**ls**

기본적으로 특정 디렉토리에 있는 내용 디렉토리나 파일 등을 확인하는 명령어로 윈도의 dir 명령어와 비슷하다. -a 옵션을 사용해 숨겨져 있는 내용들을 볼 수 있고, -l 명령어를 사용해 각 내용들의 퍼미션이나, 포함된 파일 수나 소유자 파일 크기 등 자세한 내용들을 확인할 수 있다. 그래서 일반적으로 -al 옵션을 붙여서 가장 많이 사용한다.

**echo**

지정한 문자열을 출력하는 명령어로 문자열과 함께 개행문자를 덧붙여 출력한다. 일반적으로 쉘에서 자체적으로 포함하는 내부 환경 변수를 확인하는 경우로 많이 사용된다.

**cat**

파일의 내용을 그래도 출력해 주는 명령어로 파일의 내용을 확인하기 위해 사용하고 난 여러 파일을 한 번에 지정하여 여러 파일들을 연결하는 데 사용하기도 합니다. head 옵션을 사용해 첫 행만 출력하거나 tail 옵션을 사용하여 마지막 행을 출력하는 것도 가능합니다.

**grep**

파일에서 특정한 문자열이나 패턴을 찾는 명령어로 파이프 문자를 사용하여 다른 명령어와 연결하여 사용하는 경우가 많습니다. 예를 들어 yum list installed 명령어는 설치된 모듈들을 보여주는 명령어인데 여기서 grep 명령어를 연결하여 |grep httpd를 함께 사용하면 설치된 모듈들 중에서 httpd가 포함된 모듈들만 찾아볼 수 있습니다.

**vi**

유닉스 계열에서 가장 많이 사용되는 편집기로 비주얼 에디터이다. 유닉스 환경에서 가장 많이 쓰이는 문서 편집기로 한 줄씩 편집하는 줄 단위 편집기가 아닌 한 화면을 편집하는 방식이어서 비주얼 에디터라고 한다. 편집하다가 예기치 않게 종료되거나 정상적인 동작이 이루어지지 않을 때 원본 파일은 마지막 저장 상태로 존재하고 작업하던 과정은 swap 파일로 새롭게 생성되게 되다. 작업하던 과정을 복구하고 싶으면 원본 파일을 다시 편집기로 열어 recover 옵션을 실행 후 저장하고 종료한 다음 swap 파일을 삭제하면 된다. View 명령어를 통해 파일을 수정하는 것이 아닌 읽기 모드로 확인하는 것도 가능하다.

pwd

현재 어떠한 디렉토리 경로에 있는가를 절대 경로로 표시해 주는 명령어이다. 지금 어떠한 디렉토리에서 작업 중인지, 디렉토리의 상위 디렉토리가 무엇인지 등을 파악할 수 있다.

cd

체인지 디렉토리의 약자로 작업 디렉토리의 위치를 변경하는 명렁어이다. 기본적으로 cd 나 ~옵션을 통해서 자신의 홈 디렉토리로 이동하거나 .. 을 통해 한 단계 상위 디렉토리로 이동하는 거나 절대 경로를 통해서 원하는 디렉토리로 바로 이동하는 것이 가능하다. 명령어 창에서 작업을 마치거나 창을 닫을 때 터미널을 닫는 명령어 예를 들어 일반 사용자로 접속해 root 계정으로 접속한 상황에서 사용하면 다시 일반 사용자로 돌아감

clear

현재 화면에 보이는 것들을 지우고 커서가 화면의 왼쪽 상단으로 이동

su

switch user의 약자로 현재 계정을 로그아웃하지 않고 다른 계정으로 전환하는 명렁어이다.

