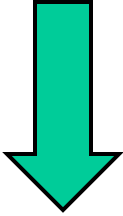


Periodic Processes

Periodically proceed the assigned jobs in BSD...
Similar the one in windows.

CRON – Schedule Commands (1)

- Thinking about programming an alarm,
or a reminding agent. (threads + sleep + time check is required)
- ❑ What we want?
 - Do things at right time automatically
 - ❑ cron daemon
 - The daemon that handles periodic execution
 - cron daemon reads configuration file and executes commands on time
- 

cron(8), crontab(1), crontab(5)

CRON – Schedule Commands (2)

❑ Configuration file

- So called: **crontab** (cron table)
- Location of user cron configuration file
 - Every user can have at most one crontab file and this file will be named the user's login ID **Be ran using the specified uid.**

System	Cron Dir
FreeBSD	/var/cron/tabs
Red Hat	/var/spool/cron
Solaris	/var/spool/cron/crontabs
SunOS	/var/spool/cron/crontabs

- Location of System Cron Configuration file
 - /etc/crontab

CRON – Schedule Commands (3)

❑ Configuration File Format

1. Ignored
 - Blank lines or leading spaces and tabs
 2. Comments
 - pound-sign Lines whose first non-space character is a #
 3. environment setting
 - name = value
 - LOGNAME: when & what & where
 - SHELL → default: sh
 - Default environment variables:
 - LOGNAME, SHELL, PATH, HOME, MAILTO
 4. cron command
- Format:

```
# minute hour day month weekday command
```

```
33 7 * * * /usr/local/bin/rsync -al --delete /home/ backup:/raid/home/
```

CRON – Schedule Commands (4)

☐ cron command format – *minute hour day month weekday command*

Field	Description	Range
minute	Minute of the hour	0 ~ 59
hour	Hour of the day	0 ~ 23
day	Day of the month	1 ~ 31
month	Month of the year	1 ~ 12
weekday	Day of the week	0 ~ 6 (0 = Sunday)

☐ Rule Matching

ALL

- * matches everything
- Single character matches exactly
- Dash(-) matches range
- Comma(,) matches any listed value
- Slash(/) matches skips of the number's value through the range.

CRON – Schedule Commands (5)

❑ crontab time format example

45 10 * * 1-5

→ AM 10:45, from Mon. to Fri.

10 * * * *

→ On 10 minutes of each hour

*/3 * * * *

→ Every three minutes

periodic

30 15 5 * *

→ PM 3:30 of each 5-th day

e.g. $1-59/2 = 1, 3, 5, 7, 9, \dots, 59$

0 0 14 2 *

→ On the Midnight of Valentine's day

5 0-6 * * *

→ On 5 minutes, from 0 to 6 o'clock.

0,30 * 13 * 5

→ every half-hour on Fri. and every half-hour on the 13-th day

❑ crontab example

20 1 * * *

`find /tmp -atime +3 -exec rm -f {} ‘;`

55 23 * * 0-3,6

`/home/chwong/cputemp-check.sh`

CRON – Schedule Commands (6)

❑ Special strings to specify the time

string	meaning	in 5 fields format
@reboot	Run once, <u>at startup</u> .	N/A
@yearly	Run once a year	0 0 1 1 *
@annually	(same as @yearly)	
@monthly	Run once a month	0 0 1 * *
@weekly	Run once a week	0 0 * * 0
@daily	Run once a day	0 0 * * *
@midnight	(same as @daily)	
@hourly	Run once an hour	0 * * * *

e.g. @yearly [command]

crontab command

❑ crontab(1)

% crontab -e [-u user]

- Edit the [user's] crontab using editor

% crontab -l

- List the content of the crontab

% crontab -r

- Remove the current crontab

% crontab *filename*

- Install *filename* as your crontab

crontab management

❑ To Allow or deny user from using cron daemon

- By default, all users can have their own crontab
- allow file
 - A list of users that may use crontab, any other not in the list can not use it
- deny file
 - Reverse meaning

❑ log

System	Allow or deny file	Log file
FreeBSD	/var/cron/{allow,deny}	By syslogd
Red Hat	/etc/cron.{allow,deny}	/var/log/cron
Solaris	/etc/cron.d/cron.{allow,deny}	/var/cron/log
SunOS	/var/spool/cron/cron.{allow,deny}	By syslogd

