

1. 리눅스 운영 및 관리

(1) 파일시스템 관련 명령어

- 명령어[옵션] 파일...

- *파일의 접근권한 변경: **chmod** [옵션] [모드] [파일] / **chmod** [옵션] [8진수 모드] [파일]
(umask: 새로운 파일 생성시에 파일 접근권한을 조정하는 프로그램)
1000: sticky 비트를 적용
2000: 파일에 SetGID비트를 적용
4000: 파일에 SetUID비트를 적용
- *파일의 소유자변경: **chown** [옵션] [사용자계정] [파일]
/home/ihd 디렉터리를 포함한 하위 디렉터리의 모든 파일을 소유자는 wangdm로,
소유자가 속한 그룹은 wangdm로 바꾸기 위한 명령어: **chown -R wang.wang** /home/ihd
- *보통 디렉터리에 있는 파일들 중 다른 사용자와 파일을 공유하고자 할 때 쓰이는 명령어:
chgrp [옵션] 그룹파일 (chgrp 변경할그룹명, 변경대상그룹명)
- *사용자가 사용하고 있는 셸을 변경: **chsh** [-옵션] [사용자계정] /chsh [사용자계정] [-옵션]
(ch* 계열의 명령어는 수정에 관한 명령어)

- 파일시스템 관리명령

- rm** : 파일삭제
- ls-al** : 디렉터리와 파일정보 보기
- cp** : 하나 또는 그 이상의 파일이나 디렉터리 복사
- cd** : 작업중인 디렉터리의 위치변경
- df** : 파일 시스템 단위로 디스크의 사용량을 보여줌
- fdisk** : 디스크의 파티션 정보를 변경
- fsck** : 파일 시스템 점검
(파일시스템 관리명령이 아닌 것: cut, passwd, fg 등)

- 기타 명령어

- cat** : 어떤 파일을 읽어서 표준 출력으로 내보내는 것으로 표준 입력으로부터 내용을 읽는 명령어
- fsck** : 시스템을 점검하고자 할때 사용되는 명령어. 파일시스템의 무결성 검사
(파일 시스템의 유형을 지정할 때 -t옵션. -t옵션을 지정하지 않으면 /etc/fstab에서 유형참조)
*s옵션은 파일 시스템을 점검하기 전에 슈퍼블록을 나열하는 옵션,
*a옵션은 어떠한 질문도 하지 않고 파일 시스템에서 발견된 모든 문제를 자동으로 복구하는 옵션
*부팅 중 fsck 명령어를 이용하여 파일시스템을 점검하려면 /etc/fstab 파일을 이용한다.
*관리자인 root만 실행할 수 있다.
(점검하고자 하는 해당 장치를 마운트 한 상태에서 실행해야하는 건 아니다.)
- mkfs** : 리눅스 파일시스템 만드는 명령어
- fdisk** : 디스크 파티션 정보를 변경
- ifconfig** : 네트워크의 인터페이스 설정 또는 확인하는 명령어
- write** : 다른 사용자에게 메시지 보내기

- nslookup** : 특정 호스트를 찾기 위해 사용되는 명령어.호스트가 속한 네임서버에 질의하는 질의도구
- netstat** : 네트워크의 연결 상태를 확인하는 명령어
- find** : 원하는 파일 검색 명령어(시스템 내에 존재하는 파일에 관계없이 무엇이든 찾아주는 명령어)
- groupdel** : 그룹명: 그룹삭제 명령어
- unencode** : USENET과 같이 ASCII코드만을 다루는 미디어를 위해 바이너리 코드를 반환할 때 사용되는 명령
- quotacheck** : quota 설정파일인 "quota.user"파일을 최근의 상태로 갱신하는 명령어
- df** : 파일 시스템 단위로 디스크의 사용량을 보여줌
- du** : 파일이나 디렉터리의 사용량을 볼 수 있는 명령어
- man** : 도움말 보기
- whatis** : 간단한 도움말 보기
- fg, bg, &** : foreground/background 작업
- alias** : Alias(별칭) 만들기
- find, which** : 필요한 파일 찾기
- cmp** : 두 파일의 차이점 비교
- comm** : 두 파일의 차이점에 대한 더욱 상세한 정보 보기
- dircmo** : 두 디렉터리의 차이점 비교

- 리눅스 장치 조작 명령어

- restore** : 백업된 파일시스템을 복구한다.
- scanpci** : PCI 버스를 스캐닝하여 관련정보를 출력한다.
- umount** : 파일시스템을 언마운트한다.
- umask** : 파일 생성시에 소유자나 그룹의 퍼미션을 조정하기 위해 사용
*그룹이나 다른 사용자에게 대해서 쓰기 권한 방지를 위해서는 022로 설정한다.
*umask가 022로 설정된 상태에서 디렉터리를 만들면 755권한을 가진 디렉터리 생성
(022: --- -w- -w-) (755: rwx r-x r-x)
*프롬프트 상태에서 umask 입력 후 엔터키를 치면 현재 설정된 umask값을 볼 수 있다.
변경은 umask nnn 입력하면 바로 설정됨.
% umask **u=rwx,g=x,o=**
% umask **067**
숫자 값은 파일의 경우 666(666-022)에서, 디렉터리의 경우 777(-022),
umask 값은 '.cshrc'나 '.profile'에 등록시 login될 때 자동 설정됨.

- 파일조작명령어

- diff** : 파일사이의 다른 점을 찾아내는 명령어.
- more** : 주어진 파일의 내용을 화면 단위로 출력
- head** : 파일의 머리 부분을 보는데 사용
- tail** : 파일의 꼬리 부분을 보는데 사용
- file** :파일종류 확인
- more, page, less**
- grep** : 특정 문자열 찾기

tee : 화면으로 보면서 파일로 저장하기
wang]# ps -ef | tee ps
wc : 파일의 라인, 단어, 문자수 보기

- 기타 기술문제

- *inode : 파일 식별자. inode를 통해 리눅스 시스템 입장에서 볼 때 파일이라 하는 것을 구별함.
리눅스의 각각의 파일시스템에서 이름과 블록사이의 관계(mapping)를 다룸.
- *리눅스 파일시스템에서 디스크는 일반적으로 고정된 크기의 블록으로 관리한다.
(리눅스 파일 시스템은 간단하게 운영체제가 파일을 시스템의 디스크상에 구성하는 방식을 말한다.)
- *etc/issue: gretty에 의해서 로그인을 위한 프롬프트가 뜨기전에 출력되는 메시지를 설정하는 파일
- *chmod ugo=r soon: soon이라는 파일의 권한을 모두 read only로 변경
- *chgrp -R root dir A: dirA 디렉터리 내의 모든 파일의 그룹을 root로 변경
(-R옵션: 파일과 그 디렉터리 아래까지 변경하는 옵션)
- *리눅스 파일 시스템인 ext2와 ext3에서는 파일명은 연속적인 문자, 숫자, 특정 구두점의 단순한 열로 구성되며, 윈도우즈와는 다르게 파일명 내에 공백이나 필드 분리자를 포함할 수 없으며, 대소문자를 엄격하게 구분한다. 파일확장자에 대한 의미가 없으며, 파일이름은 최대 256자까지 가능하다.
- *ext2: 저널링 파일 시스템 이전에 가장 많이 사용했던 파일 시스템. 뛰어난 안정성과 속도가 장점.
쉽게 호환되고, 업그레이드가 용이함.
현재 ext2를 수정 저널링 파일시스템으로 만든 ext3가 기본 파일 시스템으로 사용되고 있다.
- *사용 중인 시스템에서 새 하드디스크를 추가하여 사용하기 위한 과정
:파티션생성-파일시스템생성-마운트

