**crontab 사용법**

# crontab -l       현재사용하고 있는 crontab 파일 출력  
# crontab -e       crontab 파일의 수정(edit)

crontab -e 로 수정이 안될경우  
1. #csh        2. #setenv EDITOR vi 명령 수행후 수정함  
root crontab file

/etc/cron.d/logchecker   
  : /var/cron/log가 최대 화일 크기를 초과하는지를 판단하고, 만약  
    초과하면 내용을 /var/cron/olog로 복사한다.   
/usr/lib/newslog   
  : syslog daemon에 의해서 /var/adm 디렉토리에 생성되는 log file을 청소한다.

Controlling crontab Access   
cron utility를 제어하는 두개의 file   
/etc/cron.d/cron.allow   
/etc/cron.d/cron.deny

User Access   
- cron.allow file이 존재하면, 그 file에 적혀있는 user만이   
crontab 명령어를 사용할 수있다.   
- 이 file이 없으면 crontab은 cron.deny file을 체크해서 user가 crontab 실행을 금지할 지를 결정한다.   
- 만약 둘다 없으면 crontab은 root만 실행시킬 수 있다.   
cf) daemon, bin, smtp, nuucp, listen, nobody, noaccess file은 crontab 사용할 수 없다.   
- cron.allow file은 user가 만들어 주어야한다.   
- crontab 명령어는 crontab file을 수정, 삭제, 편집할 수있다.   
- crontab 명령어는 user에대한 입력을 /var/ spool/cron/crontabs/username에 저장한다.   
- root 만이 다른 user의 crontab file을 읽고 편집할 수있다.   
- root의 crontab file은 /var/spool/cron/crontab/root 이다.

crontab file format

# cd /var/spool/cron/crontabs 에 가신후  
   # more root  
   10 3 \* \* 0,4 /etc/cron.d/logchecker  
   10 3 \* \* 0   /usr/lib/newsyslog  
   15 3 \* \* 0 /usr/lib/fs/nfs/nfsfind  
   1 2 \* \* \* [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1  
   ----------   ========================  
   1 2 3 4 5           program

   1 : minute (0-59),  
   2 : hour (0-23),  
   3 : day of the month (1-31),  
   4 : month of the year (1-12),  
   5 : day of the week (0-6 with 0=Sunday).

   이렇게 되고 실행할 파일명을 program 부분에 적어 주면 된다.

   파일을 맞게 변경후 cron 데몬을 리스타트시키면 된다.  
   아니면..리부팅

   예로  
   \* \* 1 \* \* /usr/local/program

   이렇게 한다면 한달에 한번 매월 1일에만 program 을 실행.

   지금 root 파일을 변경했는데 root 파일이 아닌 각계정의 파일을 만들어 사용하면 각유저마다 자동으로 실행할수 있다.       
   
   
///////////////////////////////////////////////////////////////////////////////////////  
   
    
crontab 옵션  
crontab [ -u user ] file   
crontab [ -u user ] { -l | -r | -e }   
-u user : user 사용자에 대해서 crontab 작업을 수행한다. 생략하면 crontab 명령을 실행하는 사용자에 대해서 crtontab 작업을 수행.  
-l : 현재 cron table을 보여 줌  
-r : 현재 cron table을 삭제  
-e : 현재 cron table을 편집  
일반적으로  crontab -e 명령을 주로 사용한다.

crontab 사용 권한 : /etc/cron.allow, /etc/cron.deny   
특정 사용자에게만 cron의 사용을 허가할 경우  
/etc/cron.allow 파일에 허가할 사용자만 추가해준다   
/etc/cron.deny 파일에 허가하지 않을 사용자를 추가한다  
/etc/cron.allow를 먼저 참조한다.   
/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 슈퍼 유저만 cron job을 수행할 수 있다.  
/etc/cron.deny 파일이 존재하고 아무 내용 없이 비어 있다면, 모든 사용자가 cron job을 수행할 수 있다.

cron table의 각 필드  
+-------------------------------------------------------------------+  
| 분 | 시간 | 날짜 | 달 | 요일 | 사용자 | 명령 |  
+-------------------------------------------------------------------+  
minute(0~59) hour(0-23) day(1-31) month(1-12) weekday(1-7) command   
<분>은 0-59  <시>는 0-23  <날짜>는 0-31  <달>은 0-12(0또는 12는 12월, 1은 1월...) <요일>은 0-7(0과 7은 일요일, 1은 월요일...)  
''\*''  :  everytime  
''-''  :  범위 지정, 예를 들어 시간 필드에 1-3이면 1시와 3시 사이  
'',''  :  separator, 예를 들어 시간 필드에 1,3이면 1시와 3시  
/   :  시간필드 에서 23-7/2라면 23시부터 07시까지 매 2시간마다라는 의미  
      분필드에서 \*/2라면 매 2분마다라고 해석하면 된다.

