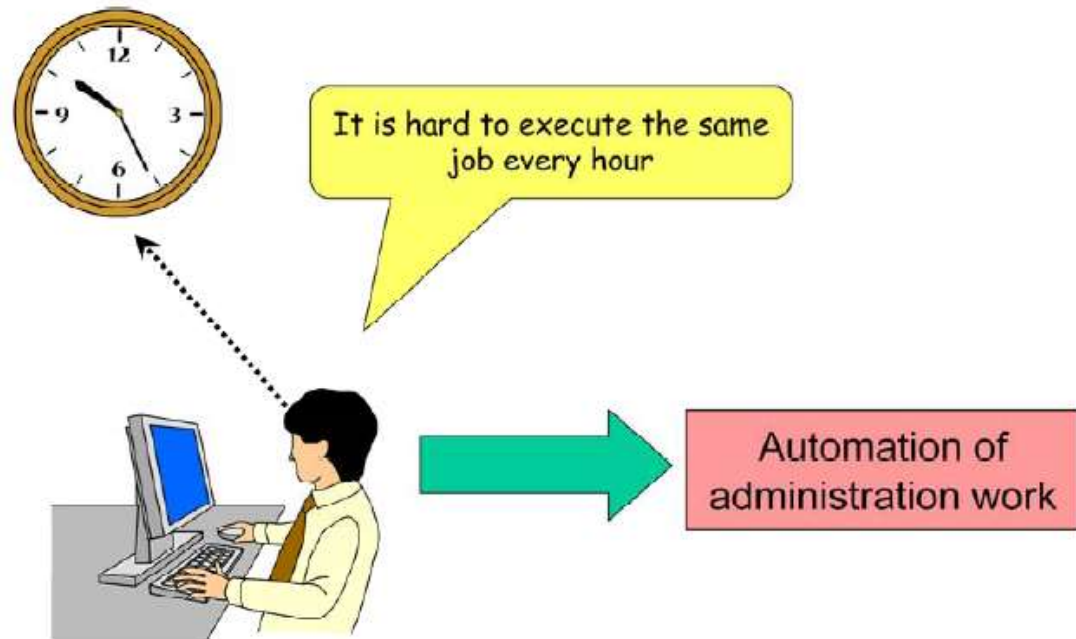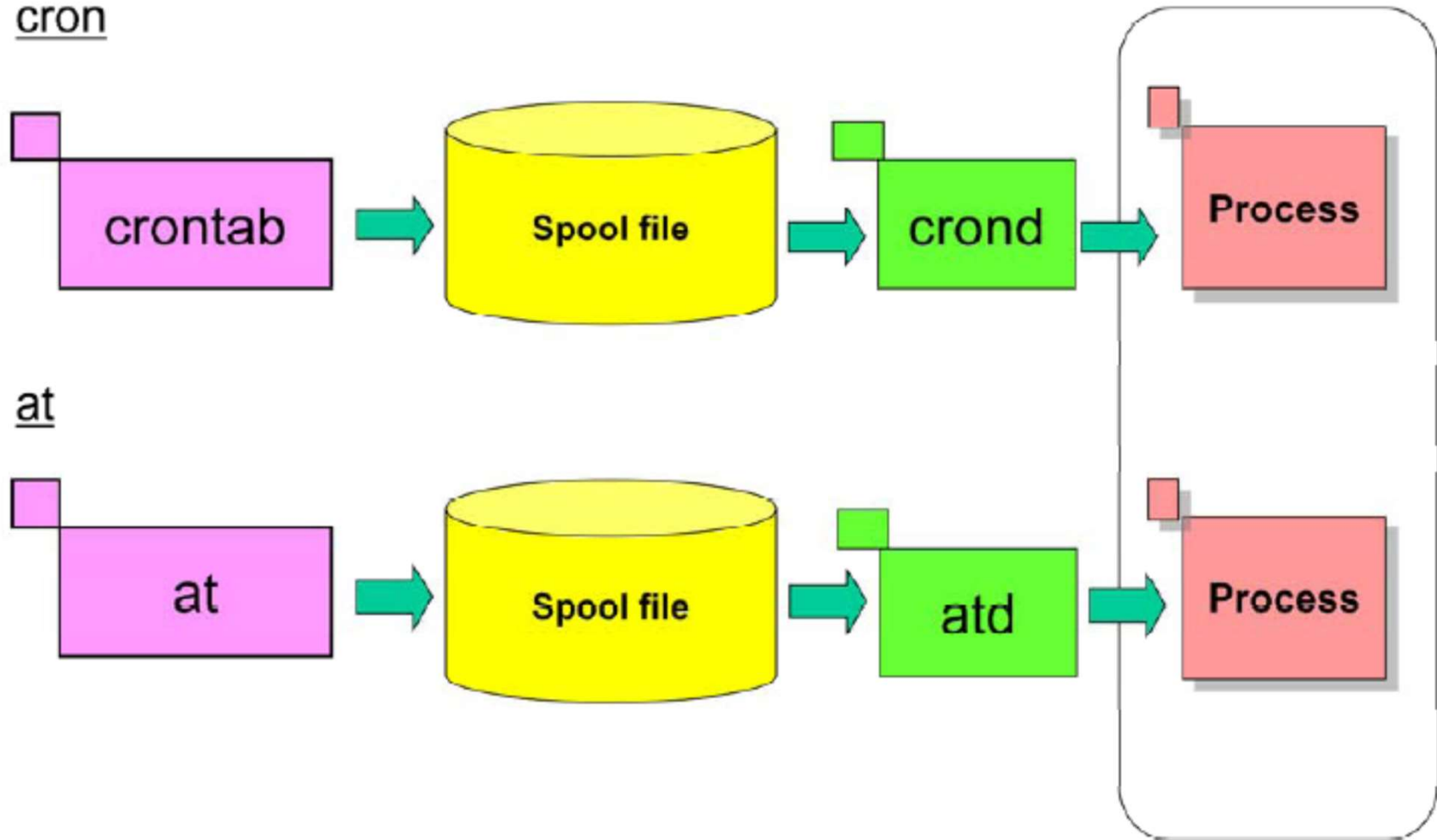# Job scheduling

