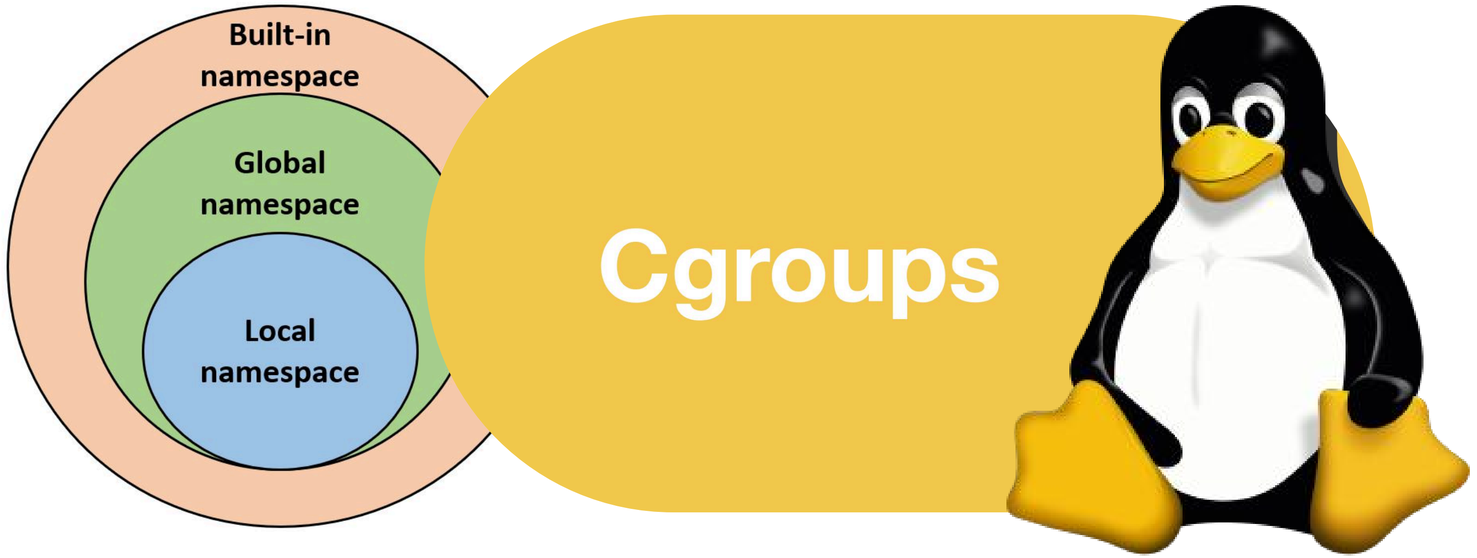