env

현재 지정되어 있는 환경 변수들을 출력하거나, 새로운 환경 변수를 설정하고 적용된 내용을 출력하는 명령어다. 환경 변수란 프로세스가 컴퓨터에서 동작하는 방식에 영향을 미치는 동적인 값들의 모임으로 쉘에서 정의되고 실행하는 동안 프로그램에 필요한 변수

tty

리눅스 디바이스 드라이브 중에서 콘솔이나 터미널을 의미합니다. 여러 개의 콘솔로 1개의 리눅스에 접근할 수 있으며, 이때 2번째부터는 실제로 존재하지 않는 콘솔(키보드와 모니터가 2개가 연결된 게 아님)이므로 가상(pseudo)라는 접두어가 붙게 됩니다

free

명령어를 사용하면 쉽게 메모리 사용량과 여유량 그리고 캐싱으로 사용되는 메모리가 얼마나 있는지 파악할 수 있습니다. 실제 free 명령어는 리눅스 /proc/meminfo에서 메모리 정보를 가져와 보여줍니다. 이 명령어를 통해 설치된 통 메모리의 크기, 현재 사용량, 사용 가능한 여유 메모리 양, 등 메모리에 관해 여러 가지 정보들을 확인할 수 있으며 기본적으로 킬로 바이트 단위로 다 오며 옵션을 활용에 메가 바이트, 기가바이트 등 사용자의 기호의 따라 원하는 단위로 출력이 가능하다.

touch

touch는 파일의 날짜 시간 정보를 변경하는 명령어이다. 즉 아무런 옵션 없이 사용하면 서버의 현재 시간으로 파일의 최근 사용한 시간(access time)과 최근 변경 시간(modification time)을 변경한다. 만약 특정 옵션이 사용되지 않는다면 파일의 크기가 0인 빈 파일을 생성한다. -t라는 옵션을 사용하면 서버의 현재 시간이 아닌 지정된 시간으로 파일의 날짜 시간 정보를 변경한다.

mkdir

Make directory의 약자로 새로운 디렉토리를 만들 때 사용한다. 아무런 조건 없이 폴더명만 작성하여 명령어를 사용하면 현재 작업하고 있는 디렉토리에 하위 디렉토리로 생성되고, 절대 경로를 사용하여 원하는 곳에 바로 생성하는 것도 가능하다. p 옵션을 사용해 상위 디렉토리까지 같이 생성하는 것도 가능하다.

chmod

파일, 디렉토리의 권한(퍼미션, 허가권)을 변경하는 역할을 합니다. 반면 chown 명령을 사용하면 파일, 디렉토리의 사용자, 그룹을 변경할 수 있습니다. 쉽게 말해 파일의 소유권을 변경하는 것이다. -rwxrwxrwx 등으로 확인할 수 있는데 처음의 -는 파일인지 디렉토리인지를 구분하는 것으로 d라고 표시되면 디렉토리를 의미하고 -라고 표시되어 있으면 파일을 의미한다. 그리고 rwx를 삼 등 분할 수 있는데 처음의 rwx는 소유자, 그다음 그룹, 다른 사용자의 권한을 표시한 것이다. 각각 r는 읽기, w는 쓰기, x는 실행의 권한이다. 이런 식으로 각 알파벳들이 쓰여있으면 그 권한을 가지고 있는 것이다. 각 권한은 2진수 비트 계산으로 rwx 순으로 4,2,1로 나타내며 각 숫자들을 조합해서 모든 권한을 부여하는 777부터 없애는 000까지 표현 가능하다.

chown

chmod는 권한을 변경하는 명령어라면 이것은 파일이나 디렉토리의 사용자, 그룹을 변경할 수 있다. 즉 소유권을 변경하는 것이다. drwxr-xr-x. 5 inhatc inhatc 141 Apr 14 16:22 inhatc를 chown inhatc2 inhatc/를 사용해 디렉토리의 소유자를 inhatc2로 변경할 수 있다.

yum

레드햇 계열의 리눅스에서 사용하는 패키지 설치 관리 도구로서 패키지 설치, 업데이트, 삭제뿐만 아니라 패키지에 대한 정보와 어떠한 패키지가 설치가 되어있는 등 여러 가지 확인할 수 있다.

