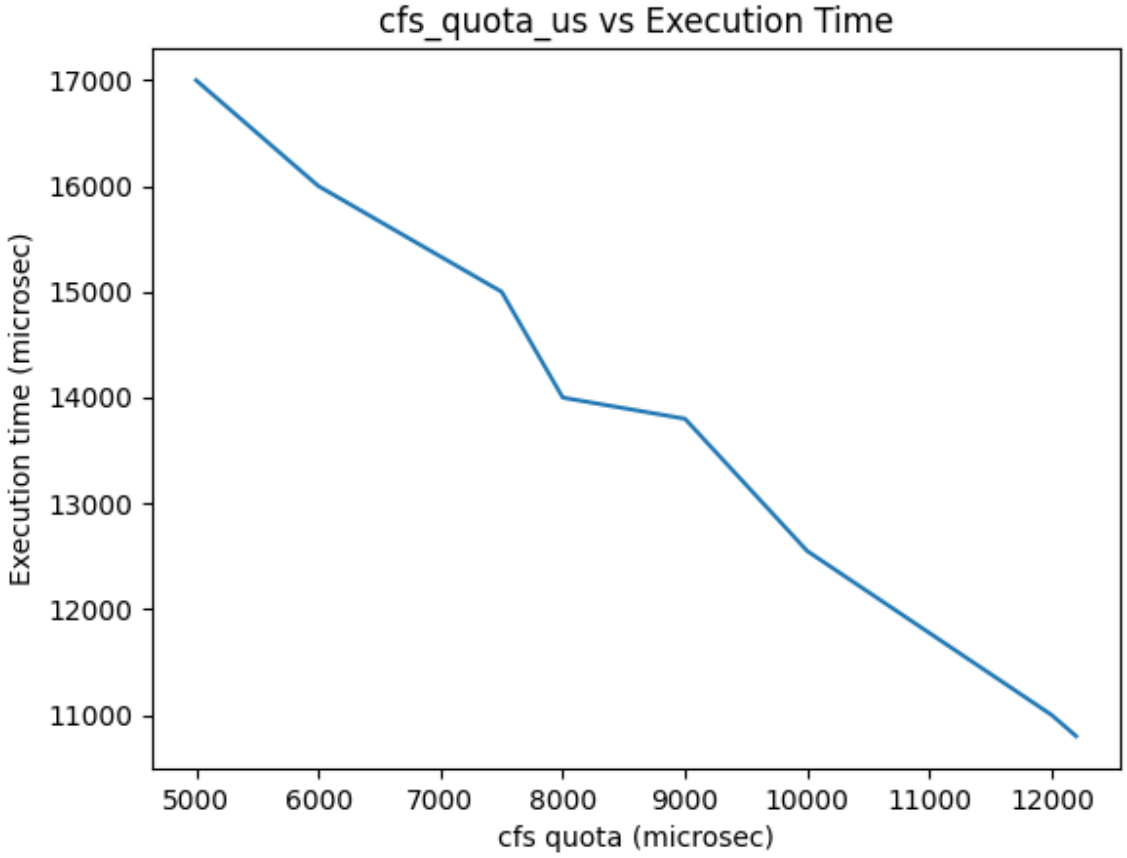
Assignment 4
Just group it!

