

Linux

※ 초기 설치

CentOS7 Server with GUI로 설정

1. 가상머신 이름 : mail-daum
 2. CPU 1개, RAM 2G, HDD 40G, 네트워크 어댑터 NAT
 3. 계정생성
 - 계정이름 : lee
 - 그룹이름 : myoffice
 - 홈 디렉토리 : /myoffice/lee
- ```
-> mkdir myoffice
-> useradd -b /myoffice -g myoffice lee
```
4. IP는 각 대역의 30으로 설정 → NAT는 게이트웨이가 2
  5. vim 설치 및 selinux disabled 설정

sudo cat /etc/shadow → 안 먹힘(sudoers 설정이 없기 때문)

sudoers → root권한을 주는 설정을 함

6. 프로그램 - 즐겨찾기 - 응용프로그램 설치 - 생산성 - 에볼루션 설치

6-1. 설치가 안 될 경우는 yum install -y evolution

## ■ CRON 서비스

주기적으로 반복되는 일을 자동으로 실행할 수 있도록 설정하는 것이 CRON

crond 데몬 프로그램이 서비스를 제공

\* (아스터리스크) → 모든 것을 의미

1. /etc/crontab \* \* \* \* \*(분, 시, 일, 월, 요일의 순서로 구성)

▷ <예제>

매일 새벽 2시 30분마다 : 30 2 \* \* \*

매달 15일 새벽 3시마다 : 0 3 15 \* \*

매주 수요일 새벽 2시마다 : 0 2 \* \* 3

매일 오후 1시마다 : 0 13 \* \* \*

매달 10일 새벽 1시마다 : 0 1 10 \* \*

매주 월요일 아침 9시 마다 0 9 \* \* 1

매달 25일 오후 3시 마다 : 0 15 25 \* \*

user-name command to be executed

매시간 01 \* \* \* \* root run-parts /etc/cron.hourly

매일 02 4 \* \* \* root run-parts /etc/cron.daily

```
매주 03 4 * * 0 root run-parts /etc/cron.weekly
매달 42 4 1 * * root run-parts /etc/cron.monthly
```

시스템 수준의 cron : 루트 권한만 쓸 수 있기 때문에 시스템 수준이라고 표현함

▷ crontab

일반 사용자가 cron 작업을 할 때 사용

vi 에디터 형식으로 사용

# crontab -e : 크론 설정

# crontab -l : 내용 보기

# crontab -r : 삭제

◇ 백업 쉘 스크립트 만들기

```
#!/bin/sh
```

set \$(date) → \$표시는 변수를 의미

```
fname="backup-$2_$3_$4.tar.xz"
```

```
tar cfJ /backup/$fname /home
```

tar → 디렉토리 파일을 하나로 묶음

실행결과 backup-Feb\_15\_11:30:14.tar.xz 파일 생성

※ [실습] user가 매달 15일 새벽 3시반에 자신의 홈 디렉토리를 /backup 디렉토리로 백업을 할 수 있도록 설정해보세요.

- 백업할 디렉토리 : /home/user

- 목적 디렉토리 : /backup

- /backup2 drwxr-xr-x

- /backup drwxr-xr-x

- /backup/user drwxr-xr-x

<풀이>

```
vim test.sh
```

```
chmod 757 test.sh
```

```
ls -l
```

```
cp test.sh /home/user
```

```
su user
```

```
cd
```

```
pwd
```

```
ls
```

```
chmod 757 /home/user/test.sh
chown user.user /home/user/test.sh
```

```
su user
```

목적 디렉토리를 /backup/user로 바꾼 후 파일 실행

```
ls -l /backup.user
```

<셸 스크립트 바꿔줄 것>

```
#!/bin/sh
```

```
set $(date)
```

```
fname="backup-$2_$3_$4.tar.xz"
```

```
tar cfJP /backup/user/$fname /home/user
```

처음에 Directory의 Permission 권한이 없어서 Access Denied가 뜬.

```
30 3 15 * * sh /home/user/test.sh
```