rm

rm은 파일이나 디렉토리를 삭제할 때 사용하는 명령어이며 -r 옵션을 붙이지 않으면 디렉토리는 삭제하지 못한다. 참고로 리눅스처럼 유닉스 형 운영체제는 삭제를 취소할 수 있는 명령어가 없다. 고로 rm 명령어로 삭제가 시작되면 되찾을 수 없다. f 옵션을 사용하게 되면 경고 없이 모두 강제(force)로 삭제한다. 디렉토리를 삭제하는 명령어는 rmdir이 있는데 이 명령어는 지우려고 하는 디렉토리 안의 파일들이 존재하면 삭제가 불가능하다. 그래서 디렉토리를 포함 내용까지 전부 지우고 싶을 때 rm – r 명령어를 사용한다.

mv

파일이나 디렉토리를 이동하거나 이름을 변경할 때 사용하는 명령어로 사용할 때 원본과 대상의 이름이 다르면 이름이 변경되고, 이동할 파일이 여러 개면 이동 모드로 작동

cp

파일을 복사하는 명령어로 copy의 약어로 파일이나 디렉토리를 복사할 때 사용

useradd

새로운 계정을 추가하는 명령어로 명령어 다음 추가하고 싶은 계정명을 작성해 실행하면 계정이 생성된다. 이 명령어는 계정이 생성될 때 다로 비밀번호를 생성해 주지 않기 때문에 psaswd 명령어를 통해서 비밀번호를 지정해야 한다.

passwd

계정을 생성한 후 일반적을 비밀번호를 지정할 때 사용한다. 그러나 옵션들을 사용해 계정의 비밀번호를 잠그거나 비밀번호 수정 날짜를 정하거나 만료일들을 추가적으로 설정할 수 있습니다.

date

현재 시간을 출력하는 명령어로 초기에는 UTC(세계 협정 시)로 출력된다. /user/share/zoneinfo/ 안에 대륙별, 국가별 설정 파일들이 존재한다. 이것을 /etc/localtime으로 연결해 우리나라 표준시로 변경이 가능하다.

ps -edf

는 프로세스의 상태를 확인하는 명령어입니다. 현재 특정 프로세스가 실행되고 있는지 실행되고 있는 프로세스가 어떤 pid를 갖고 있는지 등을 확인하는데 많이 쓰이며, 옵션을 통해 CPU와 메모리 점유율 등의 정보도 확인할 수 있습니다. e 옵션은 모든 프로세스를 표시하는 것이고 f 옵션은 프로세스의 정보를 더 많이 보여주도록 하는 옵션입니다. 어떤 유저가 프로세스를 실행시켰는지, CPU의 사용량 등 많은 정보를 알 수 있습니다.

netstat

네트워크 연결과 포트를 출력해 주는 명령어로 프로토콜의 종류, 포트가 열려있는지 닫혀있는지 포트의 상태를 알 수 있고, 라우팅 테이블, 큐를 주고받는 것과 내부와 외부의 아이피, 프로세스의 아이디나 프로그램 이름도 알 수 있다.

df

시스템에 할당된 하드디스크의 남은 용량을 확인할 때 사용되는 명령어 기본적으로 kb 단위로 출력되며 다른 옵션을 통해 사용자가 원하는 단위로 확인이 가능하다. 전체 용량이 얼마인지 그중에서 얼마나 사용 중 인지를 확인이 가능하고 마운트 지점이 표시된다.

setenforce

selinux는 리눅스 설치시 기본적으로 켜져(Enforce Mode) 있습니다. selinux는 리눅스의 보안을 강화해주는 보안강화 커널이라고 볼 수 있습니다. 하지만 selinux를 켜두면 특정 서비스가 정상적으로 동작하지 않을 수 있습니다. Enforce 모드로 설정을 하게 되면 selinux에 해당하는 규칙에 어긋나는 동작은 전부 거부해버립니다. Enforce 모드로 설정을 하게 되면 selinux에 해당하는 규칙에 어긋나는 동작은 전부 거부해버립니다. Permissive 모드는 selinux에 해당하는 규칙에 어긋나도 동작은 허용합니다. setenforce 0 은 selinux를 permissive mode로 변경하는 명령어입니다. setenforce 1 은 selinux를 enforce mode로 변경하는 명령어입니다.

