

Penjadwalan Tugas Menggunakan Cron dan AT

Pada suatu waktu, mungkin Anda membutuhkan sistem bekerja pada tengah malam ketika penggunaan sistem dalam kondisi tidak sibuk. Atau Anda menginginkan suatu pekerjaan yang dilakukan secara harian atau mingguan secara otomatis? Jika ya, gunakan saja aplikasi cron dan at yang memiliki kemampuan untuk penjadwalan tugas secara periodik.

Tugas menjadi seorang sistem administrator, terkadang banyak berhubungan dengan penjadwalan tugas yang harus dijalankan secara berkala dalam kurun waktu tertentu pada sistem Linux. Salah satu contoh pekerjaan ini, di antaranya adalah pekerjaan merotasi file log sehingga kapasitas filesystem menjadi tidak penuh, *back-up* data, dan konek ke time server untuk menjaga agar sistem time Anda selalu di sinkronisasi.

Tugas lain yang tidak ada kaitannya dengan sistem, terkadang juga membutuhkan utiliti penjadwalan tugas yang dapat menjalankan tugas pada kurun waktu tertentu. Misal, jika ingin men-*download* suatu file yang lumayan besar pada saat load sistem Internet sedang tidak ada yang menggunakan.

Untuk melakukan hal ini, sudah tersedia sejumlah utiliti di Linux yang berfungsi untuk melakukan hal ini. Salah satu yang paling banyak digunakan dan cukup andal untuk kebutuhan ini adalah cron dan at. Kedua aplikasi ini biasa dikemas secara *default* oleh kebanyakan distro Linux, karena kemudahan penggunaan dan kestabilan aplikasi ini. Utilitas Cron yang dibuat oleh Paul Vixie ini, dapat menjalankan suatu perintah sesuai dengan waktu dan hari yang telah ditentukan. Dengan ini, Anda tidak perlu lagi menunggu untuk sekadar menjalankan suatu perintah pada waktu yang Anda inginkan.

Utiliti at yang dibuat oleh Thomas Koenig, juga dapat digunakan untuk menunggu, menjalankan, atau menghapus suatu tugas, untuk kemudian dieksekusi. Perbedaannya dengan

cron, at lebih cocok digunakan untuk kebutuhan penjadwalan yang dilakukan dalam tempo satu kali saja. Hal ini berbeda dengan cron yang ditujukan untuk penjadwalan tugas yang dilakukan secara berulang.

Pada tutorial kali ini, akan dibahas beberapa hal yang berkaitan dengan penjadwalan tugas di Linux dengan menggunakan crontab dan at.

Menjalankan tugas setiap kurun waktu tertentu

Untuk dapat menjalankan tugas dalam interval waktu tertentu, kita dapat menggunakan utiliti cron, yang terdiri dari *crond* daemon dan sekumpulan tabel yang mendeskripsikan apa yang akan dilakukan dan berapa lama frekuensi itu berlangsung. Daemon akan berjalan setiap waktu dan mengecek crontabs untuk menentukan apa yang dibutuhkan. Para pengguna dapat manajemen crontab dengan menggunakan perintah crontab. Daemon *crond* biasanya dapat berjalan dengan menggunakan proses *init* pada saat startup sistem.

Sebagai latihan, dalam contoh berikut kita akan membuat sebuah bash script sederhana, yang kemudian kita gunakan crontab untuk menyetup cron job. Skrip ini dapat menampilkan hari dan waktu yang sedang berjalan pada saat itu, sehingga pada saat cron berjalan, kita akan mengetahui *output* dari skrip ini. Mengonfigurasi entiti crontab membutuhkan string dengan escaped shell metacharacter, sehingga cara ini lebih mudah dilakukan

dengan menggunakan parameter dan perintah yang simpel. Dalam contoh ini, perintah *echo* akan menjalankan skrip yang terdapat pada `/home/supriyanto/mycrontab.sh`, tanpa menggunakan parameter. Berikut listing *mycrontab.sh*.

```
#!/bin/bash
echo "It is now $(date +%T) on
$(date +%A)"
```

Membuat sebuah crontab

Untuk membuat sebuah crontab, Anda dapat menggunakan perintah crontab ditambah dengan option `-e` (untuk "edit") option. Perintah tersebut secara otomatis akan membuka editor vim, jika Anda tidak mendefinisikan editor lainnya di *environment* variabel *EDITOR* atau *VISUAL*.

