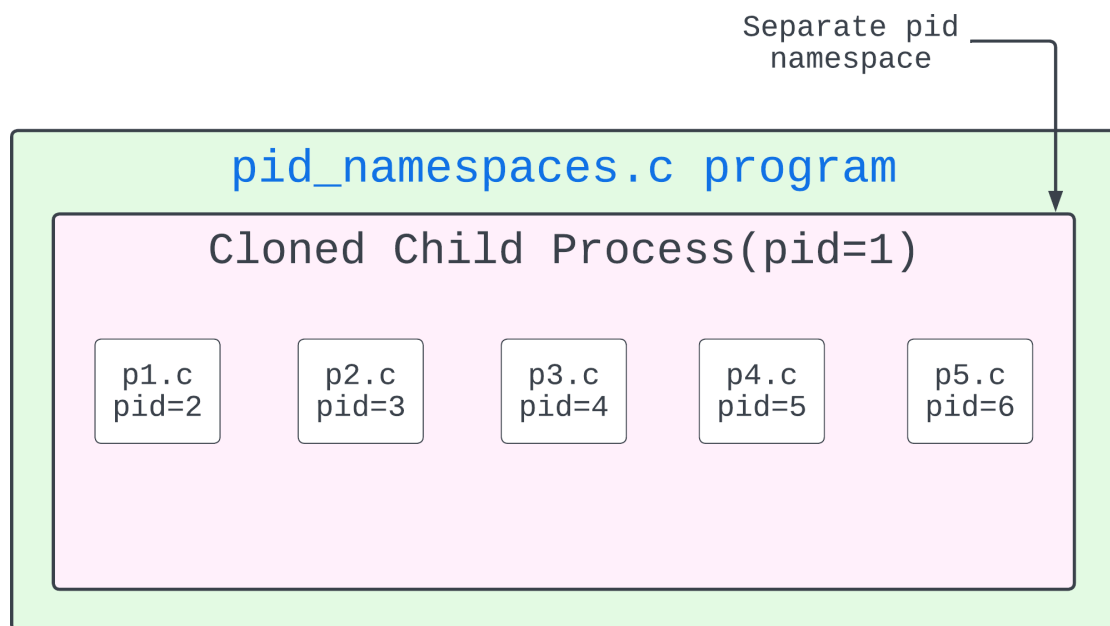


### Q3

#### Part 1



I have created a file called `pid_namespaces.c` which creates a new child process with different pid namespaces using `clone`. The child process is created with `CLONE_NEWPID` flag. The parent process waits for the child process to finish and then exits.

The child process change its root directory to `./root` using `chroot` and already added all required libraries in the root directory and program is stored in `root/program` directory.

Setup can be automatically done by running `setup.sh` script and then `compile_files.sh` to compile the program.

Inside the child process after changing the root directory, 5 processes are forked and run the all 5 programs using `exec` system call.