rpm

원래 레드햇 계열에서 사용되던 패키지 파일이었지만 지금은 패키지를 관리하기 위한 도구로 많이 사용된다.

tar

여러 개의 파일들을 묶거나 풀 때 사용되는 명령어 파일들을 묶는다고 해서 압축한다고 오해하는 경우가 있는데 tar의 본질적인 개념은 압축이 아닌 파일을 하나로 묶는 것입니다. 용량의 변화 없이 하나의 묶음으로서 백업이나 파일의 저장, 전송 같은 상황에서 효율적으로 처리 및 관리하기 위해 만들어진 프로그램입니다. 옵션들을 사용해 파일들을 묶거나 푸는 것이 가능하여 묶여있는 상태에 파일을 추가할 수 있고 gzip이나 bzip 등을 함께 사용하여 파일 압축도 가능하다.

sftp

SSH와 마찬가지로 전송 시 암호화시켜서 전송을 하게 되는데 그중에 FTP와 같이 파일을 전송할 때 암호화 시켜서 전송합니다. SSH에 부가적으로 있는 기능들 중 하나의 기능입니다. FTP와 같은 역할을 하지만, 사용하는 포트는 FTP 포트(기본 21포트) 가 아닌 SSH 접속 시 사용하는 포트를 사용하고, SSH 접속 계정으로 로그인할 수 있는 FTP입니다. 실제로도 접속은 SSH 접속 프로그램을 이용해서 접속하는 것이 아니라 일반적으로 FTP Client 프로그램을 통해 접속을 합니다. 일반 계정을 사용하는 경우에는 이 sFTP가 큰 메리트는 없어 보입니다. 다만, 사용자가 root 인 경우에는 굉장히 유용합니다. 기본적으로 Linux Server 버전들의 특성상 스크립트 명령어들은 필수적이며, 파일을 알아보는 데에도 명령어를 사용합니다. 일반적으로 많이 쓰는 경로를 찾아가서 관련 파일들을 확인하는 것은 어렵지는 않지만, 윈도우에 익숙한 분들에게는 그 조차도 힘들일 이 될 수 있습니다.

ssh

네트워크 프로토콜 중 하나로 컴퓨터와 컴퓨터가 인터넷과 같은 Public Network를 통해 서로 통신을 할 때 보안적으로 안전하게 통신을 하기 위해 사용하는 프로토콜입니다 먼저 데이터 전송의 예로는 원격 저장소인 깃헙이 있을 수 있습니다. 소스 코드를 원격 저장소인 깃허브에 푸시 할 때 여러분은 SSH를 활용해 파일을 전송하게 됩니다. FTP나 Telnet과 같은 다른 컴퓨터와 통신을 위해 사용되는 프로토콜도 있는데 SSH를 사용하는가를 생각해 볼 수 있습니다. 그 이유는 물론 "보안"입니다. 민감한 정보를 주고받는다면 정보를 직접 네트워크를 통해 넘기기 때문에 누구나 해당 정보를 열어볼 수 있어 보안에 상당히 취약합니다.