Setiap entri yang terdapat pada crontab terdiri atas enam bagian, yaitu:

1. Menit.
2. Jam.
3. Hari dalam sebulan.
4. Bulan dalam setahun.
5. Hari dalam seminggu.
6. String atau perintah yang akan dieksekusi oleh `sh/bash`.

Penjelasan masing-masing field:

- Menit dan jam berkisar antara 0-59 dan 0-12.
- Entri hari dalam sebulan dan bulan dalam setahun, berkisar antara 1-31 dan 1-12.
- Hari dalam seminggu berada dalam kisaran 0-6, di mana 0 adalah Minggu.

```

supriyanto@debian:~$ cat mycronstab.sh
#!/bin/sh
echo "It is now $(date +%T) on $(date +%A)"
supriyanto@debian:~$
    
```

Isi skrip mycronstab.sh.

```

# m h dom mon dow command
0,20,40 22-23 * 7 fri-sat /home/supriyanto/mycronstab.sh
    
```

Mengedit file crontab untuk menjalankan skrip mycronstab.sh.

- Hari dalam seminggu, dapat juga di tulis sebagai sun, mon, tue, dan seterusnya.
- Field keenam adalah inti dari entri ini. Field ini mendefinisikan string yang selanjutnya akan dijalankan oleh sh. Tanda persen (%) di terjemahkan sebagai ganti baris. Jadi jika Anda tetap menginginkan penggunaan a% atau karakter spesial lainnya, Anda dapat memisahkannya dengan menggunakan backslash (\). Baris yang mengandung % pertama akan di teruskan ke shell, di mana semua baris setelah % akan diteruskan sebagai standar input.

Period

Beragam waktu yang berhubungan dengan fields, juga dapat menspesifikasikan individual value, range of value, seperti 0-10 atau sun-wed, atau daftar pemisahan koma dari jangkauan dan nilai individual. Cara mendefinisikan entry crontab untuk perintah yang telah kita buat sebelumnya, akan terlihat seperti baris berikut:

```
0,20,40 22-23 * 7 fri-sat /home/supriyanto/mycronstab.sh
```

Dalam contoh di atas, perintah akan dijalankan setiap menit ke 0, 20, dan 40 (setiap 20 menit), antara jam 10-11 malam di setiap hari Jumat dan Sabtu yang terdapat selama bulan Juli. Anda dapat melihat option lengkap crontab di manual page crontab.

Bagaimana dengan output-nya?

Setelah mengikuti perintah di atas, dan memasukkan perintah tersebut ke dalam crontab, mungkin sebagian Anda ada yang bertanya dimana hasil outputnya? Kebanyakan perintah di desain agar dapat menggunakan cron dengan kemampuan menghasilkan output log menggunakan syslog. Setiap output akan di berikan ke stdout untuk di

kiriman ke mail user yang bersangkutan. Berikut contoh output cron yang dihasilkan.

```

From root@debian.example.com Wed
Aug 1 03:40:01 2007
X-Original-To: supriyanto
Subject: Output from your job
1
To: supriyanto@debian.example.com
Date: Wed, 1 Aug 2007 03:40:00
+0700 (WIT)
From: root@debian.example.com (root)

-It is now 03:40:00 on Wednesday
    
```

Crontab yang Anda buat dengan perintah crontab, akan disimpan pada direktori /etc/spool/cron, dalam folder nama user yang membuatnya. Jadi pada contoh diatas, hasilnakan disimpan pada direktori /etc/spool/cron/supriyanto.