(2) Shell

셸의 종류

- * 본셸(Bourne Shell): 유닉스시스템 최초의 셸. 프로그래밍 언어의 용도로 개발됨.
본셸 계열의 셸로는 ksh, bash, zsh 등이 있다.
- * Bourne-Again SHell(bash셸): 표준 본셸에 기능을 보강한 것. 리눅스에서 가장 많이 사용되는 셸
POSIX와 호환됨. GNU프로젝트에 의해 만들어지고 배포됨.
-bash환경에서 사용자 실수로 홈 디렉터리에 있는 .bashrc파일을 삭제했을 경우,
/etc/skel 디렉터리 안에서 .bashrc 파일을 복사하고 소유권을 사용자에게 넘긴다.
- 일반적으로 bash의 문자완성 기능 이용 시 사용하는 키: Tab
- * csh : C언어와 유사하여 유닉스에서 많이 사용됨. 본셸과 대부분 호환되지만 다른 인터페이스 가짐.
편집기능 제공안함.
/bin/csh: C셸은 C언어와 유사한 언어를 사용한 대화형 셸이다.
- * C Shell : csh보다 진보된 셸. 대화형 셸. 입력된 명령을 쉽게 수정할 수 있으며,
많은 태스크들을 백그라운드 작업제어를 할 수 있다. 셸 스크립트로 본셸사용.
- set명령어로 지역 셸 변수를 설정하며, setenv 명령어로 환경변수를 설정한다.
- 문자열, 수, 논리식을 구할 수 있는 것 외에 비트 단위 논리식을 구할 수 있다.
- 숫자 변수 할당과 문자열 변수 할당은 서로 다르게 선언되어진다.
- * 콘셸(Korn Shell,ksh) : 유닉스에서 가장 많이 사용됨. 명령행편집기능 제공.C셸통.
- 본셸과 호환되면서 명령행 편집기능을 제공한다.

- * TC Shell(tsh) : 확장 C셸. 명령행 편집기능 제공.
- * 셸 실행시 초기화 파일이름 : 콘셸(.kshrc), 본셸(.bashrc), TC셸(.tcshrc)
- * 명령행 편집기능 제공 : bash,ksh
- * 사용자 셸이 bash로 설정된 경우 로그인할 때 실행되는 파일: .bashrc와 .bash_profile

본셸 환경변수

- * 본셸은 산술계산, 조건식, 제어구조, 내장 명령어들을 가진 프로그램을 작성 가능케 하는 셸인 동시에 모든 셸의 기본이 된다.

- * 내장 셸 변수는 4가지 변수 중의 하나이다.
- * 특수 파라미터는 4가지 변수 중의 하나이다.
- * 사용자 정의 변수는 4가지 변수 중의 하나이다.

HOME=/home/login : 사용자가 시작하는 위치인 홈 디렉터리 설정
LOGNAME=login: LOGNAME은 자동으로 로그인 ID와 동일하게 설정된다.
PATH=path: path옵션은 셸이 명령에 대해 찾는 디렉터리 목록을 표현한다.
(PATH 환경변수에 /usr/local/bin이라는 신규 값 추가하는 명령어:

export PATH=\$PATH:/usr/local/bin)

PS1=prompt: PS1은 프롬프트의 모습을 정의하는 1차 셸 프롬프트이다.

PWD=directory: 자동으로 설정된 것으로 사용자의 현재위치(디렉터리경로)를 나타내주는 역할을 한다.

SHELL=shell : SHELL은 사용자 셸로서 가능한 프로그램의 위치를 식별한다.

TERM=temtype: 터미널 데이터베이스에 의해 지정되는 대로 터미널 유형의 이름을 설정함.

특수 환경변수

HOME: 사용자가 시작하는 위치인 홈디렉터리를 설정함. 항상 홈 디렉터리를 지정한다.

홈디렉터리가 이동되는 경우에도 명령이 여전히 작동된다.

홈디렉터리의 파일을 지정하는 셸 스크립트를 작성 중일 때 HOME변수를 사용할 수 있다.

SHELL: 현재 셸의 이름

MAIL: 우편함의 전체경로명

LOGNAME: 로그인 이름

PATH: 셸이 명령을 탐색하는 디렉터리를 지정한다. 디렉터리가 나열되는 순서로 탐색한다.
순서대로 디렉터리를 탐색하다 명령을 찾으면 탐색을 중단한다.

TZ: date명령에 대한 시간대

SECONDS: 셸을 호출한 이후의 시간(초)

PS1: 시스템 프롬프트

TERM: 사용중인 터미널의 유형

HISTFILE: 사용자가 실행했던 명령어의 목록을 저장하는 파일을 설정한다.

HOSTNME: 호스트이름

리다이렉션

*명령의 결과,출력(입력)을 다른 명령의 입력(출력)값으로 사용.

*명령어 > 파일: 명령의 출력을 파일로 보냄

[ex] 명령1: cat a>b

명령2: cat b

=> 파일a의 내용이 출력됨.

*명령어 < 파일: 파일에서 명령어 입력

*명령어 >> 파일: 명령의 출력을 파일에 덧붙임.

[ex] ls -l >>aa.log

=> aa.log 파일이 존재한다면 ls -l의 결과를 파일 끝부분에 추가하고

존재하지 않는다면, 생성한 다음 ls -l의 결과를 기록한다.

리눅스 쉘의 환경파일

.login: 특정 쉘에 관계없는 초기 환경을 지정하여 Login할 때 지정한 명령을 자동수행

.logout: 시스템 접속을 종료하기 전에 마지막으로 수행될 내용을 지정하는 파일

.newsrsc: USENET News Group 이용시 그룹 환경 저장 파일

.eml: 전자우편 프로그램에서 사용하는 디렉터리

기타 기술문제

*셸 스크립트를 작성하는 이유

- vi나 ed같은 편집기를 사용하여 셸 스크립트를 간단히 작성하고 편집할 수 있다.

- 스크립트의 생성과 유지가 쉽다.

- 스크립트는 키를 타이핑하여 입력해야 하는 노력과 시간이 절약되며 의사결정, 루핑, 에러조작 등의 명령어를 이용하여 폭넓게 문제를 해결할 수 있다.

- 셸스크립트: 본셸의 기능으로 가능한 것이다.(본셸은 셸스크립트만가능, Bash는 모두 가능)

- 셸 스크립트가 잠시 운영을 중단하는 명령어: sleep

- hello.sh라는 셸 스크립트 파일 내용

#!/bin/sh

comment lines start with the # character

echo "리눅스의 세계에 오신 것을 환영합니다."

