Lab 2.1. Cgroups

Cgroups allow users to bundle processes together and limit, account and isolate the group's resources, such as CPU, memory, disk I/O, network, devices, and hugepages.

First, as **root**, let's install a tool that allows us to interact with cgroups:

```
student@ubuntu:~$ sudo apt update

student@ubuntu:~$ sudo apt install -y cgroup-tools

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libnuma1
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  libcgroup1
The following NEW packages will be installed:
  cgroup-tools libcgroup1
0 upgraded, 2 newly installed, 0 to remove and 6 not upgraded.
Need to get 108 kB of archives.
...
```

Now we can list all cgroups on our system:

```
student@ubuntu:~$ sudo lscgroup

rdma:/
net_cls,net_prio:/
hugetlb:/
memory:/
```

```
memory:/user.slice
memory:/system.slice
memory:/system.slice/systemd-update-utmp.service
memory:/system.slice/snap-core-8268.mount
memory:/system.slice/lvm2-monitor.service
...
```

And also list cgroups associated with a process:

```
student@ubuntu:~$ sudo cat /proc/<PID>/cgroup

student@ubuntu:~$ sudo cat /proc/1/cgroup

12:pids:/init.scope
11:rdma:/
10:memory:/init.scope
9:freezer:/
8:hugetlb:/
7:perf_event:/
6:cpuset:/
5:net_cls,net_prio:/
4:devices:/init.scope
3:cpu,cpuacct:/init.scope
2:blkio:/init.scope
1:name=systemd:/init.scope
0::/init.scope
```

Let’s explore a particular cgroup, called **freezer**, which allows a group of tasks to be suspended and then resumed. We will demonstrate that a **frozen** (suspended) process does not allow any operations on it until it is **thawed** (resumed). Let’s start by creating a new cgroup hierarchy under the freezer cgroup:

```
student@ubuntu:~$ cd /sys/fs/cgroup/freezer/

student@ubuntu:/sys/fs/cgroup/freezer$ sudo mkdir mycgroup

student@ubuntu:/sys/fs/cgroup/freezer$ cd mycgroup/

student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ ls

cgroup.clone_children  freezer.parent_freezing   freezer.state  tasks
cgroup.procs           freezer.self_freezing     notify_on_release
```

The new directory is populated by default upon its creation. The **tasks** file, initially empty, would otherwise hold PIDs of processes associated with the cgroup. Let's verify the empty file, then create a new process and associate it with our cgroup:

```
student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ cat tasks
```

Let's open a second terminal as a new **bash** process, and in the new terminal retrieve its PID:

```
student@ubuntu:~$ ps

  PID TTY          TIME CMD
20913 pts/1    00:00:00 bash
20943 pts/1    00:00:00 ps
```

Keep the second terminal running, and return to the first terminal to add the PID of the second terminal to the **tasks** file of the cgroup. (An "Invalid argument" or "No such process" error may be returned if an incorrect PID is used. A "Permission denied" error may be returned if the single quotes (') are misinterpreted after copy/paste, in which case a manual edit is recommended to delete and retype the two single quotes (')):

```
student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ sudo bash -c 'echo 20913 >>
tasks'

student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ cat tasks

20913
```

Now we should be able to freeze the processes associated with our cgroup:

```
student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ sudo bash -c 'echo FROZEN >
freezer.state'

student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ cat freezer.state

FROZEN
```

The state we just modified affects all the processes listed in the tasks file, that is the second terminal window. Return to the second terminal and try running the **date** command, for example. Nothing will be registered and displayed, as the process of the terminal is in a **frozen** (suspended) state.

Finally, let's **thaw** (resume) the second terminal process:

```
student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ sudo bash -c 'echo THAWED >
freezer.state'

student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ cat freezer.state
```

```
THAWED
```

Once **thawed** (resumed), the second terminal will display the `date` command we ran earlier, while it was in a **frozen** (suspended) state:

```
student@ubuntu:~$ date

Thu 10 Jun 2021 08:07:31 PM UTC
```

It is now safe to close the second terminal window (the one with the `date` command), and return to student user’s home directory in the first terminal window:

```
student@ubuntu:/sys/fs/cgroup/freezer/mycgroup$ cd

student@ubuntu:~$
```