/etc/crontab

Sebagai tambahan file user crontab yang terdapat di /var/spool/cron, cron juga mengecek /etc/crontab dan file-file yang terdapat dalam direktori /etc/cron.d. Sistem crontab ini, memiliki satu tambahan field antara entry field kelima dan keenam. Tambahan field ini menspesifikasikan nama user yang menjalankan perintah tersebut, yang biasanya adalah user root. Isi file /etc/crontab mungkin akan terlihat seperti contoh berikut:

```

SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
    
```

```

22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
    
```

Pada contoh ini, terlihat jelas kalau pekerjaan sebenarnya dilakukan oleh perintah run-parts, dimana skrip tersebut menjalankan /etc/cron.hourly, /etc/cron.daily, dan sebagainya. File /etc/crontab dapat dengan mudah mengontrol waktu untuk menjalankan tugas. Crontab juga dapat menyertakan shell variables assignment yang dapat di set sebelum perintah dijalankan.

Anacron

Cron hanya dapat bekerja dengan baik pada sistem yang berjalan secara berkelanjutan. Untuk sistem yang sering kali dimatikan, seperti misal laptop, aplikasi lain yang bernama anacron dapat menangani penjadwalan tugas yang dilakukan perhari, perminggu, atau perbulan dengan menggunakan utilitas cron. Anacron tidak dapat menangani tugas per-jam.

Anacron meletakkan file timestamp pada /var/spool/anacron untuk mencatat setiap tugas yang sedang berjalan. Saat anacron berjalan, aplikasi ini akan memeriksa apakah masih terdapat sejumlah tugas pada hari itu, dan menjalankan kembali tugas itu jika memang diperlukan. Tabel dari pekerjaan yang dilakukan oleh anacron disimpan dalam file /etc/anacrontab, yang memiliki perbedaan format dengan file /etc/crontab. Sama halnya dengan file /etc/crontab, file /etc/anacrontab, juga menyediakan beberapa parameter konfigurasi. Setiap tugas terdiri atas empat field, yakni:

1. Period.
2. Delay.
3. Job Identifier.
4. Command.

```

# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-pa
rts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-pa
rts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-pa
rts --report /etc/cron.monthly )
#
"/etc/crontab" 19L, 745C written          7,15          50%

```

Isi file /etc/crontab.

Penjelasan masing-masing field:

- Period merupakan angka dalam hari, tetapi mungkin hanya dispesifikasikan sebagai @monthly untuk memastikan bahwa suatu tugas berjalan hanya sekali dalam sebulan.
- Delay merupakan angka dalam hitungan menit, untuk menunggu setelah tugas dilakukan dengan tujuan untuk menjalankan tugas sebenarnya sebelum memulainya. Hal ini cukup berguna untuk menangani banyaknya tugas dalam satu waktu, ketika tugas sebelumnya sedang berjalan.
- Job identifier dapat menyediakan beberapa blank character selain karakter slash (/).
- Command merupakan perintah yang akan di jalankan dalam waktu yang telah ditentukan oleh perintah at.

Baik file /etc/crontab maupun /etc/anacrontab akan selalu di update setiap kali di edit langsung. Kita juga tidak dapat menggunakan perintah crontab untuk mengupdate file ini, atau file yang terdapat pada direktori /etc/cron.d.

Menjalankan tugas pada waktu yang ditentukan

Terkadang kita menginginkan untuk menjalankan suatu tugas hanya berlangsung sekali, daripada berulang-ulang. Untuk tujuan ini, kita dapat menggunakan perintah at. Perintah ini akan berjalan setelah membaca file yang telah dispesifikasikan dengan option -f, atau dari stdin, jika -f

tidak digunakan. Sedangkan, option -m digunakan untuk mengirimkan email ke user ketika tidak terdapat stdout dari perintah yang dijalankan. Option -v menampilkan waktu ketika tugas akan dijalankan sebelum membaca tugas. Waktu juga dapat ditampilkan sebagai output.

