

# Crontab

유닉스 OS 계열에서 특정 시간에 특정 작업을 해야하는 경우 사용하는 스케줄러입니다.

## 1. crontab basic

### 1.1 스케줄 설정

아래의 커멘드를 입력하면 스케줄을 설정할수 있는 vi 에디터 페이지가 생성된다. 여기에 어떤 주기로 어떤 파일을 실행할지에 대한 리스트를 작성해주면 된다.

```
$ crontab -e
```

### 1.2 스케줄 리스트 확인

현재 crontab의 스케줄을 확인할 수 있다.

```
$ crontab -l
```

## 2. 주기 설정

```
time.py
```

```
-----  
import datetime  
today = datetime.datetime.now()  
print(str(today))  
-----
```

```
* * * * *
```

분(0-59) 시간(0-23) 일(1-31) 월(1-12) 요일(0-7)

\* 요일에서 0과 7은 일요일

### 2.1 2분 간격으로 실행

```
*/2 * * * * python3 /home/ubuntu/time.py >> time.txt
```

### 2.2 매시 10분에 실행

```
10 * * * * python3 /home/ubuntu/time.py >> time.txt
```

### 2.3 매시 10분과 20분에 실행

```
10,20 * * * * python3 /home/ubuntu/time.py >> time.txt
```

### 2.4 매일 5시 10분과 20분에 실행

```
10,20 5 * * * python3 /home/ubuntu/time.py >> time.txt
```

### 2.5 일요일 5시 10분과 20분에 실행

```
10,20 5 * * 0 python3 /home/ubuntu/time.py >> time.txt
```

### 2.6 5시에서 10시까지 매시에 5분마다 time.py를 실행하고 결과를 time.txt에 저장

```
*/5 5-10 * * 0 python3 /home/ubuntu/time.py >> time.txt
```

## 3. time zone 변경

타임존 변경

---

```
# 현재 사용 시간 확인
```

```
$ timedatectl
```

```
# 사용하는 파일 심볼릭 링크확인
```

```
$ ls -l /etc/localtime
```

# 사용 할수 있는 타임존 확인

```
$ timedatectl list-timezones | grep Asia
```

# 타임존 변경

```
$ sudo timedatectl set-timezone Asia/Seoul
```

# 심볼릭 링크 수정

```
$ sudo unlink /etc/localtime
```

```
$ sudo ln -s /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

---

#### 4. vim 에디터 인코딩 변경

```
$ vi .vimrc
```

---

```
set encoding=utf-8
```

---

#### 5. crontab 로그 확인

아래의 명령으로 crontab의 시스템 로그를 확인 할수 있습니다.

```
$ grep CRON /var/log/syslog
```

#### 6. crontab 에러 확인 및 해결

에러가 발생하면 아래와 같은 에러 로그를 확인할수 있습니다.

---

```
Mar  4 07:42:01 ip-172-31-3-64 CRON[7494]: (CRON) info (No MTA installed, discarding output)
```

---

## MTA : Mail Transfer Agent

### 6.1 에러 메시지 확인하는 방법 1

- \$ sudo apt-get install postfix
- \$ cat /var/mail/ubuntu

### 6.2 에러 메시지 확인하는 방법 2

- \$ sudo apt install mailutils
- \$ mail

### 6.3 crontab에서 실행되는 python 환경

crontab 에서는 .bash\_profile이 실행되지 않기때문에 pyenv 환경이 적용되지 않습니다.

version.py

```
-----  
import sys  
print(sys.version.split(" ")[0])  
-----
```

crontab -e

```
-----  
* * * * * python /home/ubuntu/version.py >> version.txt  
-----
```

## 6.4 pyenv 환경에 있는 python으로 실행

직접 python 경로 입력

```
***** /home/ubuntu/.pyenv/versions/python3/bin/python /home/ubuntu/version.py
>> version.txt
```

PATH 설정

```
PATH=/usr/local/bin/./sbin:/bin:/usr/sbin:/home/ubuntu/.pyenv/versions/python3/bin
***** python /home/ubuntu/version.py >> version.txt
```

crontab을 이용하여 서울의 날씨정보를 슬랙으로 전송

패키지 설치

```
pip install requests
```

weather.py

```
import requests, json
```

```
webhook_URL = "https://hooks.slack.com/services/TNKE1KJR/BU56Q9V50/XmoLXF1Sc36CRGkwu6ZgZly4"
```

```
api_key = "320d5b7837c0bbe44e012d79baf0ffe7"
```

```
def weather(lat, lon):
```

```
url = "https://api.darksky.net/forecast/320d5b7837c0bbe44e012d79baf0ffe7/{},  
{ }".format(lat, lon)
```

```
response = requests.get(url)
```

```
return response.json()["daily"]["summary"]
```

```
def send_slack(msg, channel="#dss12", username="날씨봇"):
```

```
    payload = {
```

```
        "channel": channel,
```

```
        "username": username,
```

```
        "icon_emoji": ":slack:",
```

```
        "text": msg,
```

```
    }
```

```
    response = requests.post(
```

```
        webhook_URL,
```

```
        json.dumps(payload),
```

```
    )
```

```
send_slack(weather(37.5665, 126.9780))
```

---