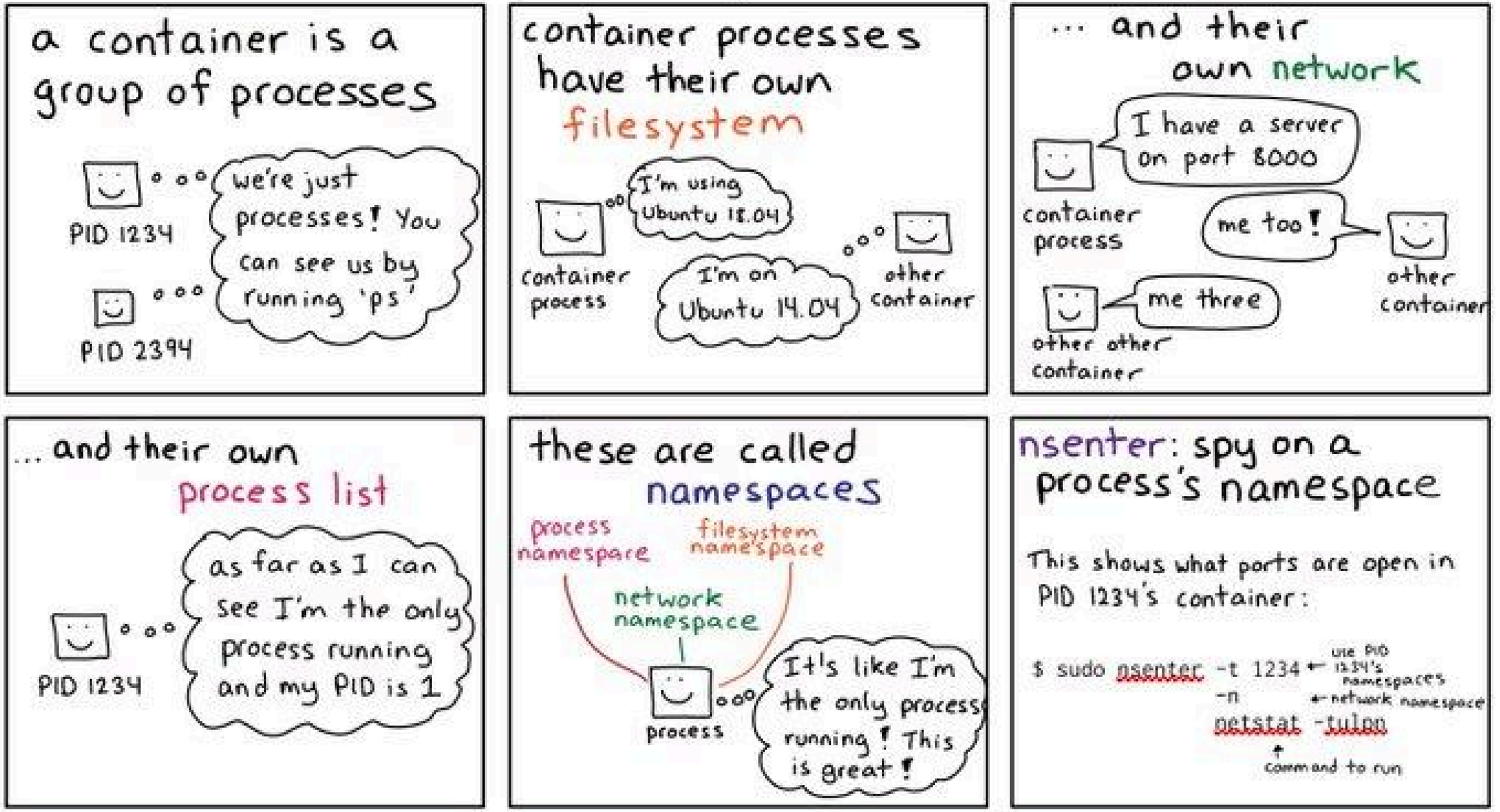
cgroups и namespaces в Linux

