

사용자와 그룹(1)

- 리눅스는 다중 사용자 시스템(Multi-User System) 임
- 기본적으로 root라는 이름을 가진 수퍼유저(Superuser)가 있으며, 모든 작업을 할 수 있는 권한이 있음
- 모든 사용자를 하나 이상의 그룹에 소속되어 있음
- 사용자는 /etc/passwd 파일에 정의되어 있음



- 각 행의 의미는 다음과 같음
사용자 이름:암호:사용자 ID:사용자가 소속된 그룹 ID:전체 이름:홈 디렉터리:기본 셸

사용자와 그룹(2)

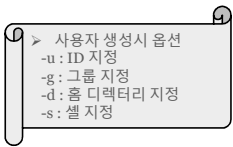
- 사용자의 비밀번호는 /etc/shadow 파일에 정의되어 있음
- 그룹은 /etc/group 파일에 정의되어 있음



- 각 행의 의미는 다음과 같음
그룹명:비밀번호:그룹 id:그룹에 속한 사용자명

사용자와 그룹 관련 명령어(1)

- useradd
새로운 사용자를 추가
예) # useradd newuser
- passwd
사용자의 비밀번호를 지정하거나 변경
예) # passwd newuser
- usermod
사용자의 속성을 변경
예) # usermod -g root newuser
- userdel
사용자를 삭제
예) # userdel newuser



사용자와 그룹 관련 명령어(2)

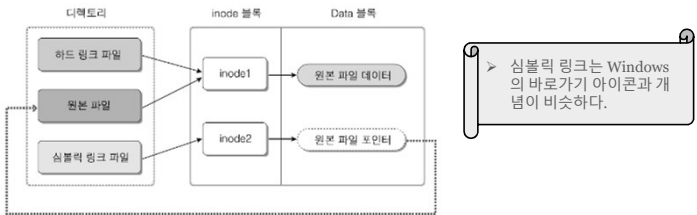
- chage
사용자의 암호를 주기적으로 변경하도록 설정
예) # chage -m 2 newuser
- groups
현재 사용자가 속한 그룹을 보여줌
예) # groups
- groupadd
새로운 그룹을 생성
예) # groupadd newgroup
- groupmod
그룹의 속성을 변경
예) # groupmod -n newgroup mygroup

사용자와 그룹 관련 명령어(3)

- `groupdel`
그룹을 삭제
예) `# groupdel newgroup`
- `gpasswd`
그룹의 암호를 설정하거나, 그룹의 관리를 수행
예) `# gpasswd newgroup`

링크

- 파일의 링크(Link)에는 하드 링크(Hard Link)와 심볼릭 링크(Symbolic Link 또는 Soft Link) 두 가지가 있음



- 하드 링크를 생성하면 "하드링크파일"만 하나 생성되며 같은 inode1을 사용 (명령 : `# ln 링크대상파일이름 링크파일이름`)
- 심볼릭 링크를 생성하면 새로운 inode2를 만들고, 데이터는 원본 파일을 연결하는 효과 (명령 : `# ln -s 링크대상파일이름 링크파일이름`)

파일의 압축과 묶기 (1)

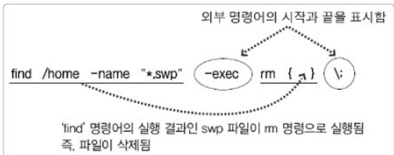
- 파일 압축
 - 압축파일 확장명은 xz, bz2, gz, zip, Z 등
 - xz나 bz2 압축률이 더 좋음
- 파일 압축 관련 명령
 - xz : 확장명 xz로 압축을 하거나 풀어준다
예) xz 파일명
`xz -d 파일명.xz`
 - bzip2 : 확장명 bz2로 압축을 하거나 풀어준다
예) bzip2 파일명
`bzip2 -d 파일명.bz2`
 - bunzip2 : "bzip2 -d" 옵션과 동일한 명령어
 - gzip : 확장명 gz로 압축을 하거나 풀어준다
예) gzip 파일명
`gzip -d 파일명.gz`
 - gunzip : "gzip -d" 옵션과 동일한 명령어

파일의 압축과 묶기 (2)