Why scheduling?

- Administration need to do jobs regularly
  - Checking logs
  - Regularly backup
  - Check accounts
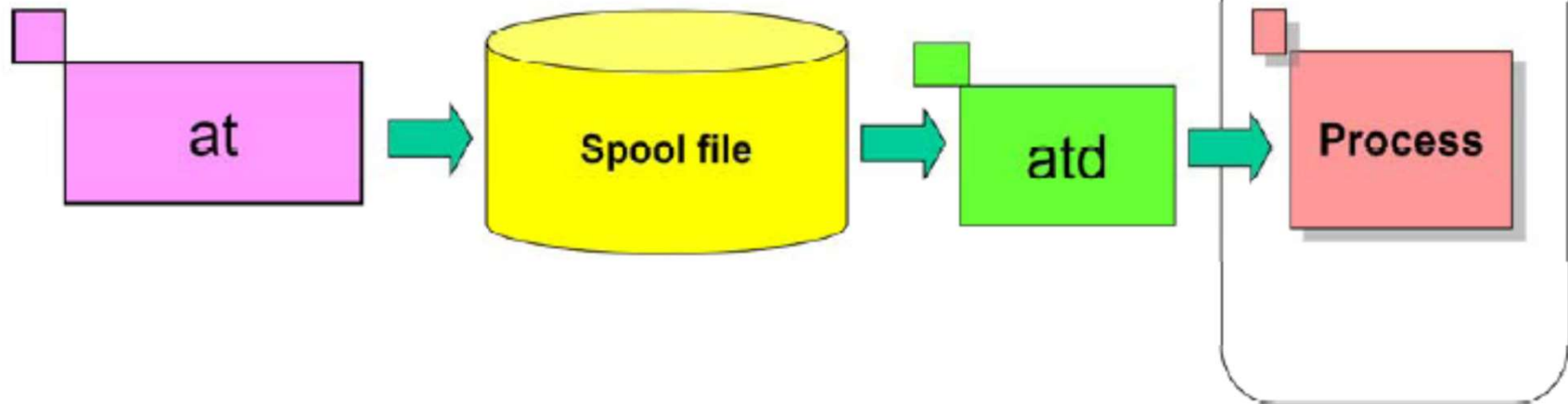 - Check security vulnerability
- Many tasks
- Repetitive tasks