Q1: **seeing cgroups**

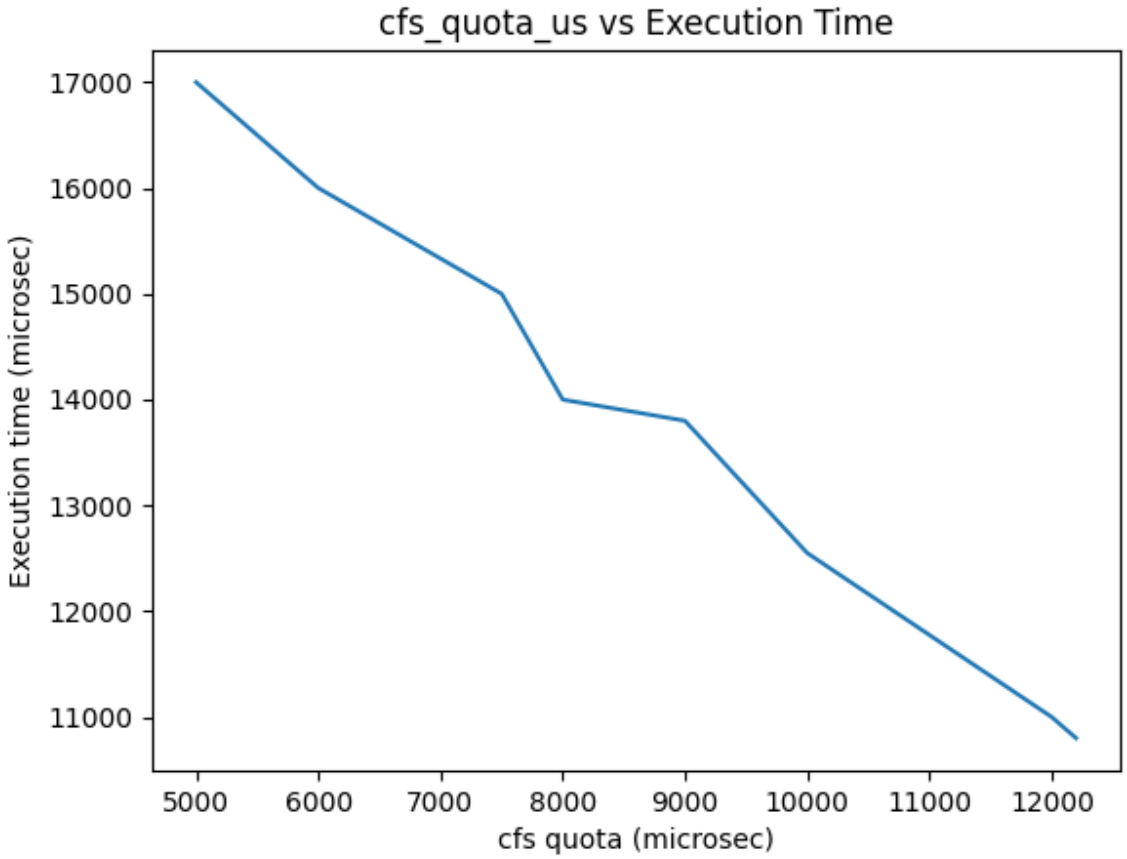
a)visual #1



Period	Shares	Quota	Execution Time
100	100	70	36.564s
100	100	50	125.172s
100	100	30	80.746s