- 파일 묶기
 - 리눅스(유닉스)에서는 '파일 압축'과 '파일 묶기'는 원칙적으로 별개의 프로그램으로 수행
 - 파일 묶기의 명령어는 'tar'이며, 묶인 파일의 확장명도 'tar'이다
- 파일 묶기 명령(tar)
 - tar : 확장명 tar로 묶음 파일을 만들어 주거나 묶음을 풀어 준다
동작 : c(묶기), x(풀기), t(경로확인)
옵션 : f(파일), v(과정보이기), J(tar+xz), z(tar+gzip), j(tar+bzip2)
 - 사용 예
`# tar cvf my.tar /etc/sysconfig/` → 묶기
`# tar cvfj my.tar.xz /etc/sysconfig/` → 묶기 + xz 압축
`# tar xvf my.tar` → tar 풀기
`# tar xvfj my.tar.xz /etc/sysconfig/` → xz 압축 해제 + tar 풀기

파일 위치 검색

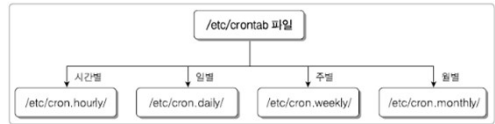
- find [경로] [옵션] [조건] [action] : 기본 파일 찾기
 - [옵션] -name, -user(소유자), -newer(전,후), -perm(허가권), -size(크기)
 - [action] -print(디폴트), -exec (외부명령 실행)
 - 사용 예
 - # find /etc -name "*.conf"
 - # find /bin -size +10k -size -100k
 - # find /home -name "*.swp" -exec rm { } \;



- which 실행파일이름 : PATH에 설정된 디렉터리만 검색
- whereis 실행파일이름 : 실행 파일, 소스, man 페이지 파일까지 검색
- locate 파일이름 : 파일 목록 데이터베이스에서 검색

CRON과 AT (1)

- cron
 - 주기적으로 반복되는 일을 자동적으로 실행될 수 있도록 설정
 - 관련된 데몬(서비스)은 "crond", 관련 파일은 "/etc/crontab"



- /etc/crontab 예
 - 01 * * * * root run-parts /etc/cron.hourly
 - 02 4 * * * root run-parts /etc/cron.daily
 - 03 4 * * 0 root run-parts /etc/cron.weekly
 - 42 4 1 * * root run-parts /etc/cron.monthly
- 첫 줄은 매시간 1분에 /etc/cron.hourly 디렉터리 안에 있는 명령들을 자동으로 실행한다

CRON과 AT (2)

- at
 - cron은 주기적으로 반복되는 작업을 예약하는 것이지만, at는 일회성 작업을 예약
 - 사용 예
 - 예약 : # at <시간>
 - 예) # at 3:00am tomorrow → 내일 새벽 3시
 - # at 11:00pm January 30 → 1월 30일 오후 11시
 - # at now + 1 hours → 1시간 후
 - at> 프롬프트에 예약 명령어 입력 후 [Enter]
 - 완료되면 [Ctrl] + [D]
 - 확인 : # at -l
 - 취소 : # atrm <작업번호>



<실습> cron, at 실습

- 실습목표
 - CRON을 활용하여 매월 15일 새벽 3시 1분에 /home 디렉터리와 그 하위 디렉터리를 /backup 디렉터리에 백업하는 방법을 익힌다.
 - AT의 사용법을 익힌다.
- 실습 화면 (백업 진행)

```
root@localhost:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
[root@localhost ~]# date 011503002017  
2017. 01. 15. (일) 03:00:00 KST  
[root@localhost ~]# systemctl restart crond  
[root@localhost ~]# ls -l /backup  
합계 4  
-rw-r--r-- 1 root root 524 1월 15 03:01 backup-01.15..tar.xz  
[root@localhost ~]# date 021503002017  
2017. 02. 15. (수) 03:00:00 KST  
[root@localhost ~]# systemctl restart crond  
[root@localhost ~]# ls -l /backup  
합계 8  
-rw-r--r-- 1 root root 524 1월 15 03:01 backup-01.15..tar.xz  
-rw-r--r-- 1 root root 524 2월 15 03:01 backup-02.15..tar.xz  
[root@localhost ~]#
```

네트워크 관련 필수 개념 (1)

