Project 5: Evaluation of Two CPU Schedulers

CS 311

The stock Linux 2.6.27.1 kernel uses the CFS scheduler, which is "completely fair" at the task level. An enterprising user, however, can conceivably circumvent this fairness by writing a program that uses a large number of tasks. In this lab you will examine this behavior in the stock kernel and a Group CPU Scheduler-enabled version of this kernel.

References

The following are all available on, or linked to from, the course web site, or in the textbook:

1. CFS Scheduler Design, from the Linux kernel documentation provided with kernel source code.

Background

The CFS scheduler is designed so that each of n running tasks receives $\frac{1}{n}$ of CPU cycles. Focusing on two users, user M running m tasks and user N running n tasks, user M receives $\frac{m}{m+n}$ of CPU cycles. An enterprising user can exploit this to receive an unfair share of CPU cycles. The Linux 2.6.27.1 kernel can be configured via a "Group CPU Scheduler" extension to raise the "fairness" determination from the task-level to the userid-level. This closes the vulnerability. The "Group CPU Scheduler" can be enabled in menuconfig, under "General setup." Don't enable or modify any other features.

Method

Using the stock kernel, demonstrate the behavior described above for a sufficient number of values of m and n to be convincing. This data should be compiled into a table or graph. You will have to decide what, exactly, to measure and how to measure it. Most likely, you will need to create a tool that generates task loads, creating either multiple processes or multiple threads. (Pthreads on the Linux kernel we're using performs a one-to-one mapping between user-threads and kernel-threads, so it makes no difference if you work with processes or threads.)

You'll find it useful to create two test user accounts:

```
sudo useradd -c "Test User 1" -g staff -m tu1
sudo useradd -c "Test User 2" -g staff -m tu2
sudo passwd tu1
sudo passwd tu2
```

I'd suggest using the same password as you use for your `kdev` account. If changing the password as `root`, `passwd` may complain, but will obey.

To switch to a test user from kdev:

```
su - tu1
```

After collecting data for the stock kernel, repeat your experiments on the Group CPU Scheduler-enabled kernel, producing a similar table or graph.

Reporting

Write a lab report describing your experiment, method, and results. Your report should contain the following sections:

1. **Introduction**, describing the problem with the CFS scheduler and how the problem is addressed with the Group CPU Scheduler extension.
2. **Method**, describing your experiment, what you are measuring to demonstrate the problem, how you're performing the measurement, and any tools you used/created to assist with the measurement. The differences between the two kernels should be described in this section. This is important: Identify and justify the assumptions you make in your experimental methodology.
3. **Results and Discussion**, presenting and discussing the results of your measurements of the two kernels.
4. **Conclusion**
5. **Appendix**, containing any source code you write.

Deliverables

The following files should be uploaded to the course GoucherLearn site by 5:00 pm on the due date:

1. Your lab report.