- echo \$SHELL 명령의 실행결과 /bin/bash가 출력된다면

: 현재 기본 사용 셸이 /bin/bash임을 나타낸다.

*alias: 하나의 명령 또는 명령의 집합에 붙여 주는 어떤 이름을 alias라 한다.

자주 사용되는 복잡한 명령을 alias를 사용하여 단축 명령어로 만들어 둘 수 있으므로

alias 기능을 잘 이용하면 UNIX 시스템을 편리하게 사용할 수 있다.

- alias의 사용서식은 alias별명 = '명령어'와 같다.

- .cshrc까지 사용하는 사용자라면 .cshrc파일을 작성하여야 한다.

- rc파일은 셸이 실행될 때마다 적용될 수 있는 특정 쉘에 관계된 변수와 설정 사항을 포함

- 파일 삭제시 확인 메시지 표시하도록 설정하는 alias 명령어: alias rm = 'rm -i'
(홈디렉터리에 있는 .bashrc파일에 설정이 이루어진다.)

*기타명령어들

- w: 시스템 사용자들의 작업내용 출력

- talk: 다른 사람에게 간단한 메시지 전달

- traceroute: 다른 시스템에 접속되는 경로 조회

- who: 시스템 사용자들이 접속하여 들어온 장소, 시간 등 출력

- rm: 삭제하는 명령어

[예] rm client.9 [0~3,a~z] 명령 실행시 패턴중 (0~3,a~z)를 포함하는 파일에 영향을 줌.

*.cshrc: Cshell이용자의 경우 로그인환경설정시 필요한 파일.

*어떤 셸이 다른 셸을 시작시킬 경우 다른 셸을 sub셸이라고 부른다.

*셸의 일부분을 제외한 모든 유틸리티 프로그램 및 기타 응용 프로그램들이 C언어로 쓰여짐.

*파이프: 여러 명령을 조합해서 사용할 때 이용. 하나의 명령행에 여러 개의 파이프 사용가능.

*!: 히스토리 기능을 사용하기 위한 셸 예약어

- Bash셸의 히스토리 기능을 이용하여 바로 이전 명령의 매개변수를 재입력하고자 할 때 사용하는 셸예약어: !!

#!877 877번 명령 실행

#!! 최근명령

#!-n 마지막에서 역순으로 "숫자"번째 명령 실행.

#!string string으로 시작하는 최근명령

* 셸의 히스토리 명령어는 **history** 이다. -> 계정/.bash_history 출력

*stty: 터미널 세팅을 보여주거나 변경해 줌. 스크립트에서 터미널 동작이나 출력하는 방법을 제어.

명령행 편집기능 즉 키보드를 이용하여 많이 사용하는 유용한 키(Ctrl -c, Backspace 등)를 설정함.

(3) 프로세스 관리

*priority: 프로세스가 들어온 프로세스의 우선순위를 보여줌.

*nice: 우선순위 설정. 기본적으로 아무런 옵션없이 nice를 사용하면 상속받은 현재 순서의 우선권을 출력 수치가 낮을수록 높은 우선순위를 갖게된다.

(renice: 이미 실행된 프로세스의 우선순위를 바꿔주는 명령어)

*ps: 프로세스 상태를 알기 위한 명령어

#ps u: 사용자와 프로세스번호(PID)등이 출력됨

r옵션: 현재 실행상태에 있는 프로세스만 보고자 할 경우 사용.

-u옵션: 사용자의 이름과 프로세스가 시작된 시간 출력.

지정한 사용자가 실행한 프로세스만 보고싶을 경우 사용하는 옵션.

-j옵션: 작업에 관련된 ID를 출력한다.

-a옵션: 모든 프로세스 나열

-w옵션: 출력결과를 생략하지 않고 출력

-f옵션: 프로세스 간 상관관계를 트리구조로 나타냄.

(PID: 프로세스마다 주어지는 번호, RSS:프로세스가 사용하는 메모리의 크기

WCHAN: 프로세스가 실행하고 있는 커널루틴, STAT:실행되고 있는 프로세스 상태)

*STAT(R:실행 중 혹은 실행될 수 있는 상태, S:sleep, l:idle 비활동상태 T:정지된 상태.서스펜드)

*fork: 리눅스에서 프로세스를 복제하는 기능.

사용자가 새로운 프로세스를 생성하는 유일한 방법.

*fsck: 예상치 못한 시스템 종료시 파일시스템의 불일치성을 체크

(-v 옵션: 명령이 실행되는 과정을 화면으로 출력한다.)

*uname: 시스템 이름

-a : 시스템 이름, 도메인, 버전, 등 출력

*shutdown -r now: reboot 명령과 같음 [예] shutdown -r +5 : 5분후에 시스템 재부팅

*pstree: 프로세스의 상관관계를 보고자 할때

*jobs: 백그라운드로 수행중인 작업확인

*top: 시스템의 프로세스 수, 사용중인 데몬, CPU프로세스의 상황 등을 보여줌.

*cron프로그램은 시스템이 부트될 때 한번만 시작된다. 개별 사용자는 cron을 직접 실행할 권한이 없다. 또한 시스템 관리자는 명령의 이름을 입력하여 cron을 실행해서는 안된다.

[예] 5 10 6 5 *echo Happy birthday : 5월 6일 10시 5분에 echo Happy birthday를 실행한다.

[예] *****echo Excuse me !!One Minute

>dev/tty1

: Excuse me !!One Minute을 매 분 터미널 장치 dev/tty1에 보낸다.

(분 시 일 달 요일 순서)

* /etc/cron.d/cron.allow - 등록된 User만 crontab 사용을 허가함.

* /etc/cron.d/cron.deny - 등록된 User만 crontab 사용을 금함.

#/etc/init.d/sh cron stop : cron중지

#/etc/init.d/sh cron start : cron시작

*nohup: 부모프로세스가 죽거나 종료되어도 자식프로세스는 백그라운드 모드에게 계속 실행하도록 함.

프로세스 특성

*프로세스 관리 블록에 저장되는 프로세스 관련 정보들이 유지하는 정보

:프로세서가 할당받은 자원들의 리스트 및 이에 대한 포인터. 프로세스 고유번호(PID)

*하나의 프로그램을 실행시키면 하나 이상의 프로세스가 생긴다.

*시스템 호출은 핵심으로 하여금 프로그램에 일부 서비스를 제공하도록 하는 서브루틴이다.

*프로세스는 실행 중인 프로그램이라고 할 수 있다. 커널에 등록되어 커널의 관리하에 있는 작업으로 이를 일반적으로 주기억 장치에서 실행중인 프로그램.

*리눅스 시스템에서 명령을 타이프하면 그 쉘은 명령의 실행을 관리한다.

*프로세스를 할당받은 상태, 즉 기억장치를 할당받은 상태: 프로세스의 실행상태일 때의 특성.

*프로세스를 요청하는 상태: 기억장치를 잃은 상태.