It is hard to execute the same job every hour

Automation of administration work

Automate tasks

cron

crontab → Spool file → crond → Process

at

at → Spool file → atd → Process

cron – configuration file of 'cron'
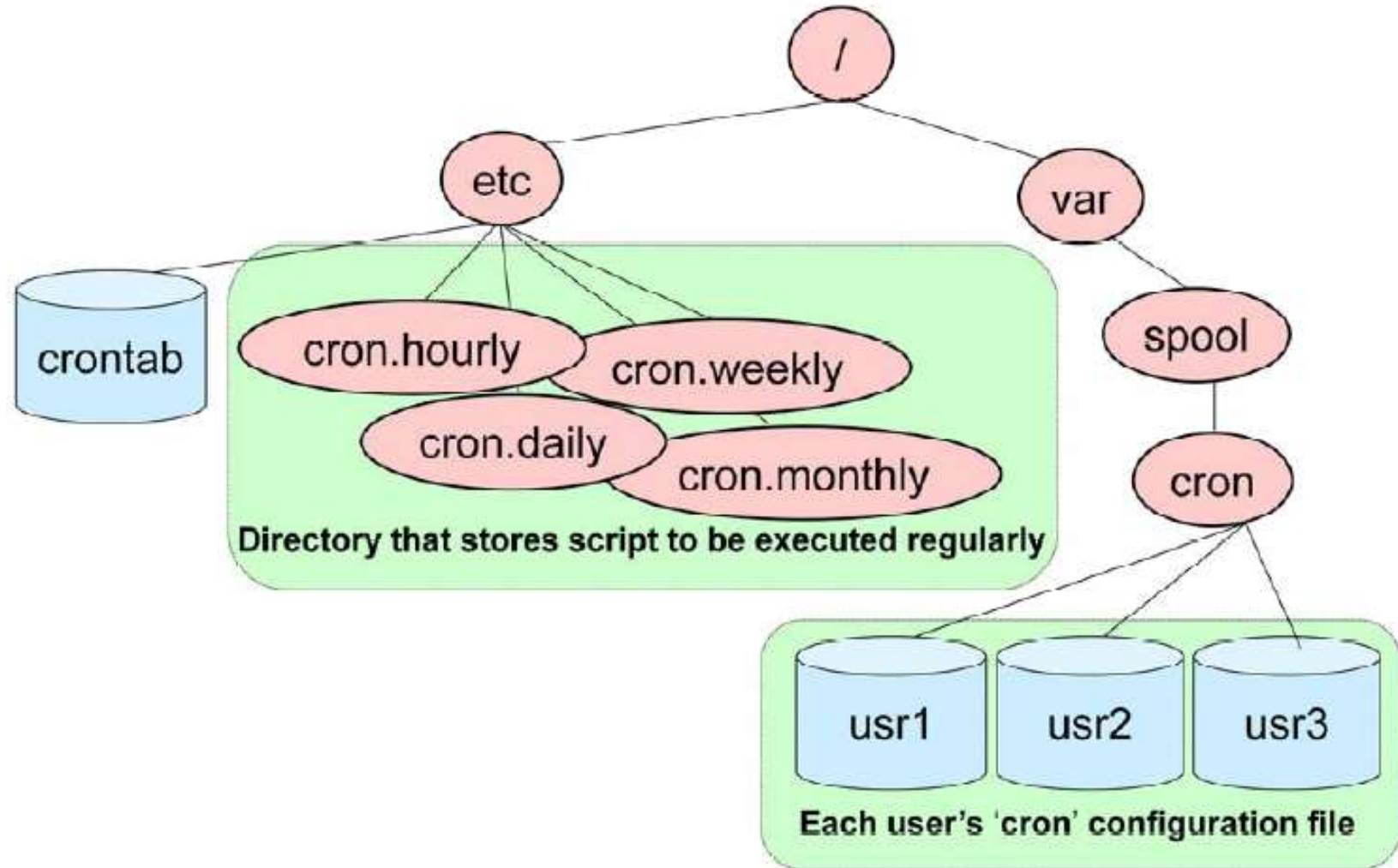

Directory that stores script to be executed regularly

Each user's 'cron' configuration file

cron

- **cron** is a job scheduler on Unix-like operating systems. It executes tasks in configuration files
  - /etc/crontab
  - /etc/cron.hourly
  - /etc/cron.daily :
  - /etc/cron.weekly :
  - /etc/cron.monthly
  - /var/spool/cron

crontab

- It can modify the content of the crontab file of each user (spool file).
- Administrators can manage crontab file of each user

  crontab [option] [user_name]

- **Options:**

  -e : Creation and modification of the 'crontab' file

  -r : Remove the 'crontab' file

  -l : Display the 'crontab' file

Crond parameters

```
#   crontab -e
```

```
0      9-17    *    *    *    /bin/date > /dev/console
minute  hour    day  month day of              command
                          the week

(1)     (2)    (3)  (4)  (5)                   (6)
```

When the output destination is not specified, it is answered by mail.

Các trường trong tệp crontab

	Field	Meaning
(1)	Minute	0 - 59
(2)	Hour	0 - 23
(3)	Day of month	1 - 31 (1st - 31th)
(4)	Month	1 - 12 (January - December)
(5)	Day of week	(0: Sunday - 6: Saturday)
(6)	Command	Command to execute at a specified time