b)visual #2

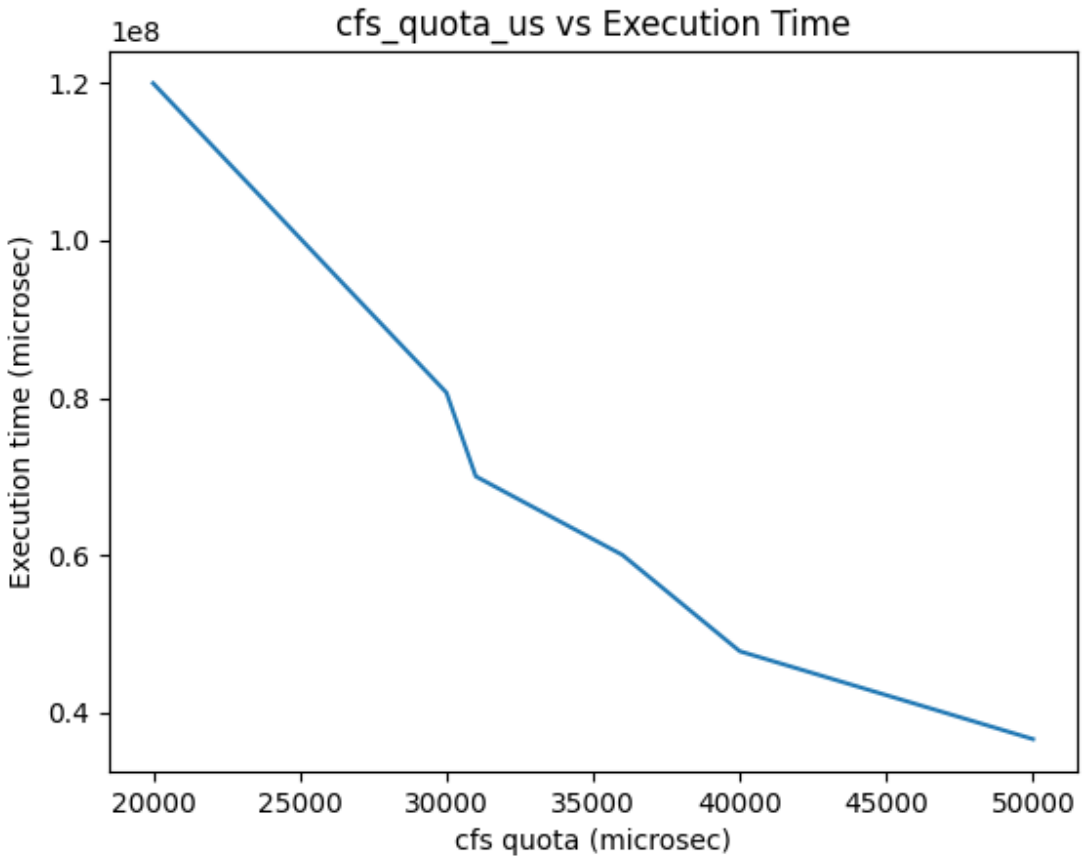
For program 1:



Period	Shares	Quota	Execution Time
100	100	70	36.564s
100	100	50	125.172s
100	100	30	80.746s

For program 2:

Period	Shares	Quota	Execution Time
100	100	20	125.7200 s
100	100	30	75.3471
100	100	50	32.8564000 s



Q2: namespaces or spaces with names?

The programs given in the link are demo_uts_namespaces.c, ns_exec.c, and unshare.c

O/P of **demo_uts_namespaces.c**

```
PID of child created by clone() is 41325
uts.nodename in child: temp
uts.nodename in parent: Nikhil
child has terminated
```

O/P of **ns_exec.c** : sudo ./setns /proc/41759/ns/uts hostname

```
temp
```

O/P of **unshare.c** : sudo ./unshare -p sudo /bin/bash

```
$ echo $$
1
```

b) Repeating the above process using the command line tools unshare and nsenter

Using **unshare**

```
$ sudo su
$ unshare -p --fork --mount-proc
$ ps
    PID TTY TIME CMD
    1 pts/2 00:00:00 bash
    11 pts/2 00:00:00 ps
```

$ echo $$
1

Using **setns**:

```
nikhil@NikhilSystem:~$ sudo su
root@NikhilSystem:/home/test# unshare -p --fork --mount-proc
root@NikhilSystem:/home/test# ps
    PID TTY TIME CMD
    1 pts/2 00:00:00 bash
    11 pts/2 00:00:00 ps
```

Another terminal output

```
nikhil@NikhilSystem:~$ sudo su
root@NikhilSystem:/home/temp# nsenter -t 44251 -p -m
root@NikhilSystem:/# ps -a
    PID TTY TIME CMD
    1 pts/2 00:00:00 bash
    12 pts/0 00:00:00 bash
    21 pts/0 00:00:00 ps
```

c) attaching the program to the namespace

Commands used:

```
sudo unshare –pid –fork bash
```

This will open :

```
root@NikhilSystem:/home/nikhil/test/Cloud_Computing/22M0814-cs695-a#
ps
```

```
  PID TTY          TIME CMD
61125 pts/10   00:00:00 sudo
61126 pts/10   00:00:00 unshare
61127 pts/10   00:00:00 bash
61134 pts/10   00:00:00 ps
```