*많은 양의 CPU시간을 이용하는 프로세스들은 낮은 우선권을 갖는다.

*가장 낮은 수준에서 프로세스는 fork 시스템 호출에 의해 창출된다.

기타

*임계영역 밖의 프로세서가 다른 프로세서의 임계영역 진입을 막아서는 안 된다.

*파이프: 유닉스 환경 하에서 프로세스 간 통신의 고전적인 방법.

*signal: 프로세스가 다른 프로세스에게 메시지를 보내기 위한 수단(프로세스간 통신수단)

HUP-1: Hangup (실행종료, 로그아웃하거나 모뎀접속 끊을때 발생)

KILL-9: kill(프로세스 강제종료)

SEGV -11: Segment Violation (허가되지 않은 메모리 영역 접근)

TERM -15: Termination(가능한 한 정상종료)

INIT -2: Interrupt(실행종료,Ctrl-c키를 누르면 발생)

TSTP -20: STOP(실행정지 후 다시 실행위해 대기, Ctrl-Z키 누르면 발생)

*시스템모드: 시스템 함수가 초기화될때 프로세스가 선택하는 모드

*대기: 프로세스들이 외부적인 사건을 기다리는 것.(S)

*exec 시스템호출: 프로세스가 지정된 파일을 실행할 수 있도록 해주는 것.

(exec: 하나의 프로세스가 다른 프로세스를 생성하여 원래 프로세스가 더이상 남아있을 필요 없을때)

*Swapping: 프로세스 관리 및 메모리 관리와 관련있음.

*init: 리눅스에서 실행되는 최초의 프로세스.

init가 실행되면 /etc/inittab 파일을 읽는데, 이 파일의 설정에 따라 프로세스들을 실행시킨다.

*좀비프로세스: 실행이 끝났는데도 종료되지 못하고 있는 프로세스.(z)

프로세서 테이블에 각각의 프로세서 아이디와 사용하는 자원을 올려놓는다.

[ex]운영중인 서버의 프로세스 확인과정에서 프로세스는 종료되었으나 부모프로세스는

해당 프로세스가 종료되었음을 인지하지 못하고 defunct로 표현되어 있음.

*프로세스 교착상태: 두 개 이상의 프로세서가 동시에 서로 사용하는 자원을 요구하려 할 때 발생 (자원에 대한 배타적 사용이 원인)

*프로세스 스케줄링: 프로세스의 시스템을 향상시키기 위한 목적

*데몬: 프로세스들 중에는 특별한 이름을 가진 것들이 있음. 데몬 프로세스는 백그라운드로 실행되면서 서버역할을 하거나 그 기능을 도와주는 프로세스를 가리키는 말이다.

(독립적으로 메모리에 계속 상주해있으면서, 서비스를 해주는 방식: standalone 방식)

- crond: 사용자가 지정한 프로그램을 특정 시간에 주기적으로 실행시켜 주는 표준 유닉스 데몬

- inet방식: 데몬을 운영하는 방식 중 하나로, 사용자의 요구가 빈번하지 않은 서비스들을 중심으로 이루어지며, TCP Wrapper를 이용하여 해당서비스 요청에 대한 로그기록

-inetd: 주로 인터넷 서비스에 이용되는 여러개의 데몬을 관리하는 슈퍼데몬의 이름.

*dev 디렉터리: 파일시스템중에서 주변장치 파일이 저장되어 있는 디렉터리

*리눅스에서 백그라운드 작업을 위해서는 실행 시 프로그램 이름뒤에 &를 붙인다.

*포그라운드로 실행되는 프로그램을 Ctrl -Z를 누르면 서스펜드된다.

*백그라운드를 포그라운드 전환하려면 fg, 포그라운드를 백그라운드로 전환하려면 bg

*RunLevel 1: 단일 사용자 모드 / RunLevel 3: 풀 멀티유저 모드

RunLevel 5: GUI환경으로의 부팅모드 / RunLevel 6: 재부팅모드

2. 리눅스 활용

(2) 인터넷 활용

컴퓨터를 통신망에 의하여 상호연결하여 S/W나 데이터베이스를 공유하도록 함.

컴퓨터 네트워크의 범위는 크게 일정지역 내의 통신을 위한 LAN과 거리에 제한이 없는 WAN으로 분류됨.

*LAN : Local Area Network

- 건물이나 공장, 학교 등 제한된 일정 지역 내에 분산 설치된 각종 컴퓨터와 주변기기를 사이의

통신을 수행하기 위하여 신뢰성 있는 고속의 통신채널을 제공하는 근거리 통신망

- 10Mbps~155Mbps의 전송속도로 데이터를 송수신

- 지역적으로 제한된 영역에 한정

- 낮은 오류율에 의한 신뢰성 있는 고속데이터 전송이 가능
- *WAN: Wide Area Network
- 넓은 지역에 분산된 통신기기들을 연결한 네트워크
- LAN 보다 통신 속도가 느림
- 전용 디지털 통신망으로 구성되며 공중교환전화망(PSTN)을 사용하기도 함.

TCP/IP

*개방형 프로토콜의 표준

*이더넷, 토큰 링, 전화선, 그리고 가상적으로 어떤 다른 종류의 물리적 전송 매체에서도 실행가능

*일관성있고 널리 사용가능한 사용자 서비스를 위해 표준화된 하이레벨의 프로토콜.

*TCP는 데이터의 흐름을 관리하고 데이터가 정확한지 확인하는 역할을, IP는 인터넷에서 사용하는 전송 프로토콜로서 주어진 패킷을 어떻게 목적지까지 보낼 것인가에 대한 역할을 담당. IP는 인터넷을 구성하는 핵심요소로 인터넷 전송기본단위인 Datagram정의, 인터넷주소체계정의 등의 기능을 한다.

* TCP/IP 내부 스택-프로토콜

| 계층도 | 기 능 | 장 비 | 프 로 토 콜 |
|--------------|--------------------------------|----------------|---|
| Application | 응용소프트웨어의 프로토콜 | 게이트웨이 | FTP, Telnet, SMTP, DNS, TFTP, SNMP, RIP |
| Presentation | 구조를 가진 데이터의 입력, 표시제어 | | |
| Session | 논리 접속의 설정, 결합, 해방 | | |
| Transport | 엔드 시스템간의 확실한 데이터 전송 관리 | | TCP, UDP, SPX |
| Network | 네트워크간의 데이터 전송, 에러제어 | 라우터 | IP, IPX, IGMP, ICMP, ARP, RARP |
| Data Link | 인접 노드의 데이터 전송, 에러제어 | 스위치,브리지 | |
| Physical | 물리적인 데이터 전송의 제어 비트 레벨의 물리전송 | 케이블,허브, 리피터 | |

☞ ICMP(Internet Control Message Protocol) : IP 데이터그램과 함께 에러,제어메시지 전송

☞ IGMP(Internet Group Management Protocol) : 인터넷 그룹 관리 프로토콜

☞ ARP(Address Resoltion Protocol) : IP → MAC

☞ RARP(Reverse ARP) : MAC → IP ; DHCP(RARP를 다른형태로 구현

*계층모델의 응용프로그램 계층

- Telnet: 네트워크를 통한 원격 로그인을 제공
- FTP: 양방향 파일전송
- HTTP: 웹페이지 전송, www 서비스에서 주로 사용하는 프로토콜
Command, Header, Body로 구성된 데이터를 주고받는다.
쿠키를 사용하여 세션(정보를 유지한다.)