- TCP/IP
 - 컴퓨터끼리 네트워크 상으로 의사소통을 하는 “프로토콜” 중 가장 널리 사용되는 프로토콜의 한 종류
- 호스트 이름(Hostname)과 도메인 이름(Domain name)
 - 호스트 이름은 각각의 컴퓨터에 지정된 이름
 - 도메인 이름(또는 도메인 주소)는 hanbit.co.kr과 같은 형식
- IP 주소
 - 각 컴퓨터의 랜카드에 부여되는 중복되지 않는 유일한 주소
 - 4바이트로 이루어져 있으며, 각 자리는 0~255까지의 숫자
 - 예) Server의 IP 주소는 192.168.111.100
- 네트워크 주소
 - 같은 네트워크에 속해 있는 공통된 주소 (예 : 192.168.111.0)

13/24

네트워크 관련 필수 개념 (2)

- 브로드캐스트(Broadcast) 주소
 - 내부 네트워크의 모든 컴퓨터가 듣게 되는 주소
 - 현재 주소의 제일 끝자리를 255로 바꾼 주소(C클래스)
- 게이트웨이(Gateway), 라우터(Router)
 - 라우터 = 게이트웨이
 - 네트워크 간에 데이터를 전송하는 컴퓨터 또는 장비
 - Vmware의 게이트웨이 주소는 192.168.111.2로 고정
- 넷마스크(Netmask) & 클래스(Class)
 - 넷마스크 : 네트워크의 규모를 결정(예:255.255.255.0-C클래스)
- DNS(Domain Name System) 서버(= 네임 서버) 주소
 - URL을 해당 컴퓨터의 IP주소로 변환해 주는 서버
 - 설정 파일은 /etc/resolv.conf
 - Vmware를 사용하면 Vmware가 192.168.111.2번을 게이트웨이 및 DNS 서버로, 192.168.111.254를 DHCP 서버로 설정함.

14/24

네트워크 관련 필수 개념 (3)

- 리눅스에서의 네트워크 장치 이름
 - CentOS 7은 랜카드를 ens32 또는 ens33으로 할당함
 - 이전버전에서는 eth0, eth1 등으로 인식했음.
 - Vmware에 CentOS를 설치할 경우에는 Vmware 버전에 따라서 완전히 다른 이름으로 인식할 수도 있음.

15/24

중요한 네트워크 관련 명령어 (1)

- nmtui
 - 네트워크와 관련된 대부분의 작업을 이 명령어에서 수행
 - 자동 IP 주소 또는 고정 IP주소 사용 결정
 - IP주소, 서브넷 마스크, 게이트웨이 정보 입력
 - DNS 정보 입력
 - 네트워크 카드 드라이버 설정
 - 네트워크 장치(ens32)의 설정
 - 텍스트 기반으로 작동함
- systemctl <start/stop/restart/status> network
 - 네트워크의 설정을 변경한 후에, 변경된 내용을 시스템에 적용시키는 명령어

➢ X 윈도의 그래픽 모드를 사용하려면 'gnome-control-center network' 명령어를 사용한다.

16/24

중요한 네트워크 관련 명령어 (2)

- ifup <장치이름> 및 ifdown <장치이름>
 - 네트워크 장치를 On 또는 Off 시키는 명령어
- ifconfig <장치이름>
 - 장치의 IP주소 설정 정보를 출력
- nslookup
 - DNS 서버의 작동을 테스트하는 명령어
- ping <IP주소 또는 URL>
 - 해당 컴퓨터가 네트워크상에서 응답하는지를 테스트하는 간편한 명령어

17/24

네트워크 설정과 관련된 주요 파일

- /etc/sysconfig/network
 - 네트워크의 기본적인 정보가 설정되어 있는 파일
- /etc/sysconfig/network-scripts/ifcfg-ens32
 - ens32 장치에 설정된 네트워크 정보가 모두 들어 있는 파일
- /etc/resolv.conf
 - DNS 서버의 정보 및 호스트 이름이 들어 있는 파일
- /etc/hosts
 - 현 컴퓨터의 호스트 이름 및 FQDN이 들어 있는 파일

> "nmcli"명령을 수행하면 위 4개 파일이 변경되는 효과임.
> 즉, 이 4개파일을 직접 편집하면 "nmcli"를 사용하지 않아도 됨.

