



제 3장 우분투 관리

✓ 3-1 파일 시스템

✓ 3-3 사용자관리

✓ 3-5 네트워크 관리

✓ 3-2 패키지 관리

✓ 3-4 반복 작업 자동화하기



한국기술교육대학교



컴퓨터공학부 박진우 책임기술연구원 2405호



반복 작업 자동화하기

- **cron**(크론)은 시스템에게 원하는 시간에 지정한 명령을 반복적으로 시키기 위한 데몬이다.
- 서버를 관리하다 보면 매일 또는 주, 월 단위로 반복해서 해야할 일(주로 백업)이 생기게 된다. 또한, 로그 파일을 회전시켜서 파일 시스템이 가득 차지 않도록 만들기, 자료 백업하기, 시스템 시각을 동기화하기, 시스템의 최적화를 위하여 주기적으로 운영 중에 만들어진 필요 없는 임시 파일을 정리하거나, 파일 목록 데이터를 갱신하는 등의 작업을 통해 시스템이 최적의 상태를 유지할 수 있게 할 수 있다. 이러한 일을 cron 데몬을 사용하여 자동으로 수행하게 한다.



반복 작업 자동화하기

● cron 설정

◆ crontab 파일 위치 및 조회

❖ 작업 설정 파일을 crontab 파일이라고 부르며, 우분투의 경우에는 이 파일의 위치는 /var/spool/cron/crontabs/ID에 저장된다.

❖ crontab 명령을 통해 반복 작업 설정과 조회를 한다. crontab 설정 파일을 새롭게 편집하고자 한다면 ☐ 옵션을 주고, 수정하기 전에 현재 등록된 내용을 보고자 한다면 ☐ 옵션을, 등록된 내용을 삭제하려면 ☐ 옵션을 사용한다.

```
root@study: ~
root@study:~# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# which includes the day, time, and command to be run.
# For example
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 3 1,15 * * (/usr/bin/rdate -s time.bora.net; /sbin/hwclock -w)
0 2 * * * updatedb
root@study:~#
```



반복 작업 자동화하기

- **crontab 옵션**

- ◆ crontab [-u user] file

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

- ❖ -u user user 사용자에게 대해서 crontab 작업을 수행한다. 생략하면 crontab 명령을 실행하는 사용자에게 대해서 crontab 작업을 수행.

- ❖ -l : 현재 cron table을 보여 줌

- ❖ -r : 현재 cron table을 삭제

- ❖ -e : 현재 cron table을 편집

- 일반적으로 crontab -e 명령을 주로 사용한다.



반복 작업 자동화하기

- **crontab 사용 권한**
- 특정 사용자에게 cron의 사용을 허가할 경우에는 **/etc/cron.allow, /etc/cron.deny** 두 파일로 지정할 수 있다.
 - ◆ /etc/cron.allow 파일에 허가할 사용자만 추가해준다
 - ◆ /etc/cron.deny 파일에 허가하지 않을 사용자를 추가한다
 - ◆ **/etc/cron.allow를 먼저 참조**한다. /etc/cron.allow에 명시된 사용자는 cron job을 실행할 수 있다.
 - ◆ /etc/cron.allow에 없는 사용자는, /etc/cron.deny에도 없다면, cron job을 실행할 수 있다.
 - ◆ /etc/cron.allow, /etc/cron.deny 두 파일이 존재하지 않으면 **cron job을 수행할 수 있다.**
 - ◆ /etc/cron.deny 파일이 존재하고 아무 내용 없이 비어 있다면, 모든 사용자가 cron job을 수행할 수 있다.



반복 작업 자동화하기

- **cron table의 각 필드**

- ◆ | 분 | 시간 | 날짜 | 달 | 요일 | 사용자 | 명령 |
<분>은 0-59 <시>는 0-23 <날짜>는 0-31 <달>은 1-12
<요일>은 0-7(0과 7은 일요일, 1은 월요일...)

- **필드에 설정하는 방법.**

- ◆ * : everytime
- ◆ - : 범위 지정, 예를 들어 시간 필드에 1-3이면 1시와 3시 사이
- ◆ , : separator, 예를 들어 시간 필드에 1,3이면 1시와 3시
- ◆ / : 시간 필드 에서 23-7/2라면 23시부터 07시까지 매 2시간마다,
분 필드에서 */2라면 매 2분마다가 된다.



반복 작업 자동화하기

예

- ◆ `30 05 * * 1 ~/bin/backup.sh`
==> 매주 월요일 오전 05시 30분이 되면 ~/bin/backup.sh을 실행
- ◆ `30 2 1 * * ~/bin/backup.sh`
==> 매월 1일 새벽 2시 30분이 되면 ~/bin/backup.sh를 실행
- ◆ `20 1 * * * find /tmp -atime +3 -exec rm -f () \;`
==> 매일 새벽 01시 20분이 되면 3일간 접근하지 않은 /tmp 디렉터리 내의 파일을 삭제
- ◆ `0 1 1 * * (/usr/bin/rdate -s time.bora.net; /sbin/hwclock -w)`
==> 매월 1일 새벽 01시이 되면 time.bora.net의 타임서버와 동기화를 한다. 그리고 하드웨어 시간(hwclock)을 맞춘다.