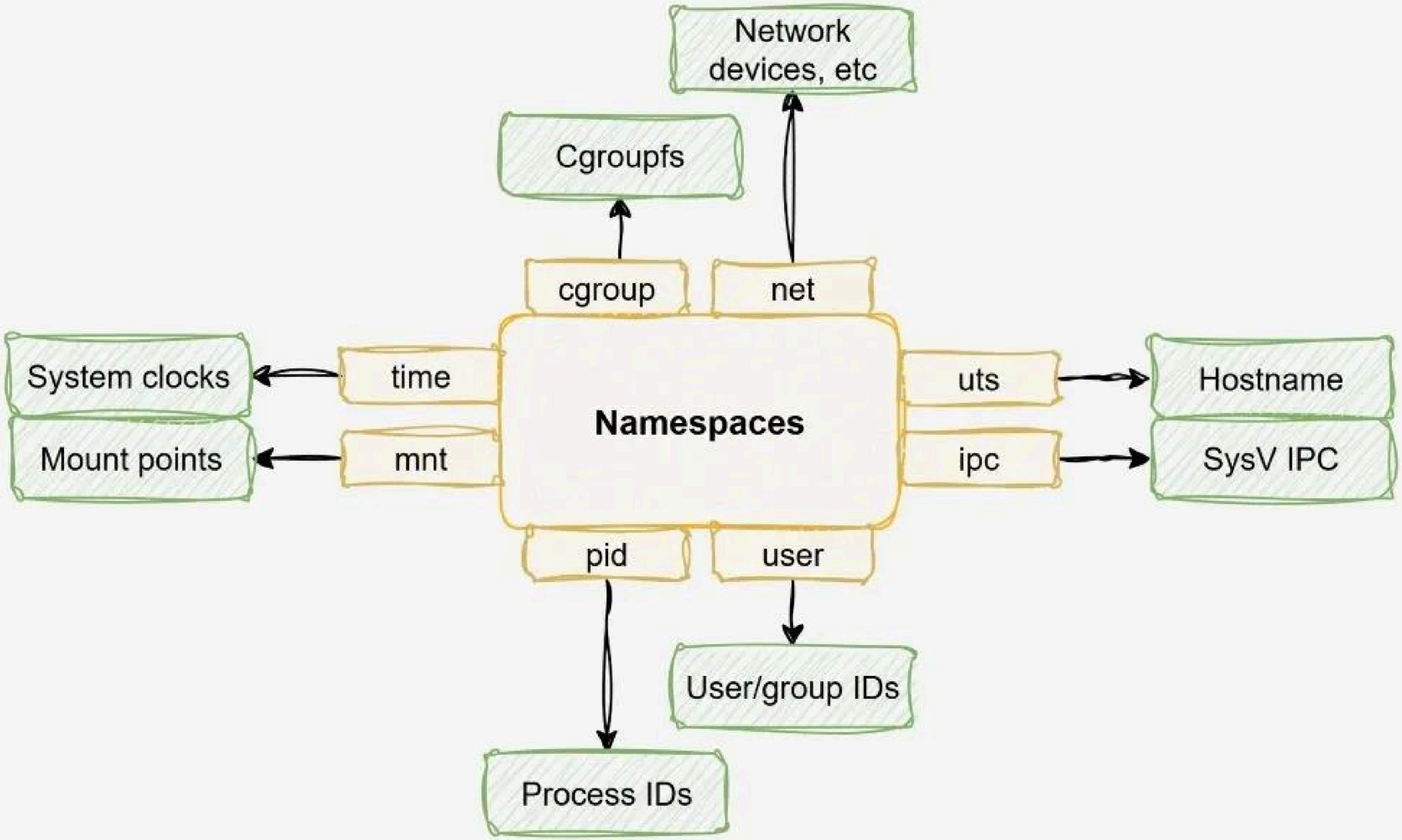


Жахонгир Ахмадалиев

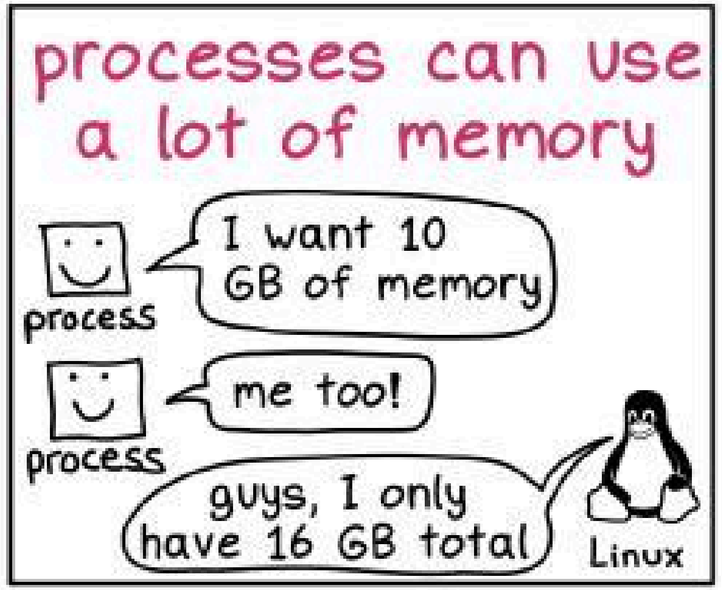
Julia Evans
@b0rk

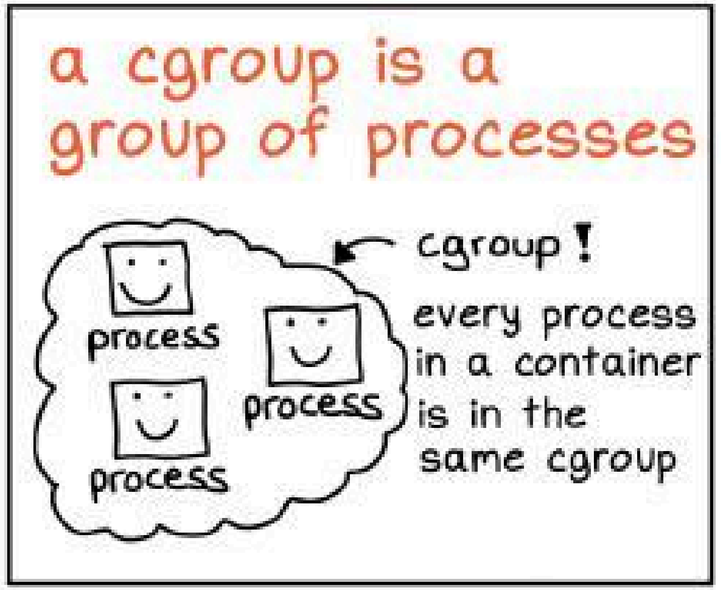
namespaces

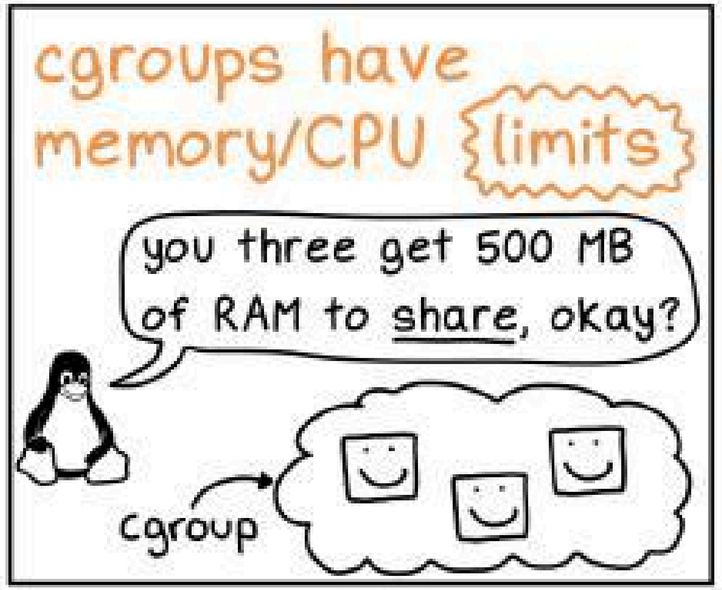