Regulation

- (1) * : Any available values.
- (2) value1- value2: value range
- (3) value1, value2: list values
- (4) range/increment: within the range and the increment
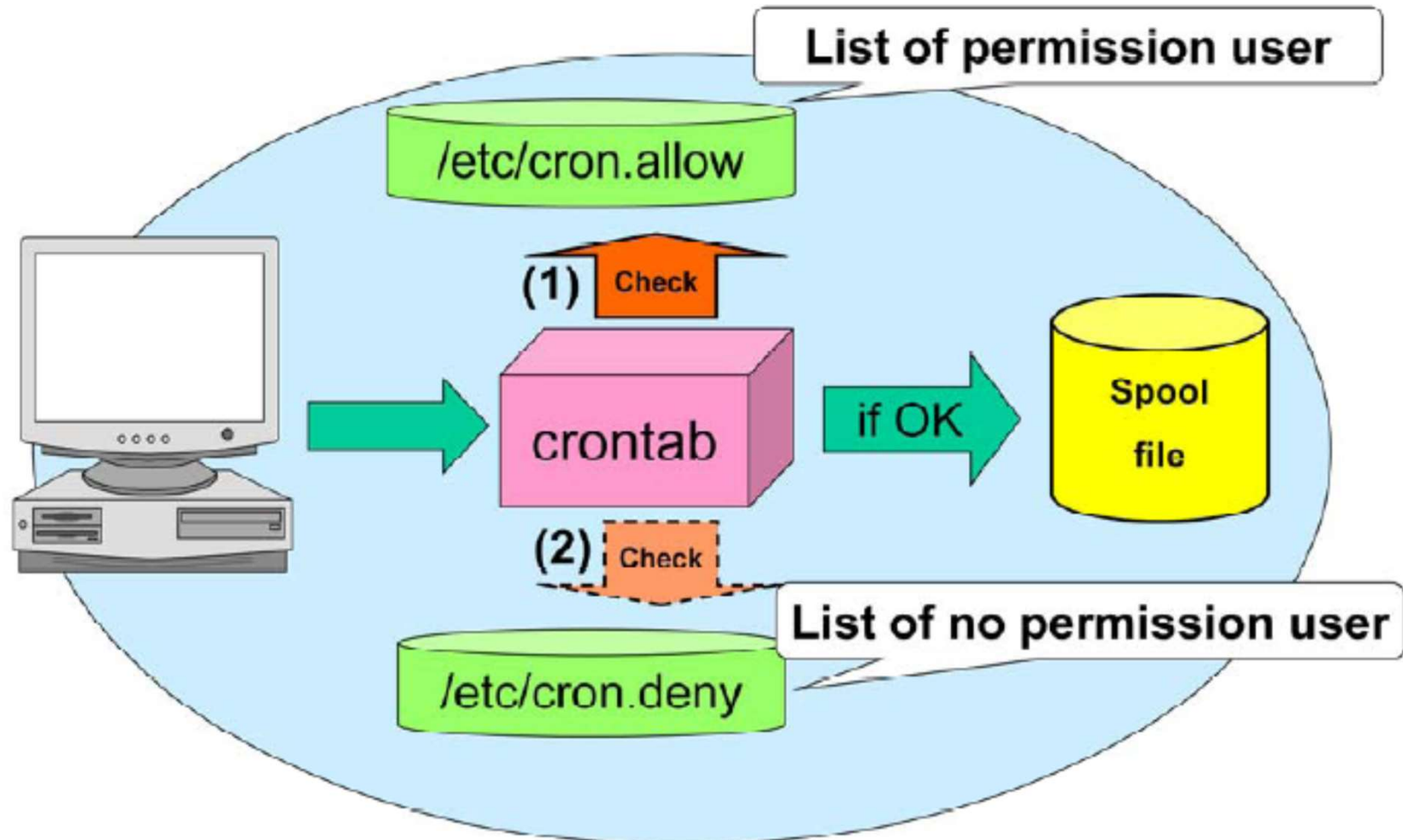
Show and delete crond file

(1) Display

```
# crontab -l
10 * * * *       /usr/local/bin/clean.sh    ................(1)
5 2 */4 * *      /usr/local/bin/backup.sh    ..............(2)
0 1 * * 1,3,5    /usr/local/bin/snap.sh      .............(3)
#
```

(2) Deletie

```
# crontab -r
#
```

Limit crond using

Command at

(1) Register

```
# at 22:40
at> /home/usr1/progA > /dev/null
at> <EOT>
job 5 at 2004-08-30 22:40
#
```

Ctrl + d

(2) Display

```
# atq
5   2004-08-30 22:40 a root
#
```

(3) Delete

```
# atq
5   2004-08-30 22:40 a root
# atrm 5
#
```

Command at

- (1) at is used to register an automatical task

at [-q queue] [-f file] [-m] TIME

- **Parameters:**

  -q : queue with different tasks/ jobs with different
       priorities

  -f : Read the command from a file

  -m : Notify the result by email

Command at

- (2) Use **atq** to check the task queue registered by at.

  atq [-q queue] [-v]

- (3) Use '**atrm**' to remove a job registered by **at**

  atrm job [job...]