리눅스 개요와 구조 & 디렉토리 구조

by grandmarnier

목차

- 리눅스란?
- •리눅스의 역사
- •리눅스의 장, 단점
- •리눅스의 구조
- •리눅스의 디렉토리 구조

Linux?

운영체제 (Operating System, OS)

OS

- 하드웨어를 직접 관리하는 소프트웨어
- ⇒사용자에게 컴퓨터의 프로그램을 쉽고 효율적으로 실행할 수 있는 환경을 제공
- ⇒컴퓨터 시스템 하드웨어 및 소프트웨어 자원을 여러 사용자 간에 효율적 할당, 관리, 보호하는 것
- ⇒사용자 프로그램의 오류나 잘못된 자원 사용을 감시하는 것과 입출력 장치 등의 자원에 대한 연산과 제어를 관리

OS에서 하는 일들

- •프로세스 관리
- •인터럽트
- •메모리 관리
- •파일 시스템
- •장치 드라이버
- •네트워킹 (TCP/IP, UDP)
- •보안 (프로세스/메모리 보호)
- •입출력 관리
- •등등 ...

Linux

유닉스 계열의 운영체제

1969 : UNIX (C로 쓰여진 최초의 운영체제)

1987: 교육용 OS 인 minix (Andrew S. Tanenbaum)

=> 다른 사람이 수정할 수 없도록 제한

Linus Tovals 가 이를 이용하여 개인 PC용 운영체제를 만들 계획을 세움 초기에는 운영체제가 아닌 운영체제(minix) 위에서 실행되는 터미널 환경

⇒파일을 제어할 수 있는 수준까지 만든 후

포직스(POSIX)에 호환되는 운영 체제 커널로 발전

1991: GNU 라이선스 아래에서 커널 소스를 공개함 (o.o1)

리눅스의 특징

- 1. 대부분의 리눅스는 무료 예외) 래드햇 리눅스, 이전의 SUSE Linux
- 2. 대부분의 플랫폼 (IA-32, AMD64, SPARC, PowerPC, ARM, etc...) 에서 사용가능
- ⇒간단히 말하면, 인텔,AMD, 썬, 애플 등 많은 회사들이 만들어낸 하드웨어 제어 방법에 호환되며 일반 PC 가 아닌 임베디드 기기에 서도 호환성을 보여줌

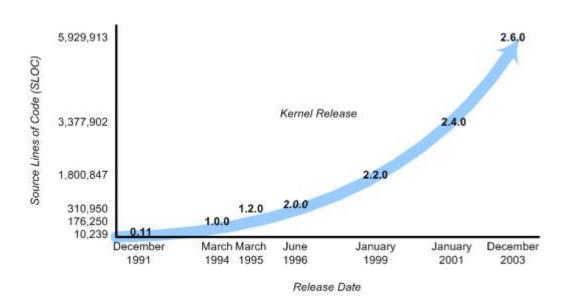
But, 슬랙웨어 리눅스: 인텔 아키텍쳐만 지원

3. 유닉스와 호환되며 표준 라이브러리, 프로그래밍 도구, 컴파일러, 디버거 등 의 완전한 유직스 프로그래밍 환경을 제공

- 4. 강력한 네트워크 지원
- •TCP/IP, IPv4,IPv6, UDP, UUCP 등의 다양한 프로토콜
- •이더넷, 토큰링, 전화 접속, 무선 네트워크 등 다양한 네트워크 기술을 사용할 수 있음
- •인터넷 기반 어플리케이션 지원
- •리눅스 자체를 라우터 및 방화벽으로 동작시킬 수 있음
- •다양한 웹서버 지원(apache 등)
- •그 외 다양한 네트워크 서비스(samba, FTP, NFS 등)
- 5. 다양한 파일 시스템 지원 (ext2,ext3, fat, ntfs 등)
- 6. 낮은 하드웨어 사양에서도 사용가능

7. 리눅스 및 리눅스 어플리케이션들은 대부분 소스가 공개되어 자유롭게 사용, 수정, 재배포가 가능함.

8. 전 세계 수백만의 개발자들에 의해 기능이 나날이 발전하고 있음



리눅스 배포판

Debian ,Ubuntu ,Redhat ,Fedora,CentOS ,Mandriva, Slackware,SUSE Gentoo , Arch 등 여러 배포판이 존재함

현재 동아리에서는 주로 debian(sparcs), ubuntu(bit), fedora, centOS(ICUBE) 를 사용하고 있음

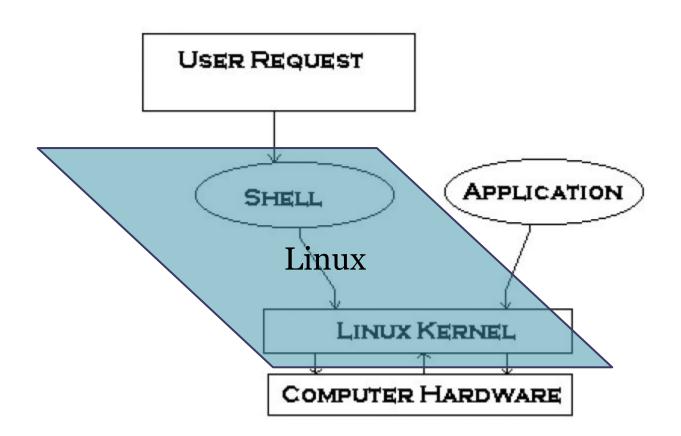
Debian: 데비안 프로젝트에서 만들