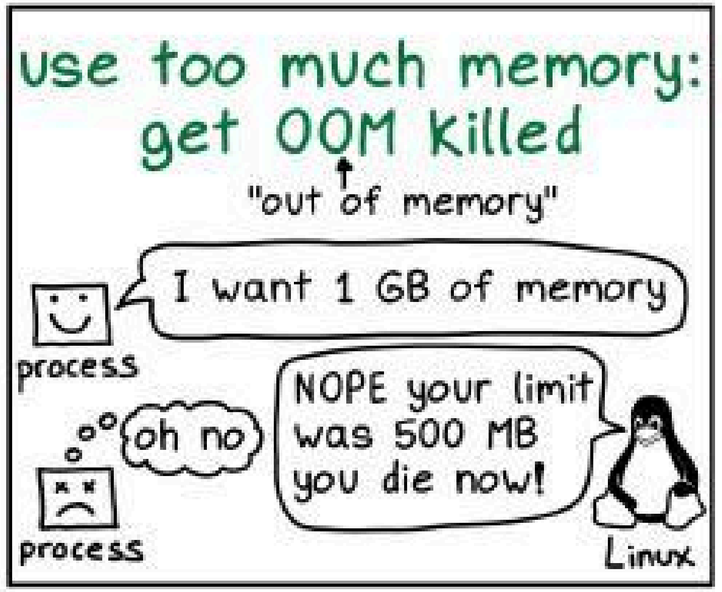

Network devices, etc
Cgroupfs
cgroup
net
System clocks
time
Namespaces
uts
Hostname
Mount points
mnt
ipc
SysV IPC
pid
user
User/group IDs
Process IDs