하나의 명령마다 독립적으로 연결이 허용된다.(연결 유지된 상태에서 여러번의 명령수행 불가능)

*HTTP 프로토콜에서 데이터를 전송하는 방식에는 POST방식과 GET방식이 있다.

- SMTP: 전자메일전달
- DNS: IP주소를 네트워크 장치에 할당된 이름과 대응시켜줌.
문자로 구성된 도메인네임을 숫자로 구성된 IP주소로 변환한다.
리눅스에서는 버클리인터넷도메인(BIND)등이 사용된다.
인터넷을 사용하기 위해서는 DNS설정을 반드시 해야한다.
- NFS: 네트워크상의 다양한 호스트들이 파일을 공유할 수 있도록 해줌.

IP 주소

*IP주소는 TCP/IP네트워크 상의 통신에서 각 호스트를 유일하게 인식시키기 위해 필요한 것.

(이렇게 인식된 호스트상의 응용프로그램이나 서비스를 구별하기 위해 사용하는 것은 포트번호)

(소켓: IP주소와 포트번호의 조합으로 전체인터넷에서 하나의 네트워크 프로세스를 단일하게 표현)

* A형: 0-127 / B형: 128-191/ C형: 192-223

* 몇 개의 비트가 네트워크를 식별하는데 사용되고, 몇 개의 비트가 호스트를 식별하는데 사용되는지를 나타내기 위해서 서브넷 마스크가 지정된다.

- 클래스 A: 255.0.0.0
- 클래스 B: 255.255.0.0
- 클래스 C: 255.255.255.0

[ex] 서브넷 마스크 255.255.0.0사용. 게이트웨이의 IP주소 211.100.0.1사용.

이 회사가 사용할수 없는 IP주소는? 211.10.100.100

(211.100.100.10, 211.100.100.100, 211.100.10.100은 사용가능)

*IP: 데이터그램과 인터넷 주소체계 정의하는 프로토콜

OSI

* 7개 계층으로 구분.

(물리계층, 데이터링크계층, 네트워크계층, 트랜스포트계층, 세션계층, 표현계층, 응용계층)

*트랜스포트계층(전송계층): 트랜스포트 계층이 가장 중요한 두 프로토콜은 TCP와 UDP이다.

TCP는 end-to-end 에러 탐지와 수정기능을 가진 안정성 있는 데이터 전달 서비스를 제공한다.

- 접속기반, 바이트 스트림 프로토콜. 안정적인 전송을 제공하기 위한 애플리케이션에서 사용.

UDP는 부하가 걸리지 않으며, connectionless 데이터그램 전달 서비스를 제공한다.

- 불안정한 비접속 데이터 프로토콜. 전송되는 데이터 양이 적다면, 접속을 생성하고 전달의 안정성을 보장하는 오버헤드가 전체 데이터를 다시 전송하는 작업보다 더 클 것이다.

이 경우에 UDP는 트랜스포트 계층 프로토콜을 위한 가장 효율적인 선택이다.

LAN 통합을 위한 통신장비

* 라우터(ROUTER): LAN(근거리통신망)을 연결해주는 장치로서 정보에 담긴 수신처 주소를 읽고 가장 적절한 통신통로를 이용하여 다른 통신망으로 전송하는 장치

* 리피터(REPEATER): 디지털 방식의 통신선로에서 전송신호를 재생하여 전달하는 전자통신장치

* 브리지(BRIDGE): 두 개의 근거리통신망(LAN)을 상호 접속해 주는 통신망 연결 장치

* 게이트웨이: 프로토콜이 다른 통신망을 상호 접속하기 하기 위한 장치.

LAN을 연결해주는 장치로서 정보에 담긴 수신처 주소를 알고 가장 적절한 통신통로를 이용하여 다른 통신망으로 전송하는 장치.
디폴트 게이트웨이: 특별히 지정되지 않은 네트워크로 보내지는 패킷들이 먼저 경유함.

네트워크 관련 용어설명

*ICMP: 호스트간 제어 메시지 교환 프로토콜

IP프로토콜은 전달을 보장하지 않는다. 그래서 전송상의 오류 등을 보고하고 위해 ping같은 프로그램에서는 ICMP를 사용한다.

*FTP: 파일전송 프로토콜

- hash: 전송할 때 # 문자를 진행도에 따라 표시할지 여부를 결정하는 FTP 명령어.
- get: 기본 FTP 클라이언트를 통해 파일을 다운받기 위한 명령어.(업로드: put)
- anonymous FTP: 비밀번호를 입력하라는 메시지에는 자신의 전자우편 주소를 입력한다.
(Archie: 인터넷 초창기 활발히 사용되던 서비스. 자신이 원하는 파일이나 문서가 어느 anonymous FTP서버에서 제공되는지를 찾아주는 프로그램)

*NFS: 네트워크 파일 시스템

*NIC: 네트워크 정보센터, 인터넷/IP, 도메인 관리

*IMAP: 전자우편 수신하고 보관하는데 사용되는 프로토콜

*NetBIOS: 응용프로그램들이 LAN상에서 서로 통신할 수 있게 해주는 프로토콜

*SSL: 네트워크 내에서 메시지 전송의 안전을 관리하기 위해 넷스케이프에서 만들어진 프로그램 계층
(네트워크 관리하는 프로토콜 아님!!, 네트워크 레이어임.)

*nslookup :도메인 네임 서비스에 관련된 설정을 출력(DNS 네임 서비스 관련 정보들을 확인)

*route, ifconfig: TCP/IP 네트워크를 설정하는 명령어

*ping: 원격의 호스트의 연결을 확인하는 명령어

*traceroute: 특정 호스트까지 연결된 노드를 알아보는 명령어

*라우팅테이블: 라우터나 기타 다른 인터네트워킹 장치에 저장되어 있는 데이터베이스로서,
네트워크상의 특정 목적지까지의 경로를 기억하기 위해 사용된다.

*TELNET 서버에 접속하기 위한 기본형식: telnet<hostname | ip_address>[port]
- "<"와 ">"은 입력하지 않고, "|"은 or을 뜻하며, "["은 옵션부분이다.
- Telnet이용하여 ihd사이트에 접속하고자 한다면 open명령사용.
open telnet.ihd.or.kr

*루프백 인터페이스에 부여되는 IP주소: 127.0.0.1

- 루프백 인터페이스는 자신에게 연결할 수 있도록 해주는 특별한 인터페이스이다.
- 편의상 '127.0.0.1'의 IP주소가 루프백용으로 할당되어 왔다.
따라서 어떤 머신으로 가더라도 127.0.0.1로 telnet연결을 건다면 localhost로 접속될 것이다.
- 리눅스에서 지원되는 네트워크 인터페이스의 일종으로 ifconfig로 설정, 확인가능.

