CRON

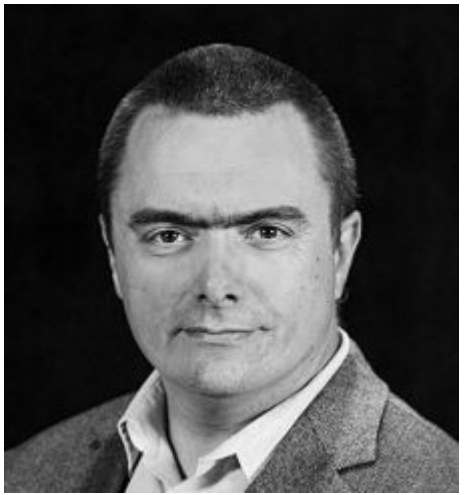
Advanced

Additions

Alternatives

CRON

- Paul Vixie
  - bind (DNS-server)
  - DHCP протокол
  - sendmail
  - cron



crontabs

- /etc/crontab
  - /etc/cron.hourly
  - /etc/cron.daily
  - /etc/cron.weekly
  - /etc/cron.monthly
  - /etc/cron.d/

```
# /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
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

- /etc/cron.daily:
  - 0anacron
  - apport
  - apt-compat
  - cracklib-runtime
  - dpkg
  - logrotate
  - man-db
  - plocate
  - sysstat

crontabs

- /etc/crontab
  - /etc/cron.hourly
  - /etc/cron.daily
  - /etc/cron.weekly
  - /etc/cron.monthly
  - /etc/cron.d/

```
# /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
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

- /etc/cron.daily:
  - 0anacron
  - apport
  - apt-compat
  - cracklib-runtime
  - dpkg
  - logrotate
  - man-db
  - plocate
  - sysstat

crontab caveates

- man 5 crontab:
  - Percent-signs (%) in the command, unless escaped with backslash (\), will be changed into newline characters, and all data after the first % will be sent to the command as standard input.
- SHELL=/bin/sh
  - * * * * * set > /tmp/env.txt
- PATH='/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin'

Typical usecase: frequently than every minute

- ```
  # once at  30 sec:
  * * * * * /path/to/script.sh
  * * * * * sleep 30 ; /path/to/script.sh
  ```
- `while true; do doing_something; sleep 10s; done`
- `homer6/frequent-cron (stale from 2012)`
  ```

Alternatives and Additions

- crontab-like alternatives:
  - cronie
    cronie-crond/cronie
  - dcron
    dubiousjim/dcron
  - fcron
    yo8192/fcron
  - bcron
    bruceg/bcron
  - homer6/frequent-cron
- Additions:
- anacron
- at/atq/atrm
- systemd timers
- inotify

Alternatives and Additions

- crontab-like alternatives:
  - cronie
    cronie-crond/cronie
  - dcron
    dubiousjim/dcron
  - fcron
    yo8192/fcron
  - bcron
    bruceg/bcron
  - homer6/frequent-cron
- Additions:
- anacron
- at/atq/atrm
- systemd timers
- inotify

Anacron

- /etc/anacrontab

```
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
HOME=/root
LOGNAME=root

# These replace cron's entries
1       5       cron.daily      run-parts --report /etc/cron.daily
7       10      cron.weekly     run-parts --report /etc/cron.weekly
@monthly        15      cron.monthly    run-parts --report /etc/cron.monthly
```

- Job-description lines are of one of these two forms:
  - `period delay job-identifier command`
  - `@period_name delay job-identify command`
- The period is specified in days
- The delay is specified in minutes.
  The job-identifier can contain any non-blank character, except slashes. It is used to identify the job in Anacron messages, and as the name for the job's timestamp file.
- The command can be any shell command. The fields can be separated by blank spaces or tabs. The period_name can only be set to monthly at the present time. This will ensure jobs are only run once a month, no matter the number of days in this month, or the previous month.

at

- sudo apt install at
- ```
  Usage: at [-V] [-q x] [-f file] [-u username] [-mMlbv] timespec ...
         at [-V] [-q x] [-f file] [-u username] [-mMlbv] -t time
         at -c job ...
         at [-V] -l [-o timeformat] [job …]
         atq [-V] [-q x] [-o timeformat] [job …]
         at [ -rd ] job …
         atrm [-V] job …
         batch

  timeformat: -t [[CC]YY]MMDDhhmm[.ss]
  ```
  ```

at schedulers

```
echo 'date > /tmp/example' | at now +2 minute

at -f /path/to/script.sh

'bash /path/to/script.sh' | at  now +1 minute
warning: commands will be executed using /bin/sh
job 1 at Wed Jan  3 01:11:00 2024
```

