

# Shell Script & Cron

19 Summer  
Made by loopy

# 1. 쉘 스크립트(Shell Script)

# 1. 셸 스크립트 - 셸 스크립트란?

- ✓ Shell이나 CLI에서 돌아가도록 작성되었거나 OS를 위해 쓰인 \*스크립트
- ✓ 쉽게 말해 셸이 실행할 수 있는 코드 → 셸 스크립트
- ✓ 컴파일 과정 없이 line-by-line 으로 실행되는 \*인터프리터 언어
- ✓ 이 세미나에서는 **bash**를 중심으로 설명할 예정 (셸의 종류에 따라 명령어가 달라질 수 있음)

\*스크립트 : 프로그램이나 명령어들의 나열

\*인터프리터 언어 : 컴파일을 하지 않고 소스코드를 한 줄씩 읽어 바로바로 실행하는 언어

# 1. 셸 스크립트 - 쉼뱅(Shebang)

- ✓ `#!` 으로 시작하는 코드이며, Sharp(`#`) + Bang(`!`) 의 합성어
- ✓ 스크립트의 첫 줄에 쓰며, 해당 스크립트를 어떤 인터프리터로 해석할 것인지를 정의
- ✓ 이 스크립트를 실행시켜 줄 **프로그램의 경로를 지정**하며, Shebang 이 있으면 실행할 때 인터프리터를 지정해주지 않아도 됨

예시 1 : `#!/bin/bash` (절대 경로) → 오늘 세미나에서는 이 Shebang 만 사용

예시 2 : `#!/usr/bin/env python` (환경 변수 사용)

- `[#!/usr/bin/env + 언어]` 형식으로 작성
- 환경 변수에서 지정한 언어를 찾아서 실행하므로 시스템 환경이 달라져도 사용 가능

# 1. 셸 스크립트 - 파일 권한

ls -al 명령을 실행하면 가장 왼쪽에 파일 접근 권한이 나타남

ex) drwxr-xr-- → ① d / ② rwx / ③ r-x / ④ r--

① → 파일 타입 ( - : 파일, d : 디렉토리, l : 다른 파일을 가리키는 링크, p : pipe 파일,  
b : block device, c : character device )

② → 소유주(USER) 권한, ③ → 그룹(Group) 권한, ④ → 기타 사용자(Others) 권한  
알파벳은 각각

r : 읽기 권한 (= 4), w : 쓰기 권한 (= 2), x : 실행 권한 (= 1), - : 권한 없음 (= 0)

# 1. 셸 스크립트 - 접근 권한 부여

**chmod** (change mode) 명령어를 통해 사용 권한을 지정

사용법 : `chmod [ option ] [ mode ] [ file ]`

option

- `-R` : 하위 파일과 디렉토리 모든 권한을 변경
- `-v` : 실행되고 있는 모든 파일을 나열
- `-c` : 권한이 변경된 파일 내용 출력

# 1. 셸 스크립트 - 접근 권한 부여

mode - 문자열 모드

- 사용법 : `chmod [ option ] (reference) (operator) (modes) [ file ]`
- reference
  - u : user 권한, g : group 권한, o : others 권한, a : 모든 유저의 권한
- operator
  - + : 해당 권한 추가, - : 해당 권한 제거, = : 해당 권한을 설정한대로 변경
- modes
  - r : read 권한, w : write 권한, x : execute 권한, - : 사용 권한 없음
- ex) `chmod u=rwx,g+x sample` : sample파일의 user는 읽기,쓰기,실행 권한 부여, group 멤버들에게 실행권한 추가

# 1. 셸 스크립트 - 셸 스크립트 실행

mode - 8진법 수 모드

- 사용법 : `chmod [ option ] [ 8진법 수 ] [ file ]`
- 권한의 알파벳에 해당하는 숫자를 모두 더해서 표시
  - ex) `rwxr-xr-- == 754`
  - ex) `chmod 777 test : test 파일의 user, group, other의 권한을 모두 rwx로 변경`
- 특수 권한
  - 4000 : `setuid` 설정 지정, 2000 : `setgid` 설정 지정, 1000 : `sticky bit` 설정 지정
  - ex) `chmod 4754 test : test 파일의 user id 설정을 지정하고, user 에게 rwx 권한 부여, group 에게 r-x 권한 부여, others 에게 r- 권한 부여`



# 1. 셸 스크립트 - 셸 스크립트 실행

Shebang line 이 있다면 → `./test.sh` (interpreter 지정 없이 실행 가능)

Shebang line 이 없다면 → `bash test.sh` ([ interpreter ] [ file ] 형식)

## 2. Cron

## 2. Cron - Cron 이란?

- ✓ Unix 기반 OS 에서 미리 \*작업을 **백그라운드에서 수행**하도록 **예약**할 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 있으며, Windows 의 서비스와 유사

## 2. Cron - Cron 사용법

- ✓ Cron 설치 : 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

## 2. Cron - Crontab

- ✓ 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 이 저장되어 있는 경로

- /var/spool/cron : 개발 사용자를 위한 crontab
- /etc/cron.d : 관리자가 직접 정한 시스템 관련 작업들을 위한 crontab
- /etc/crontab : 주기적인 패키지 설치를 위한 crontab

## 2. Cron - Crontab 기본 명령어

- ✓ 예약 작업 작성 및 수정 : `crontab -e`
- ✓ 예약 작업 삭제 : `crontab -r`
- ✓ 예약된 작업 리스트 : `crontab -l`

## 2. Cron - 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

## 2. Cron - Anacron

- ✓ 수행을 놓친 작업이 있다면 작업이 가능한 시점에서 놓쳐버린 작업을 수행해주는 daemon
- ✓ ex) 서버 보수로 실행되지 않은 작업이 있다면, anacron이 해당 작업을 수행
- ✓ anacron config 는 /etc/anacrontab 에 저장됨



감사합니다 😊

Made by loopy