*IP주소의 유형에 속하는 것: 유니캐스트(단일 송신자와 단일 수신자 사이의 데이터 전송방식),
멀티캐스트(그룹에 속한 사용자 모두에 전송),
브로드캐스트(같은 도메인에 속한 모든 사용자에게 전송)

*IP가 들어온 데이터를 Transport 프로토콜로 넘겨준 후에 Transport 프로토콜은 적당한 애플리케이션 프로세스로 그 데이터를 넘긴다. 애플리케이션 프로세스는 16비트 값이 포트번호에 의해 구분된다.

*네임서버는 Primary, Secondary, Cache only server로 구분됨.

*POP3: 이메일 받을 때 사용되는 가장 일반적인 프로토콜

(이메일을 보내는데 사용되는 서비스는 SMTP이고 서버에 저장된 메일을 사용자 컴퓨터의 메일 클라이언트 프로그램에 전달해주는 역할을 하는 서비스는 POP3이다.)

*인터넷 도메인 이름은 하이픈으로 끝날 수 없고, 특수문자 사용할 수 없다. (2자에서 63자까지 가능)

*모질라 웹브라우저 메뉴바의 파일항목: 인쇄, 페이지설정, 오프라인작업(기본설정은 편집항목에 있음)

*ifconfig: 네트워크 인터페이스 설정하거나 확인하는 명령어.(netconfig, netcfg)

네트워크 인터페이스 정보확인, 활성화, 비활성화, 네트워크 설정.

*/etc/protocols: 네트워크 인터페이스 설정하거나 확인하는 명령어. 프로토콜 번호가 정의되어 있음.

*/etc/services: 리눅스에서 기본적인 서비스에서 사용되는 잘 알려진 포트번호 확인.

*/etc/resolv.conf: 기본 DNS 서버를 변경하고자 할때 수정할 파일

*/etc/sysconfig/network: 레드햇 계열의 리눅스에서 기본 게이트웨이 주소확인.

*UUCP: dial-up 전화 링크에 기반한 광역 네트워크에 응용된다.

시리얼 라인을 통하여 파일을 교환하는 것에서 출발하였다. 유닉스 개발사이트와 교신위해 개발.

시리얼 라인을 사용하기 때문에 고속의 전송이 어렵다.

*SLIP: TCP/IP 네트워크에 다이얼업 접속을 할 수 있도록 해주는 데이터링크 프로토콜이다.

*HTTP: www 서비스에서 주로 사용하는 프로토콜

Command, Header, Body로 구성된 데이터를 주고받는다.

쿠키를 사용하여 세션(정보를 유지한다.)

하나의 명령마다 독립적으로 연결이 허용된다.(연결 유지된 상태에서 여러번의 명령수행 불가능)

*일반적으로 하나의 클래스 C 네트워크를 두 개의 네트워크로 구성할 경우
서브넷마스크는 255.255.255.128이 된다.

*서비스이름과 포트번호:

ftp-21(20), telnet-23, http-80, https-443, gopher-70, smtp-25, pop2-109, pop3-110
snmp-161, ipx-213, ssh-22, rlogin-513, nntp-119, name server(42), tftp-69, wins-1512
mysql-1433(1434), daytime-13, swat-901

*IRC: 채팅이 가능하게 하는 서비스

*라우팅의 종류: Dynamic라우팅, Default라우팅, Static라우팅
(종류가 다른 것들: Control라우팅, Bridge라우팅...)

*finger: 접속한 PC뿐만 아니라 원격의 PC사용자에 대한 정보를 해당 서버에 요청하여
수집할 수 있는 프로그램.

*삼바: 삼바는 MS에서 개발한 smb 프로토콜을 사용해서 통신한다.

윈도우즈 계열 운영체제뿐 아니라 삼바가 설치되어 있는 유닉스 계열 OS와도 통신이 가능함.
리눅스 시스템 내의 파일을 윈도우즈 계열 운영체제에서 탐색기에서 편집 및 수정할 수 있다.

* 네트워크 인터페이스 설정시 기본적으로 IP주소, 서브넷 마스크, 게이트웨이,
도메인네임서버 주소가 필요하다. 방화벽 소프트웨어 사용시에는 맥어드레스를 참조하기도 한다.

*www: 하이퍼텍스트 기능을 사용하여 문서를 참조한다.

*이더넷카드: 랜을 이용하여 인터넷에 연결된 컴퓨터에 연결한다.

*-i옵션으로 몇 초 간격으로 패킷을 보낼 건지 결정한다. TTL은 Time To Live를 말한다.

*리눅스를 활용한 고속인터넷 접속방식

- 케이블 TV망: 데이터가 오가는 통로로 전화선 대신 케이블TV망을 이용하고 다이얼업 모뎀의
역할은 케이블 모뎀이 담당한다.

Cell 내의 접속자가 증가하면 최대 전송속도가 감소한다.

- ADSL: DSL전용 모뎀으로 인터넷에 연결됨. 업로드 속도와 다운로드 속도가 다른 디지털 회선.

*프록시서버: 인터넷 액세스 속도를 증가시켜 주는 것으로 게이트웨이나 방화벽에서 활용가능한 시스템.

*32비트 사용하는 IP 방식에서 클래스 B를 배정받은 통신망은 65,534개의 호스트 구성가능.

*DHCP: IP주소를 효율적으로 사용하기 위해 중요한 네트워크 파라미터 및 설정사항들을 자동 설정.

tar <옵션> <만들 파일이름> <읽을 파일이름>

-c tar 파일을 만들때 사용
-x 묶여진 tar 파일을 해체할 때 사용
-v 파일들을 묶거나 해체할 때 파일들의 이름과 크기를 표시
-f 사용할 tar 파일을 지정
-t 묶여진 tar 파일의 내용물을 출력

tar -xvf example.tar

example.tar의 파일 해체, 해체되는 파일 표시

tar -cvf backup.tar /home/ftp/pub

/home/ftp/pub 디렉터리와 그 안의 파일들을 backup.tar 로 묶음.

tar -tvf example.tar

example.tar 안에 파일들의 리스트 표시

gzip <옵션> <파일이름>

gzip cvs.pdf : 압축하기

gzip -d cvs.pdf.gz 또는 gunzip cvs.pdf.gz : 압축풀기

-l 압축 파일 정보 출력
-1 빠른 압축 낮은 압축률
-9 느린 압축 높은 압축률

bzip2 은 gzip 보다 10%-20% 정도 압축률 뛰어남 (.bz2 확장자)

사용법 : bzip2 <옵션> <파일이름>

압축하기 : bzip2 cvs.pdf

압축풀기 : bzip2 -d cvs.pdf.bz2 또는 bunzip2 cvs.pdf.bz2

tar 와 gzip, bzip2 같이 사용하기

◆ tar로 묶어 gzip으로 압축된 파일 풀기

tar xvfz test.tar.gz

tar 로 묶고 bzip2로 압축된 파일은 tar.bz2 와 같은 확장자 형식을 가진다.

