

---

# 시스템 관리/자동화를 위한 크론과 예약작업 1

---

시스템관리 자동화와 크론(cron)

## 시스템 작업 자동화

---

Cron	지정된 시간에 정기적으로 실행하는 예약 작업
Anacron	비동기적으로 사용하여 특정 일에 반복하는 예약 작업
at	특정 시간에 한번 실행하는 예약 작업
batch	시스템 부하가 떨어지는 시기에 한번 실행하는 batch 작업
system Unit File	시스템 파일을 사용하여 다음 부팅 시 실행할 작업 예약

## 크론작업

---

시스템에서 주기적이고 반복적인 작업을 예약해두고서 원하는 시간에 자동 실행하도록 설정하는 작업

- crond 명령은 crontab 파일로 /var/spool/cron/ 디렉토리 안에 있는 각 사용자의 crontab 파일을 메모리에 싣는다.
- 또한, crond 명령은 시스템 전체의 crontab 파일인 /etc/crontab 파일을 찾아, 그것을 메모리에 싣는다.
- 다음, crond 명령은 매 분마다 저장된 crontab 내용들을 확인하고, 그에 맞는 작업을 수행한다.
- 프로그램이 실행될 때, 필요에 따라 그 결과를 MAILTO 환경 변수에 지정된 사용자에게 편지로 보내기도 한다.

- 추가로, crond 명령은 cron 스푼 디렉토리와 /etc/crontab 파일이 바뀌었는지도 검사한다. 그래서 바뀌었다면 그대로 적용한다.
- /etc/crontab 의 형식은 다음과 같다.

분	시	일	월	요일	사용자	실행명령
---	---	---	---	----	-----	------

- 분 : 0~59
- 시 : 0~23
- 일 : 1~31
- 월 : 1~12 또는 jan, feb, mar, apr ...
- 요일 : 0~6 (0=0 또는 7) 또는 월, 화, 수, 목, 금, 토
- 사용자 : 명령을 실행할 사용자
- 실행명령 : 그 시간에 실행할 명령

- [사용 예]

```
00 05 1 * * root cp -r /home /backup
```

→ 여기서 '\*'은 매월, 모든 요일을 의미한다.

→ 해석 : 매월 1 일 새벽 5 시 root 사용자권한으로 'cp -r /home /backup' 명령 수행(요일(\*))은 상관하지 않음)

#### 1. 시스템크론

시스템에 기본으로 설정되어 있는 크론으로서 시스템 관리자는 설정된 크론을 사용만 하면 된다. 최고 관리자(root)에게만 사용이 허용된다.

#### 2. 사용자크론

최고 관리자와 일반사용자들이 crontab 명령어를 사용하여 개별적으로 설정할 수 있는 크론설정을 의미한다. 이 설정 내용은 /var/spool/cron 디렉토리에 계정명과 동일한 파일명으로 보관되어 있다.

## crond 데몬 관리

시스템 크론과 사용자 크론은 크론데몬(crond)이 실행되어야 정상작동이 된다.

cron 작업을 예약하기 전에 :

#### 1. cronie 패키지 설치

```
# yum install conie
```

2. `crond` 설치 시 - 부팅 시 자동으로 시작하게 - 서비스를 사용할 수 있다.  
서비스 사용을 중지 한 경우 사용하도록 다음과 같이 설정

```
# systemctl enable crond.service
```

3. `crond` 서비스 시작

```
# systemctl start crond.service
```

4. `crond` 서비스 상태 확인

```
# systemctl status crond.service
```

## crontab 명령

---

- `crontab -l`  
개별 사용자의 cron 설정 내용 확인. `/var/spool/cron/` 디렉토리 밑에 동일 사용자계정명과 동일 파일을 불러와서 보여준다.
- `crontab -e`  
cron 설정 명령어로 `/var/spool/cron/` 사용자계정명과 동일 파일을 생성 또는 불러와서 파일의 내용을 `vi` 로 열어준다.