/proc 디렉토리는 가상 파일 시스템으로 /dev와 마찬가지로 하드 디스크상에 물리적인 용량을 갖지 않는다. 현재 시스템의 하드웨어(CPU, RAM, 파일 시스템, 인터럽트, 파티션) 정보와 현재 실행되고 있는 프로세스의 정보를 확인할 수 있다. 또한, 커널이 실행되면서 생성된 각종 정보들 또한 /proc에 파일로 관리하고 있다. /proc/cpuinfo : 상세한 프로세서의 정보가 저장되어 있는 파일, /proc/diskstats : 각각의 논리적 디스크 디바이스에 대한 정보를 제공한다. /proc/meminfo : 현재 사용 중인 메모리의 사용량, /proc/mounts :마운트 된 파일시스템에 대한 정보

httpd(httpd.conf, vhost,)

Apache HTTP Server(HTTPD)라고 불리며, 오픈소스 소프트웨어 그룹인 아파치 소프트웨어 재단에서 만드는 웹서버 프로그램 DocumentRoot "/var/www/html" 서버의 웹문서가 있는 경로를 지정한다. 경로의 마지막에 '/'를 추가해선 안되고 심볼릭 링크나 Alias를 사용하여 다른 위치를 가리키도록 할 수 있다<Directory “/home/inahtc/www”> ~ </Directory> 지시자는 지정한 디렉토리 이하의 모든 웹문서들에 대하여 어떤 서비스와 기능을 허용/거부할 것인지를 설정하는 매우 중요한 지시 자이다. <Directory>지시자의 설정은 개인에 따라 다르니 각자 목적에 맞게 설정해야 한다. 기본적으로 웹서버에 존재하는 주 호스트 즉 예로 (aaa.com)이라는 도메인을 등록하여 사용하고 있지만 추가적으로 한 서버에 (bbb.com) / (ccc.com) 등 여러 도 메인을 등록하여 사용한다는 의미입니다. 가상 호스트에도 여러 종류의 방면으로 운용을 할 수 있지만 일반적으로 하나의 컴퓨터(서버)나 IP로 여러 이름의 도메인으로 각각 운용할 수 있다는 부분입니다.

환경변수

swap(dd swapon)

리눅스에서 물리적 메모리(RAM)의 용량이 가득 차게 될 경우 사용되는 여유 공간을 말합니다. 즉, 시스템이 처리하고 있는 데이터를 저장할 RAM이 충분하지 않을 때 스왑 공간에 이 데이터를 기록한다는 말입니다. 스왑 공간은 소량의 RAM을 사용하는 시스템에서는 도움이 되지만, RAM에 대한 대체로 여겨져서는 안됩니다. 스왑 공간은 하드 드라이브 상에 위치하기 때문에 물리적 메모리에 접근하는 것보다 접근 속도가 훨씬 느립니다. 스왑 공간은 스왑 파티션에 사용되거나 (권장 사항), 스왑 파일을 저장하는 데 사용되며, 또는 스왑 파티션과 스왑 파일이 함께 스왑 공간을 차지하는 것도 가능합니다.

etc/passwd

사용자 로그인 계정, 암호화된 비밀번호, UID, 기본 gid 이름 홈디렉토리 로그인 쉘이 저장/etc/passwd 파일에는 계정과 관련된 정보가 기록됩니다. 사용자 계정뿐만 아니라 root 계정, 데몬에서 사용하는 계정들의 정보도 기록됩니다. 원래는 패스워드 정보도 함께 기록이 되었으나, 최근엔 보안상의 이유로 패스워드 정보는 /etc/shadow 파일에 암호화되어 저장되게 됩니다. 내용은 다음과 같이 구성됩니다.

etc/shadow

는 암호화된 패스워드와 패스워드 설정 정책이 기재되어 있습니다. 여기서 관리자 계정과 관리자 그룹만이 이 파일을 읽을 수 있습니다. 이 파일이 만약 평문의 비밀번호의 정보를 가지고 있다면 모든 사용자의 비밀번호가 유출될 수 있습니다.

etc/group

리눅스에는 그룹(GROUP)이라는 개념이 있다. 어떤 파일이나 폴더를 특정 권한이 있는 사용자들끼리만 공유하기 위해 주로 사용한다. 그룹 name, 그룹 password, 그룹 ID, 그룹 members의 기본항목으로 구성되어 있다.