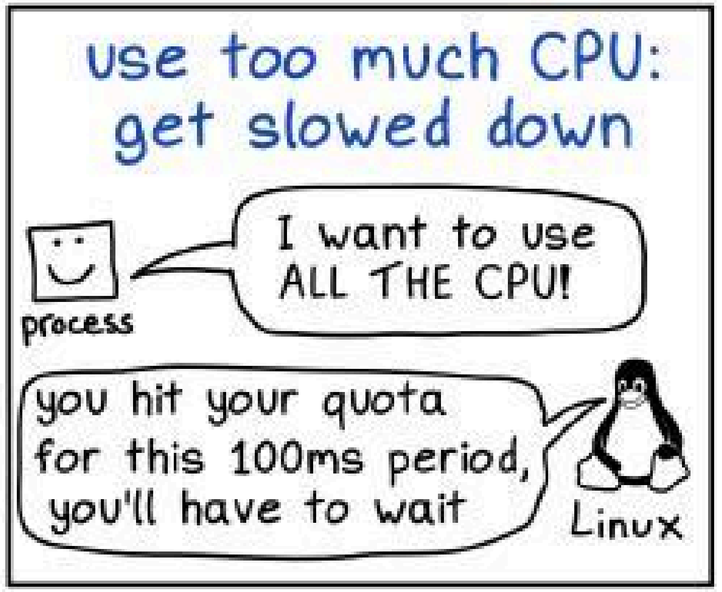
cgroups

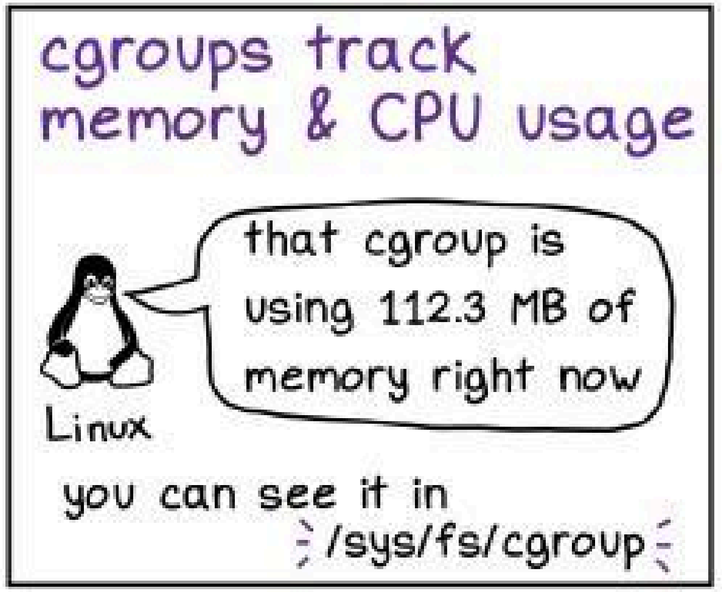

processes can use a lot of memory
process
I want 10 GB of memory
process
me too!
guys, I only have 16 GB total
Linux