## 시스템크론 설정파일

---

- 시스템크론은 `/etc/crontab` 파일의 설정을 읽어 들어서 그 내용이 설정되어있는 대로 실행한다.
- `/etc/crontab` 파일은 시스템 크론데몬인 `crond` 에 의해 실행될 시스템관리에 꼭 필요한 주기적인 자동실행내용들이 저장되어 있는 파일이다.

```
# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# 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
```

## 사용자 크론의 사용허가 또는 제한하는 방법

---

/etc/cron.allow 파일과 /etc/cron.deny 파일

- /etc/cron.allow 파일
  - : 크론 설정 허용 사용자 리스트, 기본으로 없음
- /etc/cron.deny 파일
  - : 크론 설정 불허용 사용자 리스트
- /etc/cron.deny 파일만 존재하면 모든 사용자 cron 설정 가능
- 두 파일이 모두 없을 때에는 root 사용자만 cron 설정 가능

## /etc/cron.deny 파일

---

현재 시스템에 있는 /etc/cron.deny 파일에 사용자 계정을 추가하여 사용자크론에 대해 제한할 수 있다.

1. cron user 생성

```
# useradd cron
```

2. cron user 로 로그인하여 cron 설정

```
# su - cron
```

```
$ crontab -e          → cron 설정 방법으로 뒤에 설명
```

```
00 01 * * * su - cron '/home/cron/backup.sh' → 입력 후 저장
```

```
$ crontab -l          → 자기계정의 cron 설정을 확인하는 명령
```

```
00 01 * * * su - cron '/home/cron/backup.sh'
```

```
$ exit
```

3. root 계정으로 /etc/cron.deny 파일에 cron 계정 추가

```
# cat >> /etc/cron.deny
```

cron → 추가

**# cat /etc/cron.deny**

cron

4. cron user 로 로그인하여 cron 설정

**# su - cron**

**\$ crontab -e**

You (cron) are not allowed to use this program (crontab)

See crontab(1) for more information

**\$ exit**

5. cron user 제한 해제

**# vi /etc/cron.deny**

cron → 삭제 후 저장

## [/etc/cron.allow](#)

---

리눅스 시스템 초기 설정 시에 /etc/cron.allow 파일은 존재하지 않는다.

1. cron 계정으로 cron 설정이 되는 지 확인

**# su - cron**

**\$ crontab -e** → 실행이 되지만 확인해보고 빠져나온다.

**\$ crontab -l**

001 \* \* \* su - cron '/home/cron/backup.sh'

→ 위에서 설정한 내용이 보여진다.

**\$ exit**

2. root 계정으로 /etc/cron.allow 파일 생성

**# touch /etc/cron.allow**

**# ls -l /etc/cron.allow**

-rw-r--r-- 1 root root 0 9 월 11 19:30 /etc/cron.allow

**# cat /etc/cron.allow** → 내용 없음

3. cron 계정으로 로그인하여 cron 설정이 되는 지 확인

```
# su - cron
```

```
$ crontab -l
```

You (cron) are **not allowed** to use this program (crontab)

See crontab(1) for more informatio

```
$ crontab -e
```

You (cron) are **not allowed** to use this program (crontab)

See crontab(1) for more information

→ /etc/cron.allow 이 없을 시에는 모든 사용자의 cron 설정이 가능했지만  
/etc/cron.allow 이 있다면 이 파일 내에 cron 설정을 허용할 사용자들을 추가해  
주어야 한다.

```
$ exit
```

4. root 계정으로 /etc/cron.allow 파일에 cron 계정 추가

```
# cat >> /etc/cron.allow
```

```
cron
```

5. cron 계정으로 로그인 하여 cron 설정이 가능한지 확인

```
# su - cron
```

```
$ crontab -l
```

```
00 01 * * * su - cron '/home/cron/backup.sh'
```

```
$ exit
```

6. /etc/cron.allow 파일 삭제

```
# rm -rf /etc/cron.allow
```

## 개별사용자의 크론설정 내역 확인

---

/var/spool/cron/ 디렉토리에는 root 와 개별사용자들의 cron 설정이 계정명과 동일한 파일명으로 보관되어 있다.

☆ 확인

```
# ls -l /var/spool/cron/
```

```
합계 11
```

```
-rw----- 1 cron root 45  9 월 11 19:15 cron
```

```
# cat /var/spool/cron/cron
```

```
00 1 * * * su - cron '/home/cron/backup.sh'
```

→ 위에서 cron 계정으로 crontab 을 설정("crontab -e")하였기 때문에  
/var/spool/cron/ 디렉토리에 동일 계정명으로 설정내용이 보관 되었다.