## ■ at

- 일회성 예약 작업에 사용

```
at[시각]
```

```
at> 작업
```

```
at> 작업
```

```
Ctrl+D
```

at -l or atq 예약 확인

at -r(숫자) or atrm(숫자) 예약 취소

## ■ 프로세스

### ◆ 정의

- 하드디스크에 저장된 실행코드(프로그램)가 메모리에 로딩되어 활성화된 것

### ◆ 프로세스 번호(PID)

- 각각의 프로세스에 할당된 번호
- 각 프로세스는 유일한 PID를 가짐

### ◆ 포어그라운드 프로세스 (Foreground Process)

- 실행하면 화면에 나타나서 사용자와 상호작용을 하는 프로세스
- 대부분의 응용프로그램

- 포어그라운드 프로세스가 끝나기 전에는 다른 명령을 수행할 수 없음
- 포어그라운드로 실행 중인 프로세스를 강제 중지하려면 [Ctrl]+[c]
- 포어그라운드로 실행 중에 [Ctrl]+z하면 백그라운드로 전환되며 멈춤(또는 지연) 상태가 됨

background : 시스템에서 알아서 동작하는 프로세서(백신 실시간 시 같은 것)

foreground : 사용자의 상호작용에 의해서 작업이 되는 것(한글, Excel 등)

리눅스에서 foreground processor  
ping, vim이 대표적이다.

리눅스 프로세스도 트리형식으로 이루어져있음

systemd : 가장 먼저 실행됨

#### ◆ 백그라운드 프로세스 (Background Process)

- 실행은 되었지만, 화면에는 나타나지 않는 프로세스
- 백신, 서버 데몬 등

#### ◆ 부모 프로세스와 자식 프로세스

- 모든 프로세스는 부모 프로세스를 가지고 있음
- 부모 프로세스를 종료하면 자식 프로세스도 자동으로 종료

ps tree -p : 프로세스 트리 확인



#### ◆ 프로세스 관련 명령

- ps [옵션]:현재 프로세스의 상태를 확인 (현재 사용자)

ps-l : 현재 터미널에서 동작되고 있는 프로세스에 관한 정보 출력

ps -e : 모든 프로세스 보기  
ps -ef : 모든 사용자의 프로세스에 관한 정보 출력  
ps -ef | grep[프로세스] : 특정 프로세스에 대한 정보 출력  
- 특정 프로세스가 동작하는지 확인 시 사용  
ps -aux : 프로세스 전체 상태 확인

## ■ 프로세스

ps -ef  
ps -el  
ps -aux

kill -(kill 리스트나옴)  
kill -9(강종)  
kill -15(정상종료 : 기본값)

top : 윈도우의 작업관리자와 비슷한 역할을 함  
[shift] + t : 실행된 시간이 큰 순서대로 정렬  
[shift] + m : 메모리 사용량이 큰 순서대로 정렬  
[shift] + p : cpu사용량이 큰 순서대로 정렬  
k : 프로세스 종료, k 입력후 PID입력 -> signal 9  
q : 닫기

## ■ 파일 시스템

자료를 저장하고 파일의 종류의 따라 파일의 이름 길이 수 파티션 등 지정이 됨.  
FAT32 : 단일파일 크기 제한이 있었음(4G이상 제한)  
NTFS : 윈도우  
NBR GPT 방식을 사용함  
NBR은 4TB의 용량까지만 인식함

### <리눅스 파일 시스템>

EXT2  
EXT3  
EXT4  
XFS

서버용은 마이그레이션이 힘들기 때문에 드라이버 등 환경이 잘 안바뀜  
리눅스 업그레이드나 업데이트할 때 기존에 있는 것과 맞춰줘야 함.

파일과 그 안에 든 자료를 저장하고 찾기 쉽도록 유지, 관리하는 방법을 말한다.  
파일 시스템의 종류에 따라 파티션의 수와 크기, 파일 이름의 길이, 크기, 파일 수 등에 차이가 있다.

## ■ HDD Interface

### ◆ IDE

- 병렬 방식
- 4개의 장치 장착 가능

### ◆ SCSI

- 직렬 방식
- 16개의 장치 장착 가능

### ◆ SATA

- 메인 보드에 있는 포트에 직접 연결하는 방식
- 메인 보드의 포트 개수만큼 장착 가능

mount : 연결을 시켜주는 것을 의미 (내가 지정한 디렉토리)

기존에 무언가가 있어야 마운팅가능

즉, 마운트 포인트가 있어야 마운팅이 가능

루트에 디렉토리를 연결하거나 잡으면 마운팅이 됨

리눅스 시스템은 서버가 켜거나 긴 시간을 두고 변경해야 할 때 변경하기 때문

sda는 시스템만 돌아가게 하고 나머지는 디스크로 연결하여 마운팅을 시키면 홈 디렉토리를 sdb로 연결