

Aus dem Nähkästchen geplaudert: Cron, At, Anacron

# Pünktlich wie die Feuerwehr

Viele Aufgaben fallen bei einem Unix-System regelmäßig an: Logdateien rotieren, Backups erstellen, Indizes und Datenbanken aktualisieren. Traditionell kümmern sich die Programme Cron und At darum, diese Arbeiten zu festgelegten Zeiten zu erledigen. Anacron bedient die Langschläfer. Marc André Selig



Cron startet während des Bootvorgangs und prüft von da ab jede Minute, ob Jobs anstehen. Außerdem prüft der Daemon, ob sich an den Zeitplänen etwas geändert hat, sodass Änderungen ohne Neustart wirksam sind. **Abbildung 1** zeigt einen solchen Zeitplan, auch Crontab genannt. Wie unter Unix üblich, sind Zeilen, die mit einem »#«-Zeichen anfangen, Kommentare.

## Flexible Syntax

Jede Job-Anweisung besteht aus einer Zeitangabe und dem auszuführenden Programm mit eventuellen Parametern. Die Zeitangabe besteht aus fünf Feldern, und zwar für Minute, Stunde, Tag, Monat und Wochentag. Die Wochentage sind einfach durchnummeriert, wobei »0« für Sonntag steht. Wer die Woche lieber mit Montag beginnen will, benutzt »7« für den Sonntag. Admins, die jetzt schon durcheinander kommen, können beruhigt sein: Ein Aufruf von »man 5 crontab« zeigt die Syntax der Crontabs detailliert an. Selbst erfahrene Sysadmins greifen gerne noch zu dieser Manpage, wenn es darum geht, einen neuen Job einzurichten.

Statt statischer Zeitangaben ist es auch möglich, Bereiche festzulegen. Ein »\*« steht für einen Bereich von Anfang bis Ende, also zum Beispiel für jede Minute. Die Anweisung »\* \* \* \* \*« legt fest, dass das Programm jeden Tag, zu jeder Stunde und jeder Minute starten soll. »37 \* \* \* \* \*« aktiviert es immer 37 Minuten nach der vollen Stunde. Es ist übrigens empfehlenswert, nicht die 0 zu benutzen, sondern krumme Werte wie 37 oder 21. Damit verringert sich die Wahrscheinlichkeit, dass viele Jobs gleichzei-

**Wer sich die** Liste laufender Prozesse auf seinem Unix-System einmal mit dem Aufruf »ps -ef|grep (cron|atd)« näher ansieht, findet dort auch meist die beiden Daemons »atd« und »cron«. Beide sind dafür zuständig, Programme zeitgesteuert zu starten. Bei Cron legt der Administrator vorher fest, was zu welchem Zeitpunkt und wie oft passieren soll. Viele Linux-Distributionen enthalten schon einige vordefinierte so genannten Cronjobs.

Besonders praktisch ist das für aufwändige Jobs, die selbst einen modernen Computer vorübergehend auslasten und somit die reguläre Arbeit der Nutzer stören könnten. Daher legt der Administrator oder der Distributor sie in die Nacht, wenn wenige oder gar keine Benutzer am System arbeiten.

Ein typisches Beispiel ist die Datenbank von »locate«. Dieses Programm dient dazu, Dateien schnell zu finden, indem es einen Index aller vorhandenen Dateien im System erstellt. Um auf dem

neuesten Stand zu bleiben, muss dieser Index aber regelmäßig aktualisiert werden. Auch die Datenbank für »whatis« und »apropos« muss das System stets aktuell halten.

## Fester Zeitplan

Ein weiterer wichtiger Verwendungszweck von Cron ist das Steuern von Wartungsarbeiten, die das System konsistent halten. Beispielsweise sollte der Administrator Protokolldateien regelmäßig rotieren lassen, um Plattenplatz zu sparen. Zusätzlich ist es unter Umständen erforderlich, diese mit Gzip oder Bzip 2 zu komprimieren.