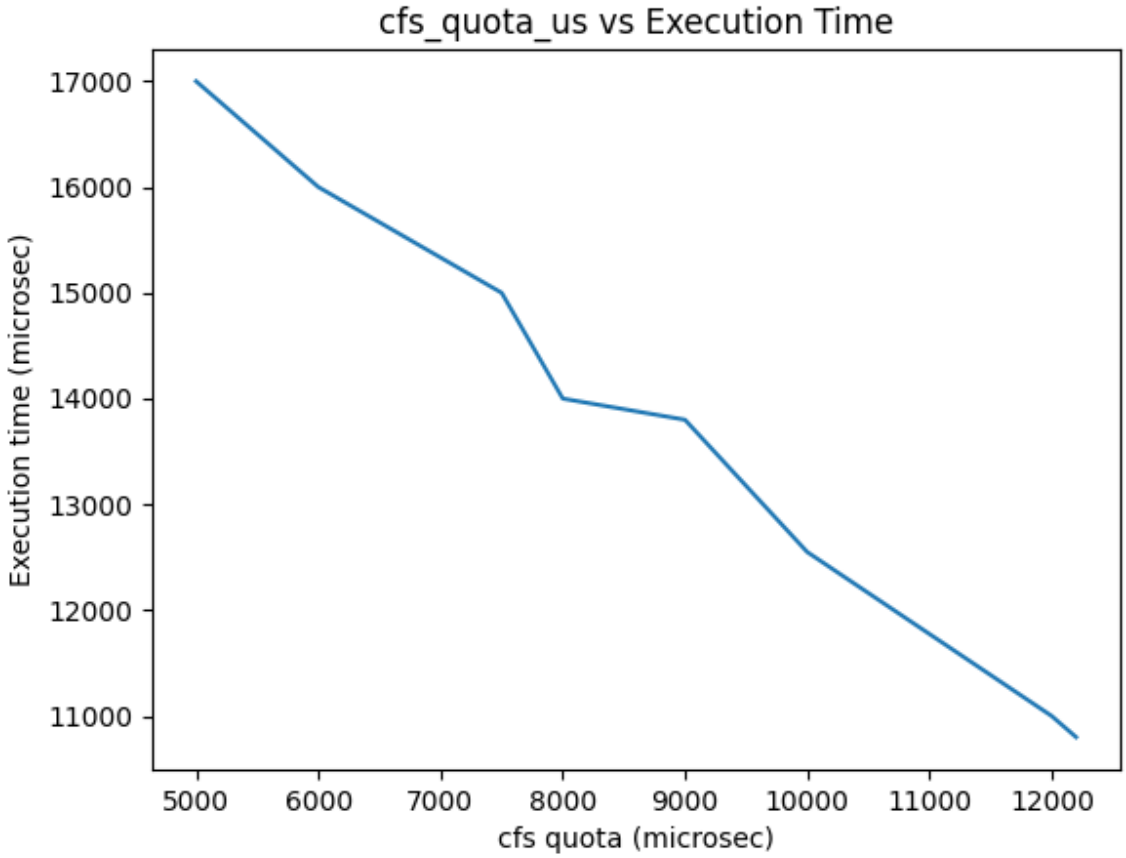
Now using the same pid

```
  PID TTY          TIME CMD
61156 pts/1    00:00:00 sudo
61157 pts/1    00:00:00 setns_pidfd
61158 pts/1    00:00:00 bash
61192 pts/1    00:00:00 ps
```

Q3 : 1+2=3

- Writing the program that creates 5 child processes
    a. By using the clone system call and passing the container process' function as an argument, we can create container processes.
    b. Then, we modify the program's root directory from main to rootdir.

c. using the the CLONE NEWPID flag of the PID namespace of the container with the setns command,which enables the child processes entry into the container process's PID namespace.
d. After that, we fork 5 processes that are located in directories that are related to the rootdir and then we use exec to run program




```
nikhil@NikhilSystem:~/test/Cloud_Computing$ ./1+2=3
Start
process id of parent - 728549
process id of container - 728550
pid created for container - 1
child pid is 2
child pid is 3
child pid is 4
child pid is 5
child pid is 6
total time 10589654
```