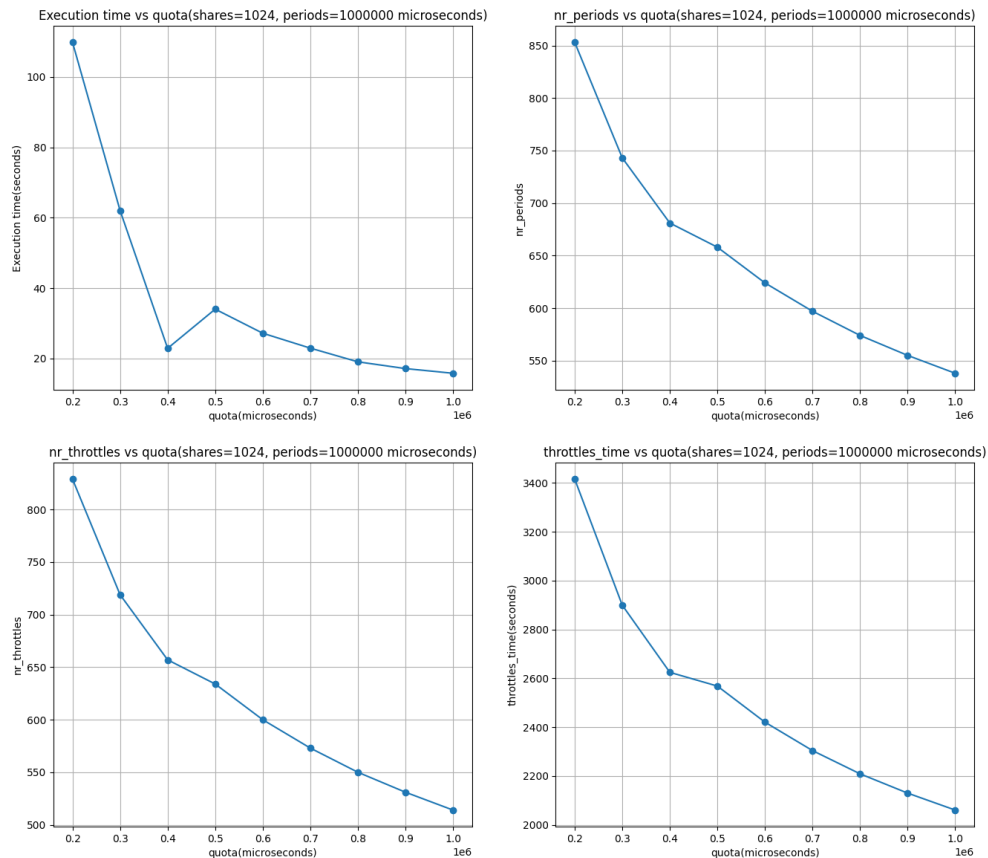
```
root@synerg-optane-2:/home/synerg/raja/A4/Q3# echo "Q3 part-1"
Q3 part-1
root@synerg-optane-2:/home/synerg/raja/A4/Q3# bash setup.sh >/dev/null
root@synerg-optane-2:/home/synerg/raja/A4/Q3# bash compile_files.sh
root@synerg-optane-2:/home/synerg/raja/A4/Q3# ./pid_namespaces.o
PID of Container created by clone(): 1447836      Main Program PID: 1447835
PID inside new PID namespace: 1      Parent PID: 0

Working Directory: /home/synerg/raja/A4/Q3
Root Directory: /home/synerg/raja/A4/Q3/root
Working Directory after chroot: /

Running programs inside new PID namespace
Program Name: ./program/p1.o, PID: 2      Parent PID 1
Program Name: ./program/p2.o, PID: 3      Parent PID 1
Program Name: ./program/p3.o, PID: 4      Parent PID 1
Program Name: ./program/p4.o, PID: 5      Parent PID 1
Program Name: ./program/p5.o, PID: 6      Parent PID 1
This is program 5 with PID 6
This is program 3 with PID 4
This is program 2 with PID 3
This is program 4 with PID 5
This is program 1 with PID 2
Terminating Cloned Process Or you can say mini container
root@synerg-optane-2:/home/synerg/raja/A4/Q3#
```

## Part 2

```
cd Q3
sudo su
[sudo] password for synerg:
root@synerg-optane-2:/home/synerg/raja/A4/Q3# bash setup.sh >/dev/null
root@synerg-optane-2:/home/synerg/raja/A4/Q3# bash compile_files.sh
root@synerg-optane-2:/home/synerg/raja/A4/Q3# bash result.sh
```



## Observations

### Part 2:

- `result.sh` and `graph.py` are used to generate the graph.
- `pid_namespaces_2.c` is used to create the child process with different pid namespaces. It also add the process to the cgroup.
- As we can see in the graph, the execution time of the program is increasing as the number of processes increases. This is because the execution time of the program is proportional to the number of processes.
- As we can see in the graph, increasing quota of the cgroup is decreasing the execution time of the program and similarly for the other metrics as well like throttled time, `nr_throttled`, `nr_periods`.
- This is because we are getting more CPU time for the program to execute and hence the execution time is decreasing.
- Same behaviour is observed in the Q1 part 1 as well.