리눅스 구조

2007. 10. 17

안 효 창

Contents

vi

- vi 편집기
  - 유닉스/리눅스 시스템의 표준 편집기
  - X윈도우 환경뿐 아니라 콘솔환경에서도 같은 방법으로 제공
  - 리눅스는 기능이 향상된 vim이라는 클론이 사용됨
  - 실행
    - # vi abc.txt
    - # vi
  - vi 의 세가지 모드 : 입력/명령/실행
    - 입력모드 : 화면에 글자를 입력하는 모드
    - 명령모드 : 파일을 편집하는 작업
    - 실행모드 : 파일의 저장, 읽기, 외부명령실행 , 종료 등

vi

- 도움말
  - vi 사용 중 [F1]
  - 실행모드에서 help 입력 – [:help]
  - 상단 : 도움말 화면
  - 하단 : 편집 하던 내용
  - 도움말 -> 편집 중 화면 이동 : [Ctrl] + ww
  - 편집 중 화면 -> 도움말 이동 : [Ctrl] + ww
  - 도움말 종료
    - 실행 모드에서 q 입력 – [:q]

vi

- 커서의 이동
  - 문자 이동

문자 이동	설명
h	키 커서 왼쪽으로 한 문자 이동
j	커서 위로 한 행 이동
k	커서 아래로 한 행 이동
l	키 커서 오른쪽을 한 문자 이동

  - 단어 단위 이동

단어의 단위 이동	설명
w	커서 다음 단어 처음문자로 이동
b	커서 이전 단어 처음으로 이동
e	커서 다음 단어 끝으로 이동

vi

- 행 단위 이동

행 단위 이동	설명
^	커서가 위치한 현재 행의 첫 단어로 이동
$	커서가 위치한 현재 행의 마지막 단어로 이동
e	커서가 위치한 다음 단어 끝으로 이동
G	파일 내의 마지막 행으로 이동
1G	파일 내의 1행으로 이동
nG	N행으로 이동

- 페이지 이동

페이지 이동	설명
[Ctrl] + r	다음 페이지 이동
[Ctrl] + b	이전 페이지 이동

vi

- 입력모드
  - 입력모드로 들어가기 위한 키 – a, A, i , I, o, O
    - ESC 입력 전까지 문자 입력만 가능
  - 입력모드에서 빠져 나오기 위한 키(명령모드로 전환) – ESC
    - 다른 모드 실행
  - 저장 (:w)및 종료(:q)
  - 저장과 종료를 동시에 (:wq)
  - 저장하지 않고 종료 (:q!)

입력 모드 키	설명
a	커서 다음부터 문자 입력
A	커서가 위치한 행의 마지막 문자 뒤에서부터 입력
i	커서 앞에서부터 문자 입력
I	커서가 위치한 행의 첫 문자 앞에서부터 입력
o	커서 아래 행을 열고 입력
O	커서 위 행을 열고 입력

vi

- 명령모드
  - 명령모드로 전환 – ESC키 입력
  - 삭제

삭제	설명
x	커서가 위치한 한 문자 삭제
dw	커서 오른쪽 단어 삭제
db	커서 왼쪽 단어 삭제
dd	커서가 위치한 행 삭제
ndd	현재 커서가 위치한 행 기준으로 아래로 n줄 삭제

vi

- 복사/잘라내기/붙여 넣기

복사/잘라내기/붙여넣기	설명
yy	커서가 위치한 행을 메모리로 복사
nyy	커서가 위치한 행 기준으로 아래로 n행 복사
p	메모리에 저장된 내용을 현재 커서 아래 행으로 붙여 넣기
np	메모리에 저장된 내용을 현재 커서 아래 행으로 n번 붙여 넣기
dd	커서가 위치한 행 잘라내기
ndd	현재 커서가 위치한 행 기준으로 아래로 n행 잘라내기

