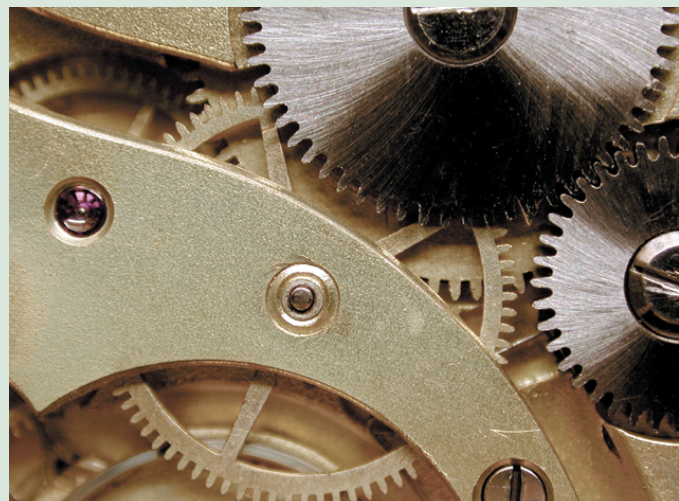## The Sysadmin's Daily Grind: Fcron

On Schedule

Cron executes the jobs you hand to it reliably and on schedule. This legacy tool just ticks away like a Swiss watch, completely unfazed by external influences – unfortunately that is not always a good thing. **BY CHARLY KÜHNAST**

Routine tasks, such as rotating log-files, deleting temp files, or performing backups in the early hours of the morning are best left to cron. This is fine until one job takes longer than expected and overlaps the time of another. In the past, I have found myself yearning for a tool that could gracefully handle that kind of situation, and now I've found exactly that tool: Fcron [1].

Fcron version 2.0.1. has just gone stable. Fcron's author. Thibault Godouet, looks set to put the good old vixie cron out of a job. To do so, Thibault has equipped the tool with all kinds of extra goodies. Installing fcron is trivial. After unpacking the 142KBytes source files, simply enter:

```
./configure && make && ⤵
make install
```

If so desired, the installation routine will add the program to your startup scripts. Following the pattern typical of cron, fcron uses an fcrontab file. You can enter *fcrontab -e* to edit the file. This is where the similarities between the old and new versions stop. Although the structure and syntax of fcrontab are based on the legacy role model, there are some differences.

To get started, let's tell fcron to do something that cron can do: I want fcron to run the */bin/date* command at five minutes to every full hour. The *fcrontab* entry looks like this:

```
5 0 * * * /bin/date
```

What's new with fcron is its ability to launch commands relative to a computer's uptime. I want to run */bin/date* every two and a half hours, starting at system boot time:

```
@ 2h30m /bin/date
```

Be Nice!

You can assign a nice value to a program for jobs that put a heavy load on the system:

```
&nice(10) 0 1 * * * /usr/local⤵
/shellscripts/MyBackupScript.sh
```

The following line runs a command on the hour between 2.00 pm and 10 pm:

```
0 14-22 * * * /bin/date
```

To launch *date* every second(!) hour, I need to modify the entry as follows:

```
0 14-22/2 * * * /bin/date
```

Fcron also supports exceptions: to run a command on weekdays but not on Sundays, another slight change to the command is required:

```
0 14-22/2 * * *~0 /bin/date
```

In other words, fcron treats elements with the tilde character (~) as exceptions. Fcron's ability to check the system load is one of its neatest features. The following line launches a command called */bin/heavyjob/* once an hour – but only if the mean system load (loadavg) over a five minute period is below a threshold of 1.5:

```
@lavg5(1.5) 1h /bin/heavyjob
```

The useful manpage installs along with fcron or fcrontab. It contains detailed references for fcron options, and examples. In my case at least, the good old cron tool is out of a job. ∎

THE AUTHOR

Charly Kühnast is a Unix System Manager at the data-center in Moers, near Germany's famous River Rhine. His tasks include ensuring firewall security and availability and taking care of the DMZ (demilitarized zone). Although Charly started out on IBM mainframes, he has been working almost exclusively with Linux since 1995. To stay in shape he tries to get in some karate training on his leisure time.