tar xvfj test.tar.bz2

◆ tar로 묶고 gzip으로 압축해 보기

tar cvfz example.tar.gz example_dir

bzip2로 압축하고 싶다면, z 옵션 대신에 j 옵션(tar-1.13 이하는 l -대문자 i)을 사용하여 bzip2 가 수행되도록 한다.

tar cvfj example.tar.bz2 example_dir

rpm -ivh 일반 설치
-Uvh 의존성 설치(업그레이드)
-qa | grep 데몬 설치 확인
-qi 자세한 정보
-e 삭제
--nodeps 의존성 무시
--force 강제 설치
-V 검증

dpkg --install *.deb 설치
--purge 삭제
--search 찾기

/etc/passwd 계정 정보
/etc/shadows 계정 패스워드
/etc/group 그룹 정보
/etc/inittab 부팅 레벨
/etc/fstab 부팅 마운트 정보
/etc/grub.conf 부트로더 설정
/etc/lilo.conf 부트로더 설정
/etc/skel 계정생성 참조 디렉터리
/etc/sysconfig/network 기본 게이트웨이
/etc/sysconfig/network-scripts/ifcfg-eth0 이더넷 IP 설정
/etc/resolv.conf DNS 설정
/etc/protocols 프로토콜 번호
/etc/services 소켓 번호
/etc/printcap 프린터 정보
/etc/termcap 터미널 정보
/etc/modules.conf 로딩된 모듈 정보
/usr/X11R6/man X 의 man 페이지
/etc/sysconfig/desktop Xclients 스크립트가 불러내는 파일
/etc/gtk/gtkrc.ko GNOME 폰트 설정
~/.bashrc 계정별 설정 파일

lsmod 모듈 정보 리스팅
insmod 모듈 로딩
rmmod 모듈 삭제
modprobe 모듈 로딩(의존성 참조)

insmod와 modprobe 의 차이

insmod : '/lib/modules/커널버전' 의 디렉터리에 해당 모듈이 있으면 로딩.

modprobe : depmod에 의해 생성된 modules.dep 에서 찾아 모듈 로딩.

또한 해당 모듈에 의존성이나 해당모듈보다 선수될 모듈이 있으면 그 모듈도 같이 로딩.

예) vfat이란 모듈을 올리기 위해선 fat이란 모듈이 먼저 올라와 있어야 하는데

fat이 올라와 있지 않은상태에서

insmod vfat 하면 fat에 관한 심벌을 찾지 못해 vfat 모듈을 올리지 못함.

modprobe vfat 을 실행하면 자동으로 fat을 먼저 로딩 그다음 vfat 모듈 로딩.

modprobe로 해당 모듈을 찾지 못 하는 경우 modules.dep에 없을 가능성이 크므로

depmod -a 해서 새롭게 갱신.

\$\$의존관계에 있는 모듈들을 한꺼번에 내릴려면 rmmod -r 옵션 사용.

X 설정 : XFree86, Xorg, XF86Setup, Xconfigurator

/etc/lilo.conf

boot=/dev/hda ; LILO 설치 위치

MBRappend="mem=128M" ;커널 옵션

map=/boot/mapinstall=/boot/boot.bprompt ;LILO 프롬프트가 나오도록 한다.

timeout=50 ;대기시간 5초

default=d ;기본 부트 운영체제의 label을 적는다

image=/boot/vmlinuz-2.2.5 ;리눅스 커널의 이미지 위치

label=; 등록하고자 하는 이름

root=/dev/hda3 ;루트 파티션 장치명

read-only ;리눅스에만 해당된 내용

others=/dev/hda1 ;도스 파티션 label=d

table=/dev/hda ;hda 또는 sda

LILO 설치

lilo

adding ladding d* ;'*' 표시는 기본 부트 운영체제임을 뜻한다.

lilo -D d ;기본 부트 운영체제 설정

MBR에서 LILO 제거하기

- 도스에서 c:>fdisk /mbr- 리눅스에서 # /sbin/lilo -u /dev/hda

/etc/grub.conf

default=0

timeout=10

splashimage=(hd0,1)/boot/grub/splash.xpm.gz

title Red Hat Linux (2.4.18-3)

root (hd0,1)

kernel /boot/vmlinuz-2.4.18-3 ro root=/dev/hda2

initrd /boot/initrd-2.4.18-3.img

title Windows 2000

rootnoverify (hd0,0) #또는root (hd0,0)---->반드시 root 다음에 띄워쓰기

makeactive

chainloader +1

Grub 삭제

[root@wangsam root]# grub

grub> uninstall

Grub 설치

[root@wangsam root]#rpm -Uvh grub.2.2.2.i386.rpm

[root@wangsam root]#grub inatall

[root@wangsam root]#reboot

Grub 복구

1. 시디 부팅후 boot: linux rescue(복구콘솔)

2. #chroot /mnt/sysimage (루트권한 획득)

#/sbin/grub-install /dev/hda (grub 재설치)

exit

exit

리눅스 데몬

amanda : 네트워크 백업 클라이언트 데몬

amandaidx : amanda 패키지의 일부인 amandaidx 데몬

amd : 자동마운트 데몬. 시스템이 요청하면 자동으로 장치와 NFS 호스트를 마운트

amidxtape : amand 패키지의 일부인 amidxtape 데몬

anacron : 지정한 시간에 프로그램을 실행하는 데몬. cron과 비슷, 계속 켜두지 않는 경우에 사용

apmd : 자동전원관리 데몬. 배터리 상태를 감시하고 전원을 내린다.

arpwatch : 이더넷 카드의 IP 주소를 유지하는 데몬

atd : 특정 시간 또는 시스템 부하가 적을때 지정된 명령을 실행시키는 데몬

autofs : 파일 시스템을 사용할 때 자동으로 마운트 시켜주는 데몬

chargen : chargen의 TCP 버전 서버

chargen-upd : chargen의 UDP 버전 서버

cipcd : IP 주소를 암호화하는 CIPE 데몬

crond : cron을 실행시키는 데몬, 특정 프로그램을 정해진 시간에 주기적 실행 유닉스 표준 프로그램

daytime : daytime의 TCP 버전 서버. daytime은 클라이언트의 질의에 응답하여 아스키 형태로 현재

시간과 날짜를 출력하는 데몬. TCP 포트 13을 사용

daytime-udp : daytime의 UDP 버전 서버. UDP포트 13을 사용

dhcpcd : Dynamic host configuration protocol server daemon.

echo : echo 의 TCP 버전 서버

echo-udp : echo 의 UDP 버전 서버

finger : finger 요청에 응답하는 서버. finger는 사용자에게 대한 로그인 네임, 디렉터리, 셸과 최종 로그인 시간에 대한 정보를 제공 하는 프로토콜

gated : 라우팅 데몬.

gpm : 텍스트 기반 리눅스용 애플리케이션에서 마우스를 쓸수 있게 해주는 데몬.
콘솔에서 마우스를 이용한 팝업 메뉴와 복사/ 붙이기 기능도 지원

httpd : 웹 서비스를 위한 아파치 데몬.

identd : 특별한 TCP 연결에서 사용자의 신원을 결정해 주는 데몬.