사용 예  
30  05 \*  \*  1    ~/bin/backup.sh  
              ==> 매주 월요일  오전 05시 30분에 ~/bin/backup.sh을 실행  
30  2  1  \*  \*    ~/bin/backup.sh  
              ==> 매월 1일 새벽 2시 30분에 ~/bin/backup.sh를 실행  
20  1  \*  \*  \*    find /tmp -atime +3 -exec rm -f () '';''  
              ==> 매일 새벽 01시 20분에 3일간 접근하지 않은 /tmp내의 파일을 삭제  
~

<http://lupusmaru.egloos.com/1090823>

[편집하기](http://www.springnote.com/session/new?return_to=http%3A%2F%2Fpcguy7.springnote.com%2Fpages%2F872610%3Fedit%3D1)

1) 현재 cron deamon이 돌고 있는지 확인  
ps -ef | grep cron

2) cron deamon kill  
kill -9 "pid of cron"

3) deamon 재실행  
/usr/sbin/cron

4) 명령어 위치  
/usr/bin/crontab

5) 사용형식  
crontab [ -u 사용자ID ] 파일  
crontab [ -u 사용자ID ] { -l | -r | -e }

1. 개요  
cron(크론)은 원하는 시간에 명령(프로그램)을 시키기 위한 데몬이다.  
서버는 늘 깨어있다는 것을 이용한 최대한의 활용법이 될 수 있다.

    - 내가 새벽 3시에 서버에 특정 작업을 해줘야하는데 그 때 깨어있을 수 있는가?  
    - 또는 30분 간격으로 HDD의 사용량을 운영자에게 알리도록 해야한다면?  
    - 매월 초에 자료를 백업 받고 싶다면?

바로 이럴 때 cron은 최고의 해결책을 제시한다.  
cron은 항상 지정한 시간이 되었는지 확인을 하여 해당 명령어을 실행하는 것이다.

시스템의 최적화의 기법은 일정 시간 단위로 최대한의 성능을 발휘하도록 도와주는 것이다. 즉, 주기적으로 운영 중에 만들어진 필요하지 않는 임시 파일을 정리하거나, 운영체제 운영에 필요한 데이터를 갱신하는 등의 작업을 통해 리눅스가 최적의 상태를 유지할 수 있게 할 수 있다.

2. cron 설정  
1) crontab 파일 위치 및 조회  
작업 설정 파일을 crontab 파일이라고 부르며, 이 파일의 위치는 OS별로 차이가 있다.  
리눅스는 /var/spool/cron/ID, 솔라리스는 /var/spool/corn/crontabs/ID에 위치한다.  
그럼 이 파일을 직접 수정해야 하는가? 그렇지 않다. crontab 명령을 통해 설정과 조회를 한다.

crontab 설정 파일을 새롭게 편집하고자 한다면 -e 옵션을 주고, 수정하기 전에 현재 등록된 내용을 보고자 한다면 -l 옵션을 사용하며, 등록된 내용을 삭제하려면 -d 옵션을 사용한다

설정 내용을 조회해 보자(-l 옵션).  
$ crontab -l  
no crontab for truefeel  
설정한 적이 없어 아직 비어있다.

* /etc/crontab 파일  
  - /etc/rc.d/init.d/crond 스크립트  
  - /var/spool/cron 디렉토리 내의 크론 설정 파일들

2) crontab 파일 형식  
----------    ----------  ---------------------------------------------------  
  필  드                 의  미           범  위  
----------    ----------  ---------------------------------------------------  
첫 번째                      분              0-59  
두 번째                      시              0-23  
세 번째                      일              0-31  
네 번째                      월              1-12  
다섯 번째                  요일             0-7 (0 또는 7=일요일, 1=월, 2=화, ...)  
여섯 번째                명령어             실행할 명령을 한줄로 쓴다.  
----------    ----------  ---------------------------------------------------

- 모든 엔트리 필드는 공백으로 구분된다.  
- 한 줄당 하나의 명령 (두줄로 나눠서 표시할 수 없음)  
- # 으로 시작하는 줄은 실행하지 않는다.