Es gibt zwar eine Menge Programme, die sich um die zeitgesteuerte Ausführung von Jobs kümmern, doch Cron (auch bekannt unter dem Namen Vixie Cron, nach seinem Erfinder Paul Vixie) ist der U(h)rvater dieser Software und viele Programme sind Derivate des ursprünglichen Vixie-Daemon.

```

$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file.
# This file also has a username field, that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
25 12 * * * root    test -e /usr/sbin/anacron || run-parts --report /etc/cron.daily
47 12 * * 7 root    test -e /usr/sbin/anacron || run-parts --report /etc/cron.weekly
52 12 1 * * root    test -e /usr/sbin/anacron || run-parts --report /etc/cron.monthly
#
$

```

**Abbildung 1:** Die Datei »/etc/crontab« ist die für das ganze System zuständige Master-Crontab. Meist kommt sie schon vorgefertigt vom Distributor auf den Rechner.

tig ablaufen und somit die Last steigt. Die Felder für Tag und Wochentag haben eine gesonderte Bedeutung: Sind beide Felder auf einen Wert (oder auf einen Bereich) gesetzt, verknüpft Cron sie zu einer Entweder-oder-Bedingung. So startet »37 2 1 \* 0« einen Befehl also um 02:37 Uhr an jedem ersten des Monats sowie an jedem Sonntag.

## Eigene Crontab

Zusätzlich zur systemweiten »/etc/crontab« hat jeder Benutzer die Möglichkeit, eine eigene Crontab einzurichten. Dazu dient der Befehl »crontab -e« (für Edit). Das Programm startet einen Editor und lädt die Crontab des Anwenders aus einer temporäre Datei. Nach dem Anpassen der Datei und Schließen des Editors kopiert »crontab« die Datei wieder ins Spoolverzeichnis von Cron (meist »/var/spool/cron/crontabs/«). In diesem Verzeichnis liegen alle Crontabs der lokalen Benutzer.

Der Befehl »crontab -l« gibt die Jobliste eines Benutzers auf der Konsole aus. Anwender, die ihre Crontab im Heimatverzeichnis aufbewahren, aktualisieren die entsprechende Datei im Spoolverzeichnis mit »crontab ~/Dateiname«.

Nur der Administrator des Rechners bearbeitet die systemweite Crontab. In **Abbildung 1** ist zu sehen, dass diese Datei möglichst klein gehalten ist und lediglich Programme in Cron-Unterverzeichnissen aufruft. Die meisten Distributionen haben in den Verzeichnissen »cron.daily«, »cron.weekly« und »cron.monthly« unterhalb von »/etc/« bereits entsprechende Bash-Skripte oder auch Binaries abgelegt.

Diese Unterteilung in Verzeichnisse hilft dabei, die Master-Crontab übersichtlich

zu halten. Immer wenn der Administrator ein neues Paket installiert, das Cronjobs benötigt, kopiert das Installationssystem der Distribution einfach die Skripte in eines der Cron-Verzeichnisse. Eine Änderung der Master-Datei ist nicht nötig. Die systemweite Crontab hat außerdem ein zusätzliches Feld in der Job-Angabe, nämlich die Benutzerkennung, mit der der Job laufen soll. Es steht direkt hinter der Zeitangabe. Mit dem Programm Fcron [1] lassen sich noch weitere Parameter angeben.

## Spiel's nur einmal

Während Cron gemäß den Crontabs immer wiederkehrende Aufgaben koordiniert, sorgt At für einmalige Arbeiten. Es erledigt den Job und löscht ihn dann von seiner Liste. Das ist nützlich, um beispielsweise eine Erinnerungs-Mail zu versenden. Daher ist At fast ausschließlich interaktiv zu verwenden.

Als Argument erwartet das Programm eine Zeitangabe, die sehr frei formuliert sein darf. Um eine Aufgabe auf Mitternacht zu legen, reicht der Aufruf von »at midnight«, oder in fünf Stunden »at now + 5 hours«. Natürlich sind auch ganz präzise Zeitangaben möglich: »at 05:45 11 Jun 2005«. Nach Eingabe des »at«-Kommandos befindet sich der Nutzer an einem Prompt, an dem er einfach Shell-Befehle eingibt. Mit [Strg] + [D] kehrt er wieder zur Shell zurück und At speichert den Job. **Abbildung 2** zeigt ein Beispiel, bei dem At in fünf Stunden von jetzt an eine E-Mail versendet.

At geht bei der Ausführung von Jobs intelligent vor. Es merkt sich die vom Benutzer verwendete Shell (die Einfluss auf die Befehlssyntax hat), das aktuelle Arbeitsverzeichnis sowie einige Umge-

bungsvariablen. Der Benutzer gibt also einfach die Befehle so ein, als befände er sich in einer Shell.

