

<http://ja.wikipedia.org/wiki/Crontab>

<http://miya0.dyndns.org/pc/settei/crontab.html>

<http://www.express.nec.co.jp/linux/distributions/knowledge/system/crond.html>

<https://qiita.com/aki-takano/items/3a3c5fca3c185173eda1>

### cron の設定ファイル

cron の設定ファイルは、以下の表の通りです。crond は、毎分、以下の設定ファイルの内容に変更がないかを確認し、変更があった場合には、それを反映して実行します。

ファイル / ディレクトリ名	利用者	主な用途
/var/spool/cron/user	全ユーザ	ユーザの自動タスク設定ファイル
/etc/crontab	root	毎時、毎日、毎月、毎週の自動タスクのメイン設定ファイル
/etc/cron.hourly	root	毎時実行される自動タスク設定ファイルを置くディレクトリ
/etc/cron.daily	root	毎日実行される自動タスク設定ファイルを置くディレクトリ
/etc/cron.monthly	root	毎月実行される自動タスク設定ファイルを置くディレクトリ
/etc/cron.weekly	root	毎週実行される自動タスク設定ファイルを置くディレクトリ
/etc/cron.d	root	上記以外の自動タスク設定ファイルを置くディレクトリ

### crontab コマンドの書式

crontab ファイルは、cron を操作する際に作成し、プロセスを定期的に行うためのファイルで、crontab コマンドで作成します。書式は、以下の通りです。

```
# crontab [-u user] {-l|-r|-e}
```

なお、指定可能なオプションについては、下の表を参照して下さい。

オプション	--
-e	crontab を対話的に編集します。通常は、vi エディタが起動して、設定ファイルの編集を行います。
-l	crontab ファイルの内容を表示します。
-r	crontab ファイルを削除します。
-u user	user で指定したユーザの crontab ファイルを操作の対象とします。なお、このオプションは、root ユーザのみ使用できます。

## 書き方

```
# (行頭の # マークはコメント行を示す)
# +-----分 (0 - 59)
# | +-----時 (0 - 23)
# | | +-----日 (1 - 31)
# | | | +-----月 (1 - 12)
# | | | | +-----曜日 (0 - 6) (日曜日 =0)
# | | | | |
# * * * * * 実行されるコマンド
```

### 例 1

```
毎日 3 時にシェル実行
00 03 * * * /mnt/sdb/backup/backup.sh
```

### 例 2

```
43 23 * * *          23:43 に実行
12 05 * * *          05:12 に実行
0 17 * * *           17:00 に実行
0 17 * * 1           毎週月曜の 17:00 に実行
0,10 17 * * 0,2,3   毎週日, 火, 水曜の 17:00 と 17:10 に実行
0-10 17 1 * *       毎月 1 日の 17:00 から 17:10 まで 1 分毎に実行
0 0 1,15 * 1        毎月 1 日と 15 日と 月曜日の 0:00 に実行
42 4 1 * * *        毎月 1 日の 4:42 分に実行
0 21 * * 1-6        月曜日から土曜まで 21:00 に実行
0,10,20,30,40,50 * * * * 10 分おきに実行
*/10 * * * * *      10 分おきに実行
* 1 * * * *         1:00 から 1:59 まで 1 分おきに実行
0 1 * * * *         1:00 に実行
0 */1 * * * *       毎時 0 分に 1 時間おきに実行
0 * * * * *         毎時 0 分に 1 時間おきに実行
2 8-20/3 * * * *   8:02,11:02,14:02,17:02,20:02 に実行
30 5 1,15 * * *    1 日と 15 日の 5:30 に実行
```

## @ を使った時刻指定

### 例

```
@reboot /path/to/script
```

記述	意味
@reboot	Run once, at startup.
@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 * * * *'.

## GUI で設定する

```
gnome-schedule
```

をインストールすることで、GUI で設定できる。

## cron で届くメールを止める

cron で実行された標準出力やエラーは cron ファイルの所有者にメールで届く。これを止めるには2つの方法がある。

### 方法 1

標準出力、標準エラーを /dev/null へ送る

```
hoge.sh >/dev/null 2>&1
```

### 方法 2

設定ファイルで、メールの宛先を空にする

```
MAILTO=""  
0 17 * * * hoge.sh
```

## 文字が化ける

cron で実行すると LANG 環境変数が設定されずにログファイル等が化けることがある。  
cron から実行されるシェル等の冒頭で

```
export LANG=ja_JP.UTF-8
```

で文字コードを設定しておくが良い。