Perintah berikut akan menjalankan skrip mycronab.sh yang Anda buat sebelumnya.

```
$ at -f mycronab.sh -v 04:31
```

```
Wed Aug 1 04:31:00 2007
```

Output tugas yang dilakukan oleh at:

Message 1:

From supriyanto@debian.example.com

Wed Aug 1 04:31:00 2007

X-Original-To: supriyanto

Subject: Output from your job

```

supriyanto@debian:~$ at -f mycronab.sh -v 04:31
Thu Aug 2 04:31:00 2007

warning: commands will be executed using /bin/sh
job 17 at Thu Aug 2 04:31:00 2007
supriyanto@debian:~$

```

Membuat penjadwalan tugas skrip mycronab.sh menggunakan at.

```

5
To: supriyanto@debian.example.com
Date: Wed, 1 Aug 2007 04:31:00
+0700 (WIT)
From: supriyanto@debian.example.com
(supriyanto)

```

It is now 04:31:00 on Wednesday

Selain menggunakan spesifikasi waktu seperti contoh di atas, kita juga dapat menggunakan spesifikasi waktu yang lebih kompleks seperti contoh berikut.

```
$ at -f mycronab.sh 10pm tomorrow
```

```
job 7 at Thu Aug 2 22:00:00 2007
```

```
$ at -f mycronab.sh 2:00 tuesday
```

```
job 8 at Tue Aug 7 02:00:00 2007
```

```
$ at -f mycronab.sh 2:00 july 11
```

```
job 9 at Fri Jul 11 02:00:00 2008
```

```
$ at -f mycronab.sh 2:00 next week
```

```
job 10 at Wed Aug 8 02:00:00 2007
```

Perintah at juga memiliki option -q, untuk meningkatkan waktu queue sehingga menambah hasil yang baik bagi suatu tugas. Selain perintah at, terdapat juga sejumlah perintah batch yang lain, yang hanya dapat menjalankan tugas pada saat load sistem sedang rendah.

Manajemen penjadwalan tugas

Pada bagian ini kita akan membahas sejumlah manajemen penjadwalan tugas berbasis cron maupun at, seperti cara menampilkan daftar tugas, cara menghapus suatu tugas yang

masih terdapat dalam daftar tugas, dan konfigurasi akses user untuk penjadwalan tugas.

Menampilkan daftar tugas

Kita dapat memanajemen tugas yang terdapat pada cron dan at. Gunakan perintah `crontab` dengan option `-l` untuk melihat daftar tugas yang terdapat pada `crontab`. Anda juga dapat menggunakan perintah `atq` untuk menampilkan daftar tugas yang sedang queue saat menggunakan perintah `at`. Berikut contoh tampilan daftar tugas yang masih terdapat pada `crontab`.

```
$ crontab -l
0,20,40 22-23 * 7 fri-sat /home/
supriyanto/mycrontab.sh
```

```
$ atq
6 Thu Aug 2 22:00:00 2007 a
supriyanto
9 Fri Jul 11 02:00:00 2008 a
supriyanto
10 Wed Aug 8 02:00:00 2007 a
supriyanto
8 Tue Aug 7 02:00:00 2007 a
supriyanto
7 Thu Aug 2 22:00:00 2007 a
supriyanto
```

Jika Anda ingin melihat perintah sebenarnya yang telah terdapat dalam jadwal yang akan dieksekusi oleh `at`, Anda dapat menggunakan perintah `at` dengan option `-c` dan job number. Anda akan diberi peringatan lingkungan kerja yang aktif pada saat menjalankan perintah `at`, dan kejadian yang terdapat akan disimpan dalam daftar tugas.

Menghapus daftar tugas

Kita dapat menghapus semua jadwal tugas cron dengan menggunakan option `-r` yang terdapat pada perintah `crontab`.

