CRON

2020 SEMINAR
ALOGON

CRON 이란

✓ Unix 기반 OS 에서 미리 *작업을 백그라운드에서 수행하도록 예약할 수 있는 *Daemon

✓ ex) 주기적으로 메일 서버에서 메일을 받아오는 작업, 주기적으로 자료 백업

✓ 시간을 뜻하는 그리스어 chronos 에서 기원

*작업 : 여기서 뜻하는 작업이란 쉘에서 할 수 있는 모든 작업을 뜻함

*Daemon : 사용자가 직접적으로 제어하지 않고, 백그라운드에서 돌면서 여러 작업을 하는 프로그램. ‘d’를 이름 끝에 달고 있음, service, systemctl 으로 제어

CRON 시작하기

✓ Cron 설치 : apt install cron (or apt-get install cron)

✓ Cron 시작 : service cron start (or /etc/init.d/cron start)

✓ Cron 종료 : service cron stop (or /etc/init.d/cron stop)

✓ Cron 재시작 : service cron restart (or /etc/init.d/cron restart)

✓ Cron 실행 확인 : ps aux | grep cron

CRONTAB 이란

✓ 예약된 작업들의 리스트를 저장하고있는 Table

✓ Cron은 Crontab (Cron Table) 으로 예약된 작업들을 관리

✓ Cron은 시작될 때 각 crontab 파일들을 읽고 sleep 상태로 변경. 그리고 매 분마다 깨어나서 변경사항과 수행해야할 작업을 확인

CRONTAB 이란

Crontab 이 저장되어 있는 경로

- /var/spool/cron : 개발 사용자를 위한 crontab <- 예제를 통해 확인해볼 예정
- /etc/cron.d : 일반적으로 패키지 설치를 위해서 추가되는 경향이 있음
- /etc/crontab : root 관리자를 위한 system crontab <- root 권한이 꼭 필요
- /etc/cron.daily, /etc/cron.hourly, /etc/cron.monthly,/etc/cron.weekly
  - 직접 sh 파일을 등록시켜 놓으면 정해진 시간에 수행됨 수행
  - 스케쥴은 이미 /etc/crontab에 저장되어 있음

CRON 환경변수

SHELL cron 이 돌아가는 shell (Default : /bin/sh)

PATH cron 프로그램 경로 (Default: /usr/bin:/bin)

MAILTO cron 실행 결과를 받아볼 메일 주소 (“” 으로 두면 메일이 오지 않는다)

=> sendmail 과 같은 메일발송 daemon이 필수

HOME cron에 쓰일 홈 디렉토리 ($HOME Default : /etc/passwd의 값)

LONGNAME crontab의 소유주

ANACRON

- crond 는 매 1분마다 crontab들의 변경사항을 확인하며 예약된 작업을 수행한다.
- 만약! 11시에 예약된 작업이 10시55분 ~ 11시 5분 사이에 있던 서버 보수 작업으로 인해서 작업이 이루어지지 않았다면? -> 수행이 당연히 안된다!
- Anacron 은 이와 같이 수행을 놓친 작업이 있다면 작업이 가능한 시점에서 놓쳐버린 작업을 수행해주는 daemon
- anacron config 는 /etc/anacrontab 에 저장됨

CRONTAB 명령어

✓ 예약 작업 작성 및 수정 : crontab –e

```
Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed
```

select-editor 이란 명령어로 editor 변경도 가능

✓ 예약 작업 삭제 : crontab –r

✓ 예약된 작업 리스트 : crontab –l

CRONTAB 형식

기본 형태 : * * * * * [command]

- *(분, 0~59) *(시, 0~23) * (일, 1~31) *(월, 1~12), *(요일, 0~7)
- 요일에서 0과 7은 일요일, 1 ~ 6은 각각 월요일 ~ 토요일을 의미

- Examples
  - * * * * * /home/script/test.sh : 매분 test.sh 실행
  - 45 5 * * 5 /home/script/test.sh : 매주 금요일 오전 5시 45분에 test.sh 실행
  - 0,20,40 * * * * /home/script/test.sh : 매일 매시간 0분, 20분, 40분에 test.sh 실행
  - 0-30 1 * * * /home/script/test.sh : 매일 1시부터 1시 30분까지 매분 test.sh 실행
  - */10 * * * * /home/script/test.sh : 매 10분마다 test.sh 를 실행
  - */10 2,3,4 5-6 * * /home/script/test.sh : 5일에서 6일까지 2시, 3시, 4시에 매 10분마다 실행

http://www.cronmaker.com/?0 사이트에서 만들 수 있음

실습 1

1분마다 date를 출력해서 .txt 파일에 넣기 (10번 이상)

Hint

1분마다 -> cron 시간표현식으로는 * * * * *

Date명령어로 날짜 및 시간 출력

Redirection명령어로 .txt 파일에 넣기

- >명령어는 파일에 쓸 때 사용
- >>명령어는 파일에 추가할 때 사용

실습 2

Cron을 통해 구구단 만들기 (단수는 원하시는 대로 선택하시면 됩니다)

Cron으로 1분마다 실행시켜 구구단 한 줄씩 .txt 파일에 추가되도록 실행

(ex) 7 x 1 = 7 => 7 x 1 = 7 => ....

7 x 2 = 14

Hint

파일이 있는지 없는지 –f $"파일이름" 으로 파악가능

곱해지는 숫자가 증가해야 함 => .txt파일의 줄 수로 파악 (cat "파일이름" | wc –l 이용)

구구단 계산을 위해서는 expr 사용 -> `expr $num1 ₩* $num2`

THANK YOU