반복 작업 자동화하기



- **at**

종종 작업을 주기적이 아니라 단지 한번만 실행하고 싶을 경우가 있다. 이때는 at 명령을 사용한다.

- at 명령의 옵션으로는

- ◆ -f 옵션으로 실행할 파일을 지정하거나,
- ◆ -f를 사용하지 않을 경우 표준 출력에서 실행할 명령을 입력해 준다.
- ◆ -m 옵션은 명령에서 표준 출력으로 결과가 나오지 않을 경우에도 사용자에게 실행 결과를 메일로 보낸다.
- ◆ -v 옵션은 작업을 읽기 앞서 어떤 작업을 수행할지 시각을 표시한다. 시각은 또한 출력 과정에서 표시된다.



반복 작업 자동화하기

- at 명령어에 사용하는 시각 값

```
root@study: ~  
root@study:~# at -f my-at-test.sh 10pm tomorrow  
warning: commands will be executed using /bin/sh  
job 10 at Fri Dec 9 22:00:00 2016  
root@study:~#  
root@study:~# at -f my-at-test.sh 2:00 tuesday  
warning: commands will be executed using /bin/sh  
job 11 at Tue Dec 13 02:00:00 2016  
root@study:~#  
root@study:~# at -f my-at-test.sh 3:00 july 11  
warning: commands will be executed using /bin/sh  
job 12 at Tue Jul 11 03:00:00 2017  
root@study:~#  
root@study:~# at -f my-at-test.sh 4:00 next week  
warning: commands will be executed using /bin/sh  
job 13 at Thu Dec 15 04:00:00 2016  
root@study:~#  
root@study:~# at -f my-at-test.sh now + 1 hours  
warning: commands will be executed using /bin/sh  
job 14 at Thu Dec 8 02:32:00 2016  
root@study:~#
```

- ◆ at -f my-at-test.sh 10pm tomorrow ==> 내일 오후 10시에 작업
- ◆ at -f my-at-test.sh 2:00 tuesday ==> 화요일 오전 2시에 작업
- ◆ at -f my-at-test.sh 3:00 july 11 ==> 7월 11일 오전 3시에 작업
- ◆ at -f my-at-test.sh 4:00 next week ==> 일주일후 오전 4시에 작업
- ◆ at -f my-at-test.sh now + 1 hours ==> 1시간후에 작업



반복 작업 자동화하기



• anacron

◆ cron 명령은 24시간 동작하는 시스템에 적합하다. **노트북이나 데스크탑 컴퓨터와 같이 자주 꺼졌다 켜졌다는 반복하는 시스템에는 anacron** ("anachronistic cron")이라는 또 다른 명령이 cron 명령이 일일, 주간, 월간으로 수행하는 일반적인 작업 일정을 관리한다. anacron은 시간별 작업을 다루지는 않는다.

◆ anacron은 /var/spool/anacron에 작업이 시작할 때를 기록한 타임스탬프 파일을 유지한다. anacron이 동작할 때, 작업이 마지막으로 수행해 흘려간 날짜를 살펴 필요하다면 작업을 수행한다.

anacron용 작업 테이블은 /etc/anacrontab에 저장되어 있는데, /etc/crontab과는 형식이 조금 다르다.

/etc/crontab과 마찬가지로 /etc/anacrontab은 환경 설정을 포함할 수 있다.

각 작업마다 다음과 같이 4개의 필드(최근실행일, 지연시간, 구분자, 명령어)가 있다.

최근실행일	지연시간(분)	구분자	명령어
1	65	cron.daily	run-parts /etc/cron.daily
7	70	cron.weekly	run-parts /etc/cron.weekly
30	75	cron.monthly	run-parts /etc/cron.monthly



반복 작업 자동화하기

최근실행일	지연시간(분)	구분자	명령어
1	65	cron.daily	run-parts /etc/cron.daily
7	70	cron.weekly	run-parts /etc/cron.weekly
30	75	cron.monthly	run-parts /etc/cron.monthly

- 최근실행일은 날짜로 세지만, 한달에 한번 작업을 실행할 경우에는 그 달에 속한 날짜에 무관하게 @monthly라고 명시할 수 있다.
- 지연은 작업이 끝나고 대기해야 하는 기간을 분 단위로 나타낸다. 시스템이 처음 시작할 때 엄청나게 많은 작업이 동시에 경쟁하지 못하도록 지연을 활용한다.
- 작업 구분자는 슬래시를 제외한 공백이 아닌 문자를 포함할 수 있다.