Berikut langkah menghapus suatu tugas yang masih terdapat di cron.

```
$ crontab -l
# m h dom mon dow command
0,20,40 22-23 * 7 fri-sat /home/
supriyanto/mycrontab.sh
```

```
$ crontab -r
```

```
$ crontab -l
no crontab for supriyanto
```

Untuk menghapus sistem cron atau tugas yang terdapat pada `anacron`, Anda dapat mengedit file `/etc/crontab`, `/etc/anac-`

```
supriyanto@debian:~$ at -f mycrontab.sh 10pm tomorrow
warning: commands will be executed using /bin/sh
job 13 at Thu Aug 2 22:00:00 2007
supriyanto@debian:~$ at -f mycrontab.sh 2:00 tuesday
warning: commands will be executed using /bin/sh
job 14 at Tue Aug 7 02:00:00 2007
supriyanto@debian:~$ at -f mycrontab.sh 2:00 july 11
warning: commands will be executed using /bin/sh
job 15 at Fri Jul 11 02:00:00 2008
supriyanto@debian:~$ at -f mycrontab.sh 2:00 next week
warning: commands will be executed using /bin/sh
job 16 at Wed Aug 8 02:00:00 2007
supriyanto@debian:~$
```

Beberapa pendefinisian waktu di `at` yang lebih kompleks.

`rontab`, atau mengedit/menghapus file yang terdapat pada direktori `/etc/cron.d`.

Anda juga dapat menghapus satu atau lebih tugas yang telah dijalankan sebelumnya dengan menggunakan perintah `at`, dengan menggunakan perintah `atrm` dan angka tugas yang sedang dijalankan. Penghapusan beberapa tugas dapat dipisahkan dengan menggunakan spasi. Berikut contoh penghapusan tugas dengan menggunakan `atrm`.

```
$ atq
6 Thu Aug 2 22:00:00 2007 a
supriyanto
9 Fri Jul 11 02:00:00 2008 a
supriyanto
10 Wed Aug 8 02:00:00 2007 a
supriyanto
8 Tue Aug 7 02:00:00 2007 a
supriyanto
7 Thu Aug 2 22:00:00 2007 a
supriyanto
```

```
$ atrm 8 9 10
```

```
$ atq
6 Thu Aug 2 22:00:00 2007 a
supriyanto
7 Thu Aug 2 22:00:00 2007 a
supriyanto
```

Hasil output diatas menunjukkan, kalau tugas nomor 8, 9, dan 10 yang terdapat pada `at` sudah dihapus dari daftar tugas, sehingga hanya tersisa tugas nomor 6 dan 7.

Konfigurasi akses user untuk penjadwalan tugas

Jika terdapat file `/etc/cron.allow`, setiap user

selain root harus terdaftar agar dapat menggunakan utiliti cron dan `crontab`. Jika `/etc/cron.allow` tidak ada, tapi `/etc/cron.deny` ada, maka user selain root yang terdaftar di sana tidak akan dapat menggunakan aplikasi cron ataupun `crontab`. Jika kedua file itu ada, hanya super user yang diizinkan untuk dapat menggunakan cron dan `crontab`. Jika file `/etc/cron.deny` kosong, maka semua user diizinkan untuk dapat menggunakan utiliti cron secara default. Keterkaitan antara file `/etc/at.allow` dan `/etc/at.deny` memiliki pengaruh yang sama seperti cron untuk penggunaan utiliti `at`.

Demikian tutorial singkat mengenai penjadwalan tugas menggunakan cron dan `at`. Kedua aplikasi open source ini merupakan aplikasi yang hebat, dan cukup handal untuk digunakan. Tidak heran jika hampir semua platform berbasis Unix, memaketkan kedua aplikasi ini pada paket distribusinya.

Hampir semua distro Linux, juga menggunakan secara default aplikasi ini untuk penjadwalan tugasnya. Kebanyakan aplikasi di Linux yang membutuhkan penjadwalan tugas untuk dapat berjalan pada waktu tertentu, juga menggunakan aplikasi `crontab` untuk kebutuhan aplikasinya. Untuk lebih mempermudah penggunaan aplikasi ini, dokumentasi berbantuan manual pages yang sudah disertakan pada kemasan paket ini, sudah sangat mencukupi jika Anda ingin mengenal lebih lanjut sejumlah parameter atau option lain yang terdapat pada aplikasi cron dan `at`. Tinggal diperlukan semangat untuk mempelajari lebih dalam penggunaan kedua aplikasi ini. Akhir kata, selamat mencoba!

Supriyanto [supriyanto@infolinix.co.id]