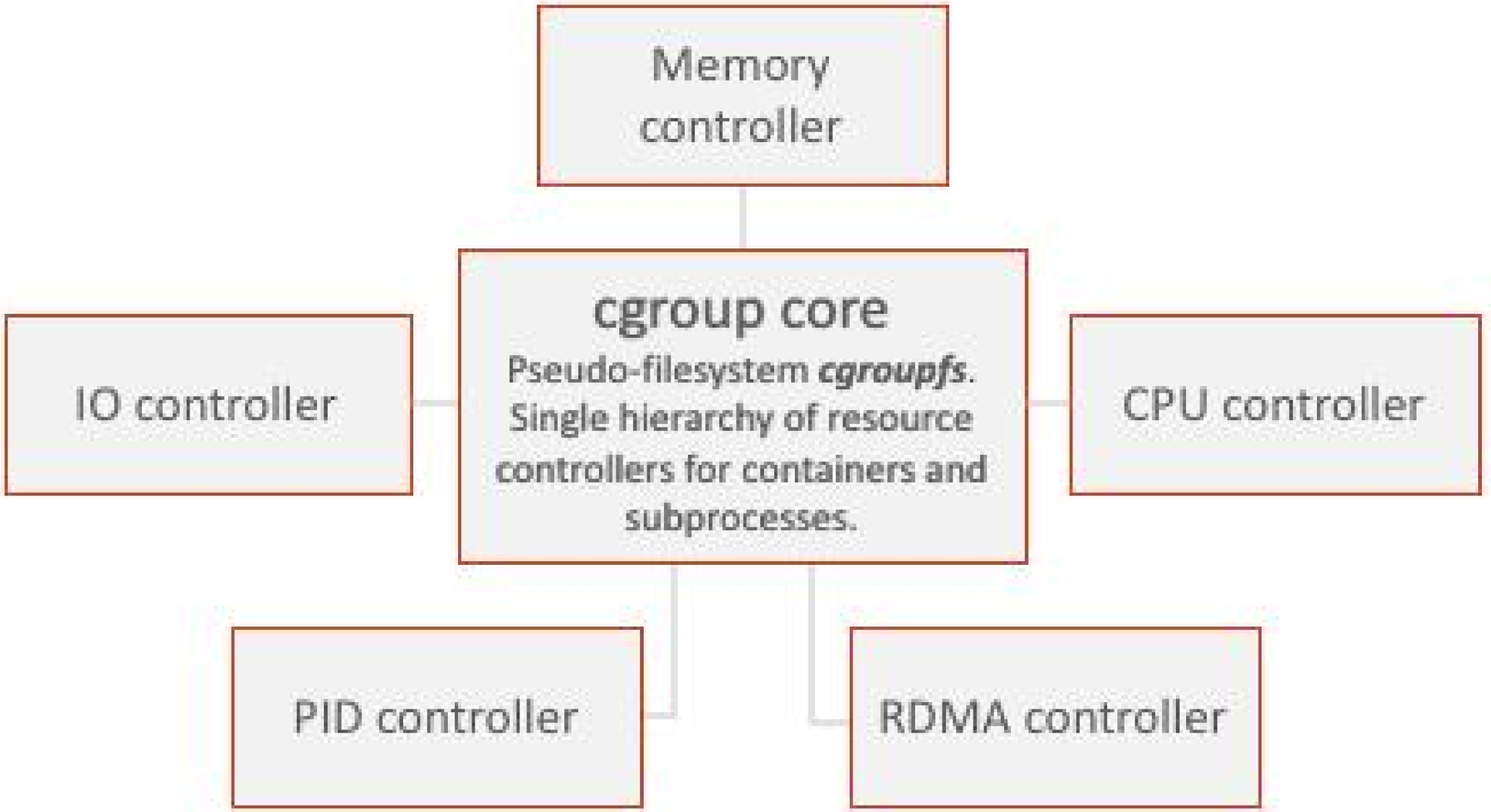

a cgroup is a group of processes
process
process
process
process
cgroup!
every process in a container is in the same cgroup


cgroups have memory/CPU limits
you three get 500 MB of RAM to share, okay?
cgroup


use too much memory: get OOM killed
"out of memory"
process
I want 1 GB of memory
process
oh no
NOPE your limit was 500 MB you die now!
Linux


use too much CPU: get slowed down
process
I want to use ALL THE CPU!
you hit your quota for this 100ms period, you'll have to wait
Linux


cgroups track memory & CPU usage
Linux
that cgroup is using 112.3 MB of memory right now
you can see it in
/sys/fs/cgroup


Memory controller
IO controller
cgroup core
Pseudo-filesystem cgroupfs.
Single hierarchy of resource controllers for containers and subprocesses.
CPU controller
PID controller
RDMA controller

UNSHARE(1) User Commands UNSHARE(1)

NAME top

unshare - run program in new namespaces

SYNOPSIS top

unshare [options] [*program* [*arguments*]]

DESCRIPTION top

The **unshare** command creates new namespaces (as specified by the command-line options described below) and then executes the specified *program*. If *program* is not given, then "${SHELL}" is run (default: */bin/sh*).

- `--fork` : форкает процесс после создания пространств имён.
- `--pid` : создаёт новый PID namespace.
- `--mount` : создаёт новый Mount namespace.
- `--uts` : создаёт новый UTS namespace.
- `--ipc` : создаёт новый IPC namespace.
- `--net` : создаёт новый Network namespace.
- `--user` : создаёт новый User namespace.
- `--map-root-user` : маппинг root пользователя внутри пространства имён.
- `--mount-proc` : монтирует новую файловую систему `/proc` внутри нового Mount namespace.