→ 이 내용은 각 계정에서 "crontab -l" 명령으로 확인한다.

```
# su - cron
```

```
$ crontab -l
```

```
00 01 * * * su - cron '/home/cron/backup.sh'
```

## root 권한으로 사용자크론 설정하는 방법

---

시스템관리자인 root 는 시스템의 모든 cron 설정을 수정, 변경, 삭제, 확인 할 수 있다.

### root 자신의 cron 설정 방법

---

```
# crontab -e
```

```
00 01 * * * su - root /root/backup.sh
```

```
# crontab -l
```

```
00 01 * * * su - root /root/backup.sh
```

### root 권한으로 일반사용자의 크론 설정

---

[사용법] **crontab -u 계정명 -e**

```
# crontab -u cron -e
```

```
00 01 * * * su - cron '/home/cron/backup.sh' → 원래 있던 내용
```

```
00 01 2 * * su - cron '/home/cron/backup2.sh' → 새로 추가
```

```
# cat /var/spool/cron/cron
```

```
00 01 * * * su - cron '/home/cron/backup.sh'
```

```
00 1 2 * * su - cron '/home/cron/backup2.sh'
```

→ root 계정으로 cron 계정에 추가한 내용이 /var/spool/cron/cron 파일에 추가 되었다.

```
# su - cron
```

```
$ crontab -l
```

```
00 01 * * * su - cron '/home/cron/backup.sh'
```

```
00 01 2 * * su - cron '/home/cron/backup2.sh'
```

```
$ exit
```

## root 권한으로 일반계정사용자의 cron 설정 확인

---

[사용법] **crontab -u 계정명 -l**

```
# crontab -u cron -l
```

```
00 01 * * * su - cron '/home/cron/backup.sh'
```

```
00 01 2 * * su - cron '/home/cron/backup2.sh'
```

## 설정된 크론 삭제하는 방법

---

[사용법] **crontab -r**

- 시스템 관리자인 root 는 자기자신의 cron 설정을 삭제할 수 있을 뿐 아니라 일반계정사용자의 cron 설정까지도 삭제할 수 있다.
- 일반 사용자는 자기자신의 cron 설정만을 삭제할 수 있다.(자신의 계정으로 switching 후 실행한다.)

1. root 사용자로 현재 cron 설정 확인

```
# crontab -l
```

```
00 01 * * * su - root /root/backup.sh
```

```
# cat /var/spool/cron/root
```

```
00 01 * * * su - root /root/backup.sh
```

2. root 계정의 cron 설정 삭제



```
# crontab -r
# crontab -l
no crontab for root
# ls -l /var/spool/cron/
합계 11
-rw----- 1 cron root 91  9 월 11 20:03 cron
```

## root 권한으로 일반사용자의 크론설정 삭제

---

### [사용법] crontab -u 계정명 -r

1. cron 계정의 cron 설정 확인

```
# crontab -u cron -l
00 01 * * * su - cron '/home/cron/backup.sh'
00 01 2 * * su - cron '/home/cron/backup2.sh'
# cat /var/spool/cron/cron
00 01 * * * su - cron '/home/cron/backup.sh'
00 01 2 * * su - cron '/home/cron/backup2.sh'
```

2. root 사용자로 cron 계정 cron 설정 삭제

```
# crontab -u cron -r
# crontab -u cron -l
no crontab for cron
# ls -l /var/spool/cron/
합계 0
```

### <실습 1>

1. crond 데몬이 실행 중인지 확인
2. root 계정으로 cron 을 설정한다.

조건 :

스크립트 파일 위치 : /root/backup.sh

실습을 위하여 현재 시간 이후 일정 시간 동안 매 분 마다 실행하도록 설정

3. /root/backup.sh 파일을 만들고 내용 입력

내용 : 현재 날짜를 추출해서 /root 디렉토리에 "backup-<현재날짜>.tar.bz2" 로 /etc 디렉토리 전체의 백업파일을 생성하는 의미이다.

```
#!/bin/sh

set $(date)
fname="backup-$2$3$5.tar.bz2"

tar cfj /root/$fname /etc
```

4. crond 데몬 재 시작
5. 지정한 시간에 /root 디렉토리에 파일이 생성되는지 확인
6. cron 설정 삭제

### <실습 2>

1. cron 계정으로 cron 설정이 되는지 확인
2. root 계정으로 /etc/cron.allow 파일과 /etc/cron.deny 파일 삭제
3. cron 계정으로 cron 설정 시도
4. cron 계정으로 cron 설정이 되도록 설정