- `echo 'echo' | at 20:00`
- `echo 'echo' | at 20:00 tomorrow`
- `echo 'echo' | at 20:00 2024-01-03 # Explicit is better than implicit.`
- `echo 'echo' | at now +10 minute  # один з найкорисніших варіантів використання IMHO`
- `echo 'echo' | at tomorrow +10 minute`
- `echo 'echo' | at friday    # (в такий же час найбличої п'ятниці)`
- `echo 'echo' | at 10:00 friday     # (в 10:00 найбличої п'ятниці)`

at -q queue

- Usage: at [-V] **[-q x]** [-f file] [-u username] [-mMlbv] timespec ...
  at [-V] **[-q x]** [-f file] [-u username] [-mMlbv] -t time
  atq [-V] **[-q x]** [-o timeformat] [job …]

- atq -q a
  8 Fri Jan 5 01:26:00 2024 **a** oleg
  10 Fri Mar 1 20:00:00 2024 **a** oleg
  7 Thu Jan 4 01:35:00 2024 **a** oleg
- at -c 7 – sh-код для виконання

- Джоби в /var/spool/cron/atjobs

systemd.timer

```
# example.timer
[Unit]
Description=Example Timer

[Timer]
OnCalendar=*-*-* *:*:00
AccuracySec=1m
Persistent=true

[Install]
WantedBy=timers.target
```

```
# example.service
[Unit]
Description=Example Service
Wants=example.timer

[Service]
ExecStart=/path/to/script
```

https://www.freedesktop.org/software/systemd/man/latest/systemd.timer.html
https://www.freedesktop.org/software/systemd/man/latest/systemd.time.html

systemd.timer

```
OnCalendar=*-*-* *:*:00

Mon *-*-* 12:00:00
*-*-02 10:00:00
*-*-01..07 10:00:00
```

```
    minutely → *-*-* *:*:00
      hourly → *-*-* *:00:00
       daily → *-*-* 00:00:00
     monthly → *-*-01 00:00:00
      weekly → Mon *-*-* 00:00:00
      yearly → *-01-01 00:00:00
   quarterly → *-01,04,07,10-01 00:00:00
semiannually → *-01,07-01 00:00:00
```

https://www.freedesktop.org/software/systemd/man/latest/systemd.time.html

systemd timer non-normalized timers

```
OnCalendar=

                      Wed *-1 → Wed *-*-01 00:00:00
            Wed..Wed,Wed *-1 → Wed *-*-01 00:00:00
                  Wed, 17:48 → Wed *-*-* 17:48:00
Wed..Sat,Tue 12-10-15 1:2:3 → Tue..Sat 2012-10-15 01:02:03
              03-05 08:05:40 → *-03-05 08:05:40
                    08:05:40 → *-*-* 08:05:40
                       05:40 → *-*-* 05:40:00
                  *-*-7 0:0:0 → *-*-07 00:00:00
```

systemd.timer transient ad-hoc

- `systemd-run --on-active=30 /path/to/script.sh`
- `systemd-run --on-calendar='2024-03-01 10:00:00' /tmp/timer.sh`
- `systemd-run --on-calendar="*-*-* *:*:00/10" /tmp/timer.sh`
- Перелік, інформація
  ```
  systemctl list-timers
  systemctl list-timers run-\*
  systemctl cat run-….
  ```
- відміна
  ```
  systemctl disable run-….
  ```

incron/inotify

- sudo apt-get install incron
- man incrontab
  /etc/incron.allow
- man 5 incrontab
  incrontab -e
- <path> <mask> <command>

```
/home IN_CREATE /usr/local/bin/abcd $#
/home IN_CREATE,recursive=false /usr/local/bin/abcd $#
```

incron/inotify

```
/home IN_CREATE /usr/local/bin/abcd $#
/home IN_CREATE,recursive=false /usr/local/bin/abcd $#
```

MASK:

- IN_ACCESS File was accessed (read) (*)
- IN_ATTRIB Metadata changed (permissions, timestamps, extended attributes, etc.) (*)
- IN_CLOSE_WRITE File opened for writing was closed
- IN_CLOSE_NOWRITE File not opened for writing was closed
- IN_CREATE File/directory created in watched directory
- IN_DELETE File/directory deleted from watched directory
- IN_DELETE_SELF Watched file/directory was itself deleted
- IN_MODIFY File was modified
- IN_MOVE_SELF Watched file/directory was itself moved
- IN_MOVED_FROM File moved out of watched directory
- IN_MOVED_TO File moved into watched directory (*)
- IN_OPEN File was opened (*)

ARGUMENTS MACROS:

- $@ watched filesystem path
- $# event-related file name
- $% event flags (textually)
- $& event flags (numerically)
- $$ dollar sign