18/24

네트워크 보안을 위한 SELinux

- SELinux(Security Enhanced Linux)는 보안에 취약한 리눅스를 보호하기 위해 탄생
- 강제(Enforcing), 허용(Permissive), 비활성(Disabled) 세 가지 레벨
- 설정 파일인 /etc/sysconfig/selinux를 편집하거나, system-config-selinux 명령으로 설정
- 작동 방식
 - '강제(Enforcing)'는 시스템 보안에 영향을 미치는 기능이 감지되면 아예 그 기능이 작동되지 않도록 시스템에서 막아줌.
 - '허용(Permissive)'은 시스템 보안에 영향을 미치는 기능이 감지되면 허용은되지만 그 내용이 로그에 남음.
 - '비활성(Disabled)'은 SELinux를사용하지 않음

19/24

파이프, 필터, 리다이렉션

- 파이프(pipe)
 - 두 개의 프로그램을 연결해 주는 연결통로의 의미
 - "|"문자를 사용함
 - 예) # ls -l /etc | more
- 필터(filter)
 - 필요한 것만 걸러 주는 명령어
 - grep, tail, wc, sort, grep, awk, sed 등
 - 주로 파이프와 같이 사용
 - 예) # ps -ef | grep bash
- 리다이렉션 (redirection)
 - 표준 입출력의 방향을 바꿔 줌
 - 예) ls -l > list.txt
 - soft < list.txt > out.txt

20/24

프로세스, 데몬 (1)

- 정의
 - 하드디스크에 저장된 실행코드(프로그램)가, 메모리에 로딩되어 활성화된 것
- 포그라운드 프로세스(Foreground Process)
 - 실행하면 화면에 나타나서 사용자와 상호작용을 하는 프로세스
 - 대부분의 응용프로그램
- 백그라운드 프로세스(Background Process)
 - 실행은 되었지만, 화면에는 나타나지 않고 실행되는 프로세스
 - 백신 프로그램, 서버 데몬 등
- 프로세스 번호
 - 각각의 프로세스에 할당된 고유번호
- 작업 번호
 - 현재 실행되고 있는 백그라운드 프로세스의 순차번호

21/24

프로세스, 데몬 (2)

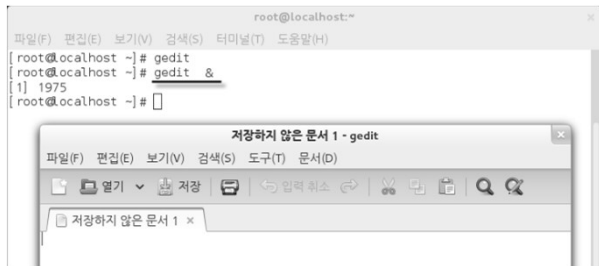
- 부모 프로세스와 자식 프로세스
 - 모든 프로세스는 부모 프로세스를 가지고 있음
 - 부모 프로세스를 kill 하면, 자식 프로세스도 자동으로 kill 됨
- 프로세스 관련 명령
 - ps
 - 현재 프로세스의 상태를 확인하는 명령어
 - "ps -ef | grep <프로세스 이름>"을 주로 사용함
 - kill
 - 프로세스를 강제로 종료하는 명령어
 - "kill -9 <프로세스 번호>"는 강제 종료
 - pstree
 - 부모 프로세스와 자식 프로세스의 관계를 트리 형태로 보여 줌

22/24



<실습> 프로세스 연습

- 실습목표
 - 포그라운드 프로세스와 백그라운드 프로세스의 상호 전환 연습을 한다.
- 실습 화면 (백그라운드 프로세스로 실행)



23/24

서비스와 소켓

- 서비스
 - 시스템과 독자적으로 구동되어 제공하는 프로세스를 말한다. 예로 웹 서버(httpd), DB 서버(mysql), FTP 서버(vsftpd) 등이 있다.
 - 실행 및 종료는 대개 'systemctl start/stop/restart 서비스이름'으로 사용된다.
 - 서비스의 실행 스크립트 파일은 /usr/lib/systemd/system/ 디렉터리에 '서비스이름.service'라는 이름으로 확인할 수 있다. 예를 들어 웹 서비스는 httpd.service라는 이름의 파일로 존재한다.



24/24