Ausnahmen sind Befehle, die eine grafische Oberfläche voraussetzen, denn At speichert die Variable »\$DISPLAY« nicht. Das Problem ist aber leicht zu beheben, wenn der Benutzer statt »xeyes« einfach »DISPLAY = :0.0 xeyes« eingibt. Mechanismen wie der SSH-Agent greifen nicht, sobald der User ausgeloggt ist.

## Hohe Last vermeiden

Im Spoolverzeichnis von At (das ausschließlich von Root einzusehen ist) stehen die Jobs als Shellskripte. Ohne Root-Rechte verschafft »atq« eine Übersicht über die anstehenden Jobs, »atrm« entfernt sie wieder aus der Warteschlange. Benutzer, die zeitraubende Jobs starten wollen, hängen auf dem At-Prompt einfach ein »&« an, damit das Programm im Hintergrund weiterläuft.

Tun das jedoch 20 Mitbenutzer ebenso, steigt die Last des Systems unter Um-

ständen ins Unerträgliche. Daher gibt es die Sonderform von At namens »batch«. Es nimmt wie At Befehle entgegen, startet sie aber nur, wenn die Systemlast einen bestimmten Wert nicht überschreitet. Dieser Wert ist fest in den At-Daemon einkompiliert, er lässt sich ändern, indem der Administrator den »atd« mit der Option »-l Wert« startet.

In klassischen Unix-Systemen starten die At-Jobs über einen Cron-Eintrag, der jede Minute das Programm »atrun« aufruft. Bei den meisten aktuellen Linux-Distributionen läuft jedoch ein echter Daemon namens »atd«.

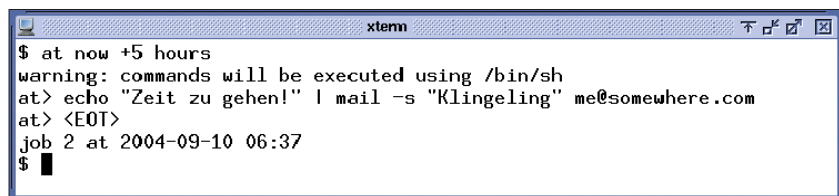
## Anacron für Schlafmützen

Cron und At gehen davon aus, dass der Rechner 24 Stunden am Tag eingeschaltet ist. Denn die Programme stammen noch aus Zeiten, in denen Unix fast ausschließlich auf Server-Computern lief. Heim- und Office-Anwender schalten jedoch ihre Rechner oft aus, Laptop-Benutzer legen sie darüber hinaus ab und

zu schlafen. In diesen Ruhezeiten läuft natürlich auch kein Cron- und kein At-Job. Mit etwas Pech – wenn der Benutzer beispielsweise seinen Rechner niemals nachts einschaltet – werden manche Aufgaben nie ausgeführt. Das wäre bei Routinen wie der Aktualisierung der Locate-Datenbank fatal.

Anacron ist eines von mehreren Programmen, die sich dieses Mankos annehmen. Es eignet sich für Befehle, die ungefähr einmal täglich oder seltener ausgeführt werden sollen, also vor allem für die oben angesprochenen Aufgaben rund um die Systemwartung. Die genaue Zeit, zu der ein Befehl startet, ist dabei unwichtig.

Wenn der Computer bootet und Anacron feststellt, dass ein Befehl fällig oder überfällig ist, startet es ihn. Damit der Rechner während des Bootvorgangs nicht unerträglich langsam wird, baut Anacron Verzögerungen ein und sorgt dafür, dass die Jobs niemals gleichzeitig, sondern nur nacheinander starten. Lässt der Anwender seinen Computer ausnahmsweise doch einmal nachts durchlaufen, startet Anacron alle Jobs zu ihren regulären Zeiten und wartet das System ganz unauffällig. (*mwe*) ■



**Abbildung 2:** Das Programm At dient dazu, nur einmal auftretende Arbeiten zu verrichten. Hier lässt sich der Benutzer in fünf Stunden eine E-Mail schicken. Der Job startet um 06:37 Uhr.

### Infos

**[1]** Charly Kühnast, „Aus dem Alltag eines Sysadmin: Fcron“: Linux-Magazin 07/04, S. 59