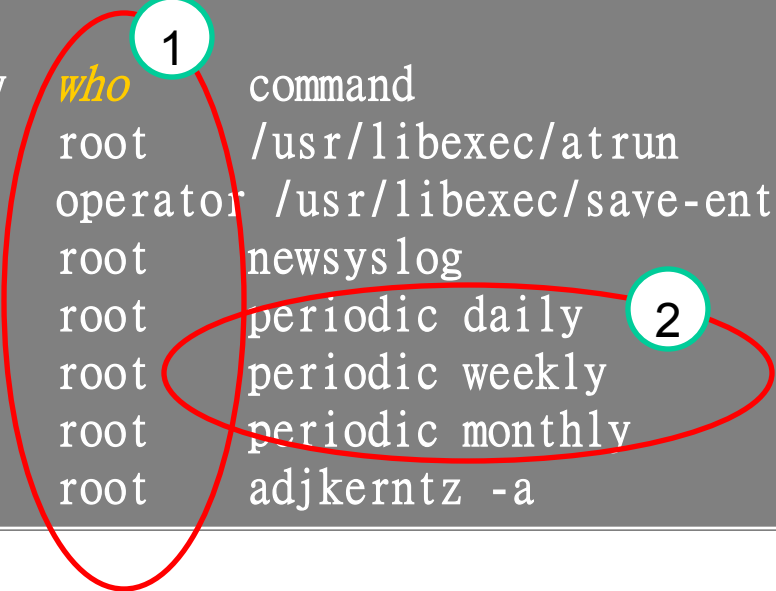
※ Default – all enabled

System crontab: /etc/crontab

❑ System crontab

- /etc/crontab

```
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
HOME=/var/log
#minute hour mday month wday who command
*/5 * * * * root /usr/libexec/atrun
*/11 * * * * operator /usr/libexec/save-entropy
0 * * * * root newsyslog
1 3 * * * root periodic daily
15 4 * * 6 root periodic weekly
30 5 1 * * root periodic monthly
1,31 0-5 * * * root adjkerntz -a
```



periodic utility

□ periodic utility

Run all the scripts under the directory.

- Run periodic system function under `/etc/periodic`

```
sabsd [/home/chwong] -chwong- ls -ld /etc/periodic/*
drwxr-xr-x  2 root  wheel  1024 Sep 26 21:43 /etc/periodic/daily
drwxr-xr-x  2 root  wheel   512 Sep 27 03:49 /etc/periodic/monthly
drwxr-xr-x  2 root  wheel   512 Sep 27 03:49 /etc/periodic/security
drwxr-xr-x  2 root  wheel   512 Sep 27 03:49 /etc/periodic/weekly
```

```
sabsd [/home/chwong] -chwong- ls /etc/periodic/daily
100.clean-disks          200.backup-passwd      405.status-ata-raid     430.status-rwho
110.clean-tmps           210.backup-aliases     406.status-gmirror     440.status-mailq
120.clean-preserve       300.calendar           407.status-graid3      450.status-security
130.clean-msgs           310.accounting         408.status-gstripe     470.status-named
140.clean-rwho           330.news                409.status-gconcat     500.queuerun
150.clean-hoststat       400.status-disks       420.status-network     999.local
```

- `/etc/periodic.conf`
- `/etc/defaults/periodic.conf`
- `/usr/local/etc/periodic`

e.g. daily checks.. (daily, security)

e.g. send mails to root

Details on what programs in the dir. should be run.

e.g. `100.clean-tmps_enable="NO"`

```
derek[/usr/local/etc/periodic/security] -chiahung- ls -al
total 8
drwxr-xr-x  2 root  wheel   512 Jan  7  2009 ./
drwxr-xr-x  4 root  wheel   512 Apr  7  2010 ../
-r-xr-xr-x  1 root  wheel  2286 Jan  7  2009 410.portaudit*
```

at command

~~/usr/libexec/atrun~~ in the /etc/crontab

❑ at command

By maintaining a queue... {time, cmd.}

- executes commands at a specified time

at [-q queue] [-f file] [-mldbv] time **Than keyin the cmd., then ^D**

or at [-q queue] [-f file] [-mldbv] -t [[CC]YY]MMDDhhmm[.SS]

❑ at management

Time: timestamp, -t to specify readable time

- atq, atrm
- /var/at/at.{allow,deny}

❖ **Default -- all disabled; set empty deny file to allow all users.**

Question: can “at command” be used to run a program within minute? second?

at(1), atrun(8)