пример создание pid namespace

```
> sudo unshare --fork --pid --mount-proc /bin/bash

[root@archlinux mars]# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   7968  6404 pts/1    S    13:32   0:00 /bin/bash
root           2  0.0  0.0   9852  5976 pts/1    R+   13:32   0:00 ps aux
[root@archlinux mars]#
```

network namespace

```
> sudo unshare --net /bin/bash

[sudo] password for mars:
[root@archlinux mars]# ip link
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
[root@archlinux mars]#
```

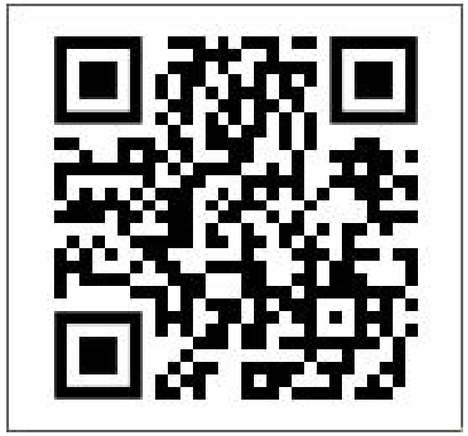
Комбинированный пример

```
 mars  ~  v3.13.7  13:45
→ sudo unshare --fork --pid --net --mount --uts --ipc --mount-proc /bin/bash

[root@archlinux mars]# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   7968  6380 pts/2    S    13:45   0:00 /bin/bash
root           2  0.0  0.0   9852  6004 pts/2    R+   13:46   0:00 ps aux
[root@archlinux mars]# ip link
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
[root@archlinux mars]# sleep 1000 &
[1] 4
[root@archlinux mars]# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   7968  6384 pts/2    S    13:45   0:00 /bin/bash
root           4  0.0  0.0   5744  3916 pts/2    S    13:47   0:00 sleep 1000
root           5  0.0  0.0   9852  6004 pts/2    R+   13:48   0:00 ps aux
[root@archlinux mars]#
```

```
 mars  ~  v3.13.7  13:47
→ ps aux | grep zsh
mars        4914  0.1  0.0  20980 18724 pts/0    Ss   13:06   0:02 /usr/bin/zsh
mars        9798  1.0  0.0  20944 18688 pts/1    Ss   13:45   0:01 /usr/bin/zsh
mars       10663  0.0  0.0   6620  4160 pts/1    S+   13:48   0:00 grep --color=auto zsh

 mars  ~  v3.13.7  13:48
→ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default
 qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DORMANT group
 default qlen 1000
    link/ether 00:45:e2:cb:45:65 brd ff:ff:ff:ff:ff:ff
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT
 group default
    link/ether f2:bb:e7:71:5f:76 brd ff:ff:ff:ff:ff:ff
4: br-0e33d5dbd8db: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode
 DEFAULT group default
    link/ether a2:3e:5f:3d:6f:95 brd ff:ff:ff:ff:ff:ff
5: br-5b4a454eeb3c: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode
 DEFAULT group default
    link/ether ae:ca:74:a6:50:b0 brd ff:ff:ff:ff:ff:ff

 mars  ~  v3.13.7  13:48
→ ps aux | grep sleep
root       10436  0.0  0.0   5744  3916 pts/2    S    13:47   0:00 sleep 1000
mars       10811  0.0  0.0   6620  4156 pts/1    S+   13:48   0:00 grep --color=auto sleep

 mars  ~  v3.13.7  13:48
→
```

```
 mars  ~  v3.13.7  15:52
→ sudo mkdir /sys/fs/cgroup/demo
[sudo] password for mars:

 mars  ~  v3.13.7  15:52
→ echo "10485760" | sudo tee /sys/fs/cgroup/demo/memory.max
10485760

 mars  ~  v3.13.7  15:52
→ sudo bash -c "echo \$\$ > /sys/fs/cgroup/demo/cgroup.procs && python3"

Python 3.13.7 (main, Aug 15 2025, 12:34:02) [GCC 15.2.1 20250813] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> data = []
>>>
>>> while True:
...     data.append(' ' * 1000000)
...

zsh: killed     sudo bash -c "echo \$\$ > /sys/fs/cgroup/demo/cgroup.procs && python3"

 mars  ~  v3.13.7  15:54
```

https://github.com/Jahamars/pycontainer



```
→ sudo python main.py shell

============================================================
Контейнер: demo
============================================================
✓ CGroup: 50MB RAM, 25% CPU
✓ RootFS: /tmp/container_demo_6ou3hx4w
□□ Команда: /bin/sh

/ # echo $$
1
/ # ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
/ # ls
bin   dev   etc   lib   lib64 proc  tmp   usr
/ #
```