

# Periodic Processes

---

lctseng / Liang-Chi Tseng

Edit: yench

# Run commands in a specific period?

---

- ❑ Using a shell script

```
1 #!/bin/sh
2
3 while true ; do
4     echo "Hello We!"
5     sleep 3600
6 done
7
```

- ❑ Use the built-in command: crontab

```
@hourly    echo "Hello We!"
```

# CRON – Schedule Commands (1)

---

## ❑ What do 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

## ❑ Reference

- cron(8) – the daemon
- crontab(1) – the command
- crontab(5) – the /etc/crontab table file

# First impression of crontab

## ❑ Example configuration of crontab

```
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
HOME=/var/log
MAILTO=admin@nasa.cs.nctu.edu.tw
#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
```

# 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
  - Edit using crontab(1) command

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
  - Default environment variables:
    - LOGNAME, SHELL, PATH, HOME, MAILTO

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

## 4. cron command

Format:

```
# minute hour day month weekday command
33 7 * * * /bin/date >> /tmp/log
```

# 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 ~ 7 (0,7 = Sunday)

❑ Rule Matching

- \* 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   |
| 30 15 5 * *   | → PM 3:30 of each 5-th day  |
| 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<br>half-hour on the 13-th day |

## ❑ crontab example

- |                 |  |
|-----------------|--|
| 20 1 * * *      | find /tmp -atime +3 -exec rm -f {} ';' |
| 55 23 * * 0-3,6 | /home/lctseng/cputemp-check.sh         |



# CRON – Schedule Commands (6)

## ❑ Special strings to specify the time

string	meaning	in 5 fields format
@reboot	Run once, at startup.	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 * * * *
@every_minute	Run once a minute	*/1 * * * *
@every_second	Run once a second	N/A

# crontab command

---

## ❑ crontab(1)

% crontab -e [-u user]

- Edit the [user's] crontab using editor
- Only privileged user (root) can use “-u” option

% 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

- 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

# System crontab: /etc/crontab

## ❑ System crontab

- /etc/crontab

```
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
HOME=/var/log
MAILTO=admin@nasa.cs.nctu.edu.tw
#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
```

# The implementation of crontab(1)

---

❑ Get a look at the source code from Minix

- <http://www.cise.ufl.edu/~cop4600/cgi-bin/lxr/http/source.cgi/commands/simple/cron.c>

❑ How to become a daemon?

```
if ((status = fork()) < 0) {
    fprintf(stderr, "cron: cannot fork!\n");
    exit(1);
} else if (status > 0) exit(0);

/* We are now the child. Ignore ALL signals. */
for (i = 1; i <= _NSIG; i++) signal(i, SIG_IGN);
```

# The implementation of crontab(2)

---

❑ Get a look at the source code from Minix

- <http://www.cise.ufl.edu/~cop4600/cgi-bin/lxr/http/source.cgi/commands/simple/cron.c>

❑ How to do tasks periodically?

```
/* The endless loop. */
while (1) {
    /* Sleep till the next full minute */
    (void) time(&clk);
    sleep((unsigned) (60 - clk % 60));
}
```

# The implementation of crontab(3)

## ❑ Get a look at the source code from Minix

- <http://www.cise.ufl.edu/~cop4600/cgi-bin/lxr/http/source.cgi/commands/simple/cron.c>

## ❑ How to do tasks periodically?

```
/* Wakeup from the sleep(), and check for any work. */
void wakeup()
{
    register struct tm *tm;
    time_t cur_time;
    CRON *this_entry;
    int st, pid;

    this_entry = head;
    load_crontab();

    time(&cur_time);
    tm = localtime(&cur_time);

    while (this_entry->next && this_entry->mn) {
        if (match(this_entry->mn, tm->tm_min) &&
            match(this_entry->hr, tm->tm_hour) &&
            match(this_entry->day, tm->tm_mday) &&
            match(this_entry->mon, tm->tm_mon + 1) &&
            match(this_entry->wkd, tm->tm_wday)) {
```

# periodic utility (1)

## □ periodic utility

- Run periodic system function under /etc/periodic

```
21:26 lctseng@nasa(140.113.17.225)[~]
[^^] > ls -ld /etc/periodic/*
drwxr-xr-x  2 root  wheel  29 11 12  2014 /etc/periodic/daily/
drwxr-xr-x  2 root  wheel   5 11 12  2014 /etc/periodic/monthly/
drwxr-xr-x  2 root  wheel  17 11 12  2014 /etc/periodic/security/
drwxr-xr-x  2 root  wheel   8 11 12  2014 /etc/periodic/weekly/
```

```
21:40 lctseng@nasa(140.113.17.225)[~]
[^^] > ls /etc/periodic/daily
100.clean-disks*      210.backup-aliases*  406.status-gmirror*  450.status-security*
110.clean-tmps*      300.calendar*        407.status-graid3*   460.status-mail-rejects*
120.clean-preserve*  310.accounting*      408.status-gstripe*  480.status-ntpd*
130.clean-msgs*      330.news*            409.status-gconcat*  500.queuerun*
140.clean-rwho*      400.status-disks*    420.status-network*  800.scrub-zfs*
150.clean-hoststat*  401.status-graid*    430.status-rwho*     999.local*
200.backup-passwd*  404.status-zfs*      440.status-mailq*
```