TCP 포트번호를 주면 연결된 서버 시스템 소유자를 확인할 수 있는 문자열을 돌려줌

imap : 원격 사용자가 imap 클라이언트(Pine, netscape communicator)를 이용하여 자신의 메일에 접근할 수 있게 하는 서비스

imaps : 원격 사용자가 SSL을 지원하는 imap 클라이언트(netscape communicator, fetchmail 등)를 이용하여 자신의 메일에 접근할 수 있게 하는 서비스

innd : 유즈넷 뉴스 서버를 설정할 수 있는 데몬

ipchains : 패킷 필터링 방화벽 규칙을 실행, 관리하는 데몬

iptables : 패킷 필터링 방화벽 규칙을 실행, 관리하는 데몬 ipchains의 상위 버전

ipop2 : 원격 사용자가 pop2 클라이언트를 이용하여 메일에 접근할 수 있게 하는 서비스

ipop3 : 원격 사용자가 pop3 클라이언트를 이용하여 메일에 접근할 수 있게 하는 서비스

irda : irda 가 정상적으로 동작하도록 해 주는 데몬

keytable : /etc/sysconfig/keytable로 키보드 유형을 변환할 수 있게 하는 서비스.

kudzu : 새롭게 추가된 하드웨어를 검색, 설정하는 데몬

linuxconf : 리눅스 시스템의 표준 관리도구 데몬

linuxconf-web : 웹을 통해 linuxconf를 실행할 수 있게 하는 데몬

lpd : 프린트 서비스 데몬

mars-nwe : netware IPX 프로토콜을 사용하는 클라이언트에게 리눅스 머신에서 파일과 프린트 서버를 호환시켜 주는 데몬

mcsvr : midnight command(MC) 서버. MC끼리 네트워크를 공유한다

mysqld : mysql 데이터 베이스 서버 데몬

named : 도메인 네임과 IP주소를 해석해주는 DNS서버(BIND) 데몬.
로컬 호스트에서 DNS서버를 운영할 때만 실행 시킨다.

netfs : samba, 네트워크 파일 시스템(NFS), NCP(netware)등의 마운트와 언마운트에 관여하는 데몬.

network : 시스템 부팅시 네트워크 인터페이스를 설정하는 데몬.

nfs : TCP/IP 네트워크에서 파일을 공유할 수 있게 하는 데몬.

nfslock : NFS파일을 locking 하는 데몬

nscd : NIS/NS 를 사용할 수 있게 하는 데몬.

nscd는 실행중인 프로그램의 그룹을 살피고 패스워드를 변경 결과를 캐시

ntalk : 서로 다른 시스템끼리 채팅이 가능하게 하는 서버

ntpd : NTPv4데몬

pcmcia : 휴대용 PC에서 pcmcia장치를 쓸 수 있게 하는 데몬.

pop3s : SSL을 지원하는 pop3클라이언트를 사용하여 메일에 접근할 수 있게 하는 서비스

portmap : RPC(NFS, NIS, mcsev등) 연결을 관리하기 위한 포트 매핑 데몬으로 RPC를 사용하는 프로그램을 실행하기 위해서는 반드시 선택하여야 하는 데몬.

postgresql : postgresql 데이터베이스 데몬

pppoe : adsl서비스에 연결시켜 주는 데몬

proftpd : ftp 서버 데몬

pxe : 부팅 전 실행환경 서버. 다른 PXE기반 머신에 네트워크 부팅을 제공

random : 시스템에 필요한 난수 발생 및 저장 데몬

rawdevices : HDD 파티션과 같은 블록 디바이스를 위한 스크립트. /etc/sysconfig/rawdevices 파일을 편집하여 원시 디바이스를 블록 디바이스로 매핑 할수 있다.

reconfig : /etc/reconfigsys 파일이 존재하면 재설정을 실행하는 데몬

rexec : rexec(3) 루틴을 위한 서버 데몬. 인증된 사용자 이름과 패스워드로 원격 실행 제공 서버

rlogin : rlogin 프로그램을 위한 서버 데몬. 신뢰하는 호스트로부터 인증을 통해 원격 로그인을 제공

routed : RIP 프로토콜을 통해 업데이트된 자동 IP 라우팅 테이블 설정 데몬

rsh : 신뢰하는 호스트로부터 인증 통해 원격 실행을 제공.

rstat : 네트워크에 연결된 사용자에게 네트워크 상의 머신에 대한 퍼포먼스 매트릭스 회수 프로토콜

rsync : 원격 머신간 파일을 동기화하는 데몬

rusersd : 네트워크에 특정 사용자가 있는 지 검색하는 데몬.

rwalld : 시스템에 동작중인 모든 터미널에 메시지를 표시할 수 있게 해 주는 데몬

rwild : 원격 접속자의 목록을 볼 수 있게 해주는 데몬. finger와 비슷한 기능

sendmail : 메일을 다른 호스트로 전송하는 메일 전송(Mail Transport Agent)데몬

smb : SMB 네트워크 서비스를 제공하기 위한 samba 서버(smbd와 nmbd)데몬

snmpd : SNMP(Simple Network Management Protocol)데몬

squid : 프록시 서버 데몬. HTTP, FTP, gopher와 같은 프로토콜 사용시 캐싱 속도를 높이는 데몬.

sshd : openssh 서버 데몬

swat : samba 웹 관리 툴, samba 서버의 설정을 웹 브라우저로 할 수 있게 하며 901포트 사용

syslog : 로그 메시지를 로그파일에 기록하는데 사용하는 데몬. syslog는 항상 실행.

talk : 터미널의 내용을 다른 사용자에게 뿌려주는 데몬.

telnet : 원격접속(telnet 세션)을 제공하는 서버. 인증을 위해 사용자 이름과 패스워드를 사용

tftp : 파일 전송을 위한 프로토콜.

time : rdate 데몬에 의해 사용되는 RFC 868 시간 서버의 TCP 버전

time-udp : rdate 데몬에 의해 사용되는 RFC 868시간 서버의 UDP 버전

webmin : webmin 관리자 서버 데몬

xfs : 부팅과 셧다운시 X 폰트 서버를 시작하거나 종료시키는 데몬

xinetd : inetd 데몬의 보안 강화 버전. telnet, ftp 등과 같은 서비스를 처리하는 슈퍼 데몬.

ypbind : NIS/YP 클라이언트에서 실행되는 데몬으로 NIS(Network Information Service)도메인을 바인드한다. NIS를 사용하지 않는 시스템에서는 실행하지 않는다.

yppasswd : NIS클라이언트 사용자의 패스워드를 변경할 수 있게 해 주는 데몬

ypserv : 표준 NIS/YP 네트워크 프로토콜 서버. 호스트 네임, 사용자 네임과 다른 정보 DB를 네트워크를 통해 배포. 클라이언트에서는 필요하지 않으며 NIS 서버에서 실행.