설정을 해보자. (-e 옵션)  
crontab -e을 하면 vi 에디터가 나온다(환경변수 EDITOR에 따라 다른 에디터를 사용할 수 있다).  
   
$ crontab -e  
# /home 디렉토리를 /BACKUP/home으로 백업해둠  
#  
# 30분, 새벽 4시와 낮 12시, 모든 일, 모든 월, 모든 요일  
30 4,12 \* \* \*  /usr/bin/rsync -avxH --delete /home /BACKUP/home > /dev/null 2>&1  
#  
# 파일/디렉토리 퍼미션 설정  
# 40분, 새벽 1시, 매주 일요일  
40 1 \* \* 0  /root/bin/perm\_set.sh   > /dev/null 2>&1

위는 매일 4:30분과 12:30분에 rsync 명령을, 매주 일요일 1:40분에 perm\_set.sh를 실행함을 의미한다.

3) 설정 예

시간 설정에서 몇가지 의미있는 것들을 알아보자.  
- '\*'표시는 해당 필드의 모든 시간을 의미한다.  
- 3,5,7와 같이 콤마(,)로 구분하여 여러 시간대를 지정할 수 있다.  
- 2-10와 같이 하이픈(-)으로 시간 범위도 지정할 수 있다.  
- 2-10/3와 같이 하이픈(-)으로 시간 범위를 슬래쉬(/)로 시간 간격을 지정할 수 있다(2~10시까지 3시간 간격으로. 즉, 3, 6, 9시를 의미함).

원하는 시간                                  형  식  
매주 토요일 새벽 2:20                     20  2 \*  \*  6  명령어  
매일 오후 4,5,6시                         0  4-6   \*  \*  \*  명령어  
매일 2시간 간격으로 5분대에               5  \*/2 \*  \*  \* 명령어  
매월 1일 새벽 1:15                        15  1   1  \*  \*  명령어  
1,7월 1일 새벽 0:30                       30  0   1  1,7  \*  명령어

4) /etc/crontab 파일로 설정  
매시 1회 자동실행하기 위한 시스템 크론 설정

01 \* \* \* \* root run-parts /etc/cron.hourly  
  - 매일 매시 01분마다 /etc/cron.hourly 디렉토리내에 존재하는 파일들을 실행

매일 1회 자동실행하기 위한 시스템 크론설정

02 4 \* \* \* root run-parts /etc/cron.daily  
 - 매일 새벽 4시 02분마다 /etc/cron.daily  디렉토리내에 존재하는 파일들을 실행

매주 1회 자동실행하기 위한 시스템 크론설정

22 4 \* \* 0 root run-parts /etc/cron.weekly  
 - 매주 일요일 새벽 4시 22분마다 /etc/cron.weekly 디렉토리내에 존재하는 파일들을 실행

매월 1회 자동실행하기 위한 시스템 크론설정

42 4 1 \* \* root run-parts /etc/cron.monthly  
 ->매월 1일 새벽 4시 42분마다 /etc/cron.monthly 디렉토리내에 존재하는 파일들을 실행

\* root 이외의 사용자에게 crontab 명령어를 이용할 수 있게 하는 방법

  - /etc/cron.allow 파일에 사용자의 id를 등록

* 일반사용자의 crontab 명령어사용을 제안하고자 한다면  
    - /etc/cron.deny 파일에 사용자의 id 를 등록

3. FAQ

1) cron 설정한 후에는 crond 데몬을 재실행해야 하나요?

    아닙니다. crontab -e 으로 설정 후 빠져나오면 바로 적용됩니다.

2) truefeel 사용자는 cron을 못 쓰게 하고 싶습니다.  
    /etc/cron.allow : 허용할 사용자 ID 목록  
    /etc/cron.deny  : 거부할 사용자 ID 목록  
    cron.allow 파일이 있으면 이 파일에 들어있는 ID만 사용 가능  
    cron.deny  파일이 있으면 이 파일에 들어있는 ID는 사용 불가  
    따라서 cron.deny에 truefeel ID를 추가해주면 됩니다.

3) > /dev/null  2>&1 이 무슨 뜻입니까?  
    지정한 명령어 처리 결과와 발생할지 모르는 에러메시지를 출력하지 않고 모두 버린다는(/dev/null)는  
    뜻입니다. 만약 결과와 에러를 파일로 저장하려면 /dev/null 대신 파일명을 적어주면 됩니다.

<http://pcguy7.springnote.com/pages/872610>