- /etc/periodic.conf
- /etc/defaults/periodic.conf



# periodic utility (2)

## □ periodic utility

- For custom system programs: /usr/local/etc/periodic

```
21:54 lctseng@bsd5(140.113.235.135)[/usr/local/etc/periodic]
[^^] > ls -al
total 12
drwxr-xr-x  7 root  wheel  7  7 22 16:29 ./
drwxr-xr-x 38 root  wheel 99 10 25 17:28 ../
drwxr-xr-x  2 root  wheel  8 10 25 17:27 daily/
drwxr-xr-x  2 root  wheel  9  7 22 16:29 hourly/
drwxr-xr-x  2 root  wheel  6  7 22 16:29 minutely/
drwxr-xr-x  2 root  wheel  4 10 25 17:27 security/
drwxr-xr-x  2 root  wheel  3 10 25 17:27 weekly/
```

```
21:55 lctseng@bsd5(140.113.235.135)[/usr/local/etc/periodic/daily]
[^^] > ls -al
total 17
drwxr-xr-x  2 root  wheel    8 10 25 17:27 ./
drwxr-xr-x  7 root  wheel    7  7 22 16:29 ../
-r-xr-xr-x  1 root  wheel  480  3 13  2015 100.updateCsPorts*
-r-xr-xr-x  1 root  wheel  939  3 13  2015 110.generateDescribe*
-r-xr-xr-x  1 root  wheel 1028  3 13  2015 120.updatePorts*
-r-xr-xr-x  1 root  wheel 2746 10 21 03:04 411.pkg-backup*
-r-xr-xr-x  1 root  wheel 2506 10 21 03:04 490.status-pkg-changes*
-r-xr-xr-x  1 root  wheel 2065  8  7 09:16 smart*
```

## periodic utility (3)

---

- ❑ Execution order depends on filenames
  - Use number as prefix to control the order
  
- ❑ All scripts under that directory will be executed
  - Unlike /etc/rc.conf
  - Even though there is no “YES” in /etc/periodic.conf
  
- ❑ /etc/periodic.conf

```
6 daily_clean_tmps_enable="YES"
7 daily_clean_tmps_verbose="NO"
8 daily_status_security_inline="YES"
9 daily_status_security_pkgaudit_quiet="NO"
10
11 monthly_statistics_enable="YES"
12 monthly_statistics_report_devices="YES"
13 monthly_statistics_report_ports="YES"
14 monthly_status_security_inline="YES"
15
16 weekly_noid_enable="YES"
17 weekly_status_pkg_enable="YES"
18 weekly_status_security_inline="YES"
```

# periodic utility (4)

## □ Driven by crontab

```
1 # /etc/crontab - root's crontab for FreeBSD
2 #
3 # $FreeBSD: releng/10.1/etc/crontab 194170 2009-06-14 06:37:19Z brian $
4 #
5 SHELL=/bin/sh
6 PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
7 #
8 #minute hour    mday    month    wday    who    command
9 #
10 */5      *      *      *      *      root    /usr/libexec/atrun
11 #
12 # Save some entropy so that /dev/random can re-seed on boot.
13 */11     *      *      *      *      operator /usr/libexec/save-entropy
14 #
15 # Rotate log files every hour, if necessary.
16 0        *      *      *      *      root    newsyslog
17 #
18 # Perform daily/weekly/monthly maintenance.
19 1        3      *      *      *      root    periodic daily
20 15       4      *      *      6      root    periodic weekly
21 30       5      1      *      *      root    periodic monthly
22 #
23 # Adjust the time zone if the CMOS clock keeps local time, as opposed to
24 # UTC time. See adjkerntz(8) for details.
25 1,31    0-5    *      *      *      root    adjkerntz -a
```

# at command (1)

---

## ❑ at command

- executes commands at a specified time (one-time use)

at [-q queue] [-f file] [-mldbv] time

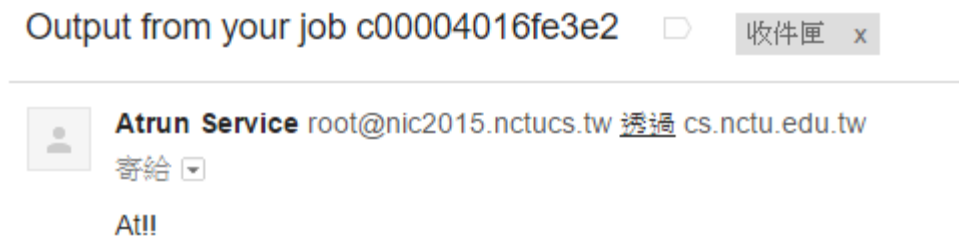
or at [-q queue] [-f file] [-mldbv] -t [[CC]YY]MMDDhhmm[.SS]

## ❑ at management

- atq
  - View job queue
- atrm
  - Remove jobs
- /var/at/at.{allow,deny}
  - Specify who can/cannot use at
  - By default, only root can use “at”

# at command (2)

- ❑ Output will send via email



- ❑ Driven by crontab (invoked every 5 minutes)

```
7 #
8 #minute hour      mday      month      wday      who      command
9 #
10 */5 * * * * root /usr/libexec/atrun
11 #
```

- ❑ Reference

- at(1)
- atrun(8)