- 행 연결

행 연결	설명
j	커서가 위치한 행과 다음 행을 하나의 행으로 묶음

vi

- 복구

복구	설명
u	이전 단계로 복구(undo)
[Ctrl] + r	u키로 복구된 상황을 되돌리기(redo)
e	커서 다음 단어 끝으로 이동

- 치환

치환	설명
r + 문자	커서가 위치한 문자를 다른 문자로 치환
R	치환모드로 변환, 모든 문자 변경
cw	커서가 위치한 문자 삭제 후 입력모드로 진입

vi

- 변환

검색	설명
~	커서가 위치한 문자가 대문자라면 소문자로, 소문자라면 대문자로 변환

- 검색

검색	설명
/검색어	커서가 위치한 행 포함 아래로 해당 검색어 검색
?검색어	커서가 위치한 행 포함 위로 해당 검색어 검색
n	커서의 위치를 검색한 검색어의 다음 위치로 이동

vi

- 실행모드
  - 실행모드 전환방법 : 명령모드에서 콜론(:) 을 입력
  - 치환관련 명령

삭제	설명
:%s/old/new/g	파일 내의 특정 단어를 새로운 단어로 치환
:%s/^old/new/g	파일 내의 특정 단어로 시작하는 단어만 새로운 단어로 치환
:%s/old$/new/g	파일 내의 특정 단어로 끝나는 단어만 새로운 단어로 치환
:%s/aaa//g	파일 내의 aaa라는 단어를 삭제
:10,50s/old/new/g	파일 내의 10행에서 50행 사이의 특정 단어를 새로운 단어로 치환

  - 특수 문자를 사용 할 경우
    - ₩를 사용 – 특수 문자를 문자로 인식

파일관련 명령

삭제	설명
:w 파일명	변경한 파일 내용 저장
:q	vi 종료
:e 파일명	vi 실행 후 특정 파일을 불러들여 편집
:r 파일명	특정 파일을 불러들임
:! 명령어	외부명령어 실행

외부 명령어 실행 관련

삭제	설명
:! 명령어	외부명령어 실행

vi

- 환경설정
  - 환경설정파일
    - /usr/share/vim/vim61/vimrc_example.vim을 자신의 홈디렉토리에 .vimrc 로 복사하여 편집

vi

- 주 설정내용

설정값	적용 내용
set autoident	자동 들여쓰기
set cident	C 언어 프로그램 작성시 자동 들여쓰기
set autowrite	파일 수정 시 자동 저장
set ruler	커서 위치 표시
set backspace=indent, eol, start	입력 모드에서 backspace 허용
set backup	백업 파일 자동 생성
set history=50	명령어 히스토리 기록(50라인)
set showcmd	명령어 표시
set incsearch	순차적 검색 허용
syntax on	컬러 터미널의 경우 컬러로 표시
set hlsearch	문자열을 검색할 때 컬러로 표시
set magic	문자열을 검색할 때 와일드카드 사용
set binary	바이너리 파일 편집 가능
set ignorecase	문자열을 검색할 때 대소문자 구별

vi

- 주의 사항
  - 저장하지 않고 종료했을 경우의 파일복구
    - vi a.txt 강제종료 -> .a.txt.swap 파일 생성
  - 다시 편집할 때 복구 메세지 출력
    - R : 해당화면에서 복구
    - E : 그냥 고치기 (나중에 :recover를 통해 복구가능)

시스템 정보

- CPU 정보 (명령어 : uname)
  - # uname – p ➔ i686은 펜티엄 4를 의미
  - # uname --help
- 메모리 정보 (명령어 : free)
  - # free –m ➔ MB 단위
  - # free –k ➔ KB 단위
- 프로세스 정보 (명령어 : ps , pstree)
  - # ps auxw | grep httpd
  - # pstree

시스템 정보

- 프로세스별 CPU와 메모리 점유율 (명령어 : top)
  - 시스템이 갑자기 느려졌을 경우 사용
  - # top
  - 주요 영역별 의미
    - PID : 프로세스 ID
    - USER : 소유주
    - SIZE : 데이터 크기 (단위 KB)
    - %CPU : CPU 점유율 (%)
    - %MEM : 메모리 점유율 (%)
  - 종료 : q

시스템 정보

- 마운트한 파일시스템의 정보 (명령어 : df)
  - 마운트한 파일시스템의 전체크기, 사용한 공간, 여유공간, 사용율, 마운트 포인트 등의 정보
  - # df -h
- 디렉토리별 용량 (명령어 : du)
  - 각 디렉토리별 용량 확인
  - # du --max-depth=1 -h /home ➔ 사용자별 디스크 사용 현황
- 사용중인 장치정보 (명령어 : lsdev)
  - 시스템에서 인식하고 있는 장치(DMA,IRQ,I/O포트 등)의 정보
  - # lsdev
  - 세부적인 정보확인 ➔ /etc/sysconfig/hwconf 파일

시스템 정보

- 현재 시스템 사용자 정보 (명령어 : w)
  - 사용자명, 사용중인 터미널 사용중인 명령어 등의 현황
  - # w
- 시스템에 설정된 시간 (명령어 : date , rdate)
  - # date ➔ 현재 시간
  - # rdate -s time.bora.net ➔ 표준 시간을 가져옴
  - # clock ➔ CMOS의 시간
  - # clock -w ➔ 현재시간은 CMOS에 덮어 씀
  - 부팅 시마다 rdate로 정확한 시간을 가져와서 CMOS에 덮어 쓰기 위해서는/etc/rc.local을 편집

시스템 정보

- 시스템에 연결된 소켓 정보
  - # netstat
- 최근 시스템 접속자 정보
  - # last
- 네트워크 장치 설정 정보
  - # ifconfig

시스템 정보

- IP 주소 변경 방법 1 ➔ 임시 변경
  - # ifconfig eth0 down == # ifdown eth0
  - # ifconfig eth0 inet <new-ip> netmask <netmask>
  - # ifconfig eth0 up == # ifup eth0
  - # ifconfig eth0
- IP 주소 변경 방법 2
  - # redhat-config-network
    ➔ 네트웍 서비스 재시작 (# service network restart)
- IP 주소 변경 방법 3
  - /etc/sysconfig/network-scripts/ifcfg-eth0 파일 편집
    ➔ 네트웍 서비스 재시작

디렉토리 구조와 파일 특성

- 리눅스의 디렉토리/파일의 특성 요약
  - 처음 설치 시에 생성되는 디렉토리는 각 파일의 목적이나 시스템의 특징에 따라 분류되어 있음
  - 각 특성에 맞게 해당 디렉토리에 저장되어 있음
  - 주요 디렉토리의 이름을 바꾸거나 이동하면 안됨
  - 리눅스에는 파일만 존재하며, 각 파일은 속성에 의해서 일반 파일,디렉토리,링크,장치파일 등으로 구분됨
  - 리눅스는 다중사용자 환경이므로 소유자,다른 사용자,그룹으로 각각의 권한이 부여됨 ➔ 파일의 속성

디렉토리 구조와 파일 특성

- 디렉토리의 분류
  - /
    - 루트 디렉토리, 최상위 디렉토리
    - 모든 디렉토리는 “/”를 기준으로 생성
    - 별도로 생성한 파일시스템은 “/”하부의 디렉토리에 연결하여 사용
    - C:,D: 등의 드라이브 개념은 없음.
    - 예) CD-Rom
      - CD 삽입 ➔ 마운트 명령 ➔ /mnt/cdrom 에 연결됨
  - /bin
    - 기본 명령어가 존재하며, 사용 빈도가 높음

디렉토리 구조와 파일 특성

- 디렉토리의 분류 (계속)
  - /boot
    - 부팅에 필요한 핵심 파일들이 위치함
    - 부팅과정 요약

      전원공급 ➔ 하드웨어체크(CMOS) ➔ MBR읽기(GRUB가 커널위치 참조) ➔ /boot영역의 커널로 리눅스 부팅
  - /dev
    - 장치파일이 위치, 파일이 저장되지 않음
    - 블록장치(block device)
      - 읽기/쓰기 장치
      - IDE하드(hda,hdb…), CD-ROM, SCSI하드(sda,sdb…), 플로피(fd0), 테이프
    - 문자장치(character device)
    - 블록장치를 제외한 장치
      - 콘솔(tty1…), 시리얼포트(ttyS0…), 프린터포트(lp0…), 마우스(psaux), 사운드(audio), …

디렉토리 구조와 파일 특성

- 디렉토리의 분류 (계속)
  - /etc
    - 응용프로그램과 서버프로그램의 환경 설정에 필요한 설정파일이 위치
    - 중요한 디렉토리이므로, 수시 백업
  - /home
    - 일반사용자의 홈디렉토리가 위치
    - 별도의 파일시스템으로 분할을 권장
  - /lib
    - 공유라이브러리 및 부팅시의 커널 모듈 포함
  - /lost+found
    - 부팅시 파일시스템 손상이 되었을 때 사용되는 디렉토리 (fsck : File System CheckK)

디렉토리 구조와 파일 특성

- 디렉토리의 분류 (계속)
  - /mnt
    - 기본 마운트 포인트 제공 (필수 사용은 아님)
    - /mnt/cdrom , /mnt/floppy
  - /root
    - 관리자(root)용 홈 디렉토리
  - /sbin
    - 관리자가 사용하는 시스템 운영 명령어가 위치
    - fsck, ifconfig, lsmod, mkfs, reboot…
  - /tmp
    - 임시 파일 저장 디렉토리
    - 재구동시 모두 삭제됨

디렉토리 구조와 파일 특성

- 디렉토리의 분류 (계속)
  - /usr
    - 프로그램 설치지 패키지의 대부분 파일이 위치
  - /var
    - 내용이 자주 변경되는 정보가 저장
    - 로그, 메일스풀, 프린트스풀 …
    - 아파치 웹서버, MySQL, FTP 서버 …
    - 서버 운용시 별도파일시스템으로 분할 권장

디렉토리 구조와 파일 특성

- 파일 속성
  - "ls -l" 명령이 보여주는 파일의 속성

-rwxrwxrwx	2	root	root	300	3월17일	11:34	filename.txt
속성/허가권	링크수	소유자	그룹	파일크기	날짜	시간	파일명

-	rwx	r-x	r--
1	7	7	7
파일속성	소유자(user)	그룹(group)	그외(other)

- 파일의 속성
  - - : 일반파일
  - b : 블록장치
  - c : 문자장치
  - d : 디렉토리
  - l : 심볼릭 링크
  - p : 파이프
  - s : 소켓

- 파일명 : 최대 256자
- (.) 사용규정 없음
- 확장자 개념 없음
- 특수문자도 사용가능
  : 도스와 자료교환시 문제발생 소지 있음 (*,공백,/ 제외)
- 대소문자 구분
- 색깔로 표시 (/etc/DIR_COLOR)

디렉토리 구조와 파일 특성

- 파일 속성 (계속)
  - 숫자로 표현한 파일허가권
    - 7 (rwx = 4+2+1) 읽기/쓰기/실행 가능
    - 6 (rw- = 4+2+0) 읽기/쓰기 가능
    - 5 (r-x = 4+0+1) 읽기/실행 가능
    - 4 (r-- = 4+0+0) 읽기만 가능
    - 3 (-wx = 0+2+1) 쓰기/실행 가능
    - 2 (-w- = 0+2+0) 쓰기만 가능
    - 1 (--x = 0+0+1) 실행만 가능
    - 0 (--- = 0+0+0) 읽기/쓰기/실행 불가능
  - 파일허가권 변경은 root 및 소유자만 가능
    - 명령어 : # chmod
  - 파일 소유권 변경은 root 만 가능
    - 명령어 : # chown

디렉토리 구조와 파일 특성

- umask ➔ 최초의 자동 퍼미션 부여
  - # umask ➔ 기본으로 0022 세팅
    - r + w + x = 7 (777)
    - 실행 x를 제외한 666 - umask(022) = 644 (rw-r--r--)
  - 예)
    - # touch test2 ➔ rw-r--r--
    - # umask 002
    - # touch test3 ➔ rw-rw-r--

디렉토리 구조와 파일 특성

- 파일 퍼미션 변경 (Change Mode)
  - 1) # chmod [옵션] [사용자] [+또는-또는=] [퍼미션] [파일/디렉토리]
  - 2) # chmod [퍼미션] [파일/디렉토리]
    - 옵션 : -R 하위까지 변경
    - 사용자 : u, g, o, a(all)
    - + | - | = : +(새로추가) , -(삭제), =(reset)
    - 퍼미션 : r,w,x,s,u,t
  - 예)
    - #chmod go-w file ➔ 그룹과 other의 읽기권한 삭제
    - #chmod a+x ➔ 모두에게 실행권한 부여
    - #chmod 755 file ➔ rwxr-xr-x 퍼미션 부여

디렉토리 구조와 파일 특성

- 파일의 소유권
  - 예) # touch test ➔ 빈파일 생성
  - # ls –l test
- 소유권 변경 ➔ root 권한 필요
  - # chown <변경할사용자id> <변경할파일또는디렉토리>
  - # chgrp < 변경할그룹id> <변경할파일또는디렉토리>
  - -R 옵션 : 하위디렉토리 및 파일까지 변경
  - 예) #chown mrwoo.mrwoo test
  - == #chown mrwoo test + #chgrp mrwoo test

디렉토리 구조와 파일 특성

- 기타 허가권필드
  - s (set UID) ➔ 파일 실행시 소유권자의 권한으로 실행
    - 예) # ls -al /usr/bin/passwd
  - g (get GID) ➔ 파일 실행시 소유그룹의 권한으로 실행
  - t (Sticky) ➔ 디렉토리 삭제시 삭제는 소유자만 가능
    - 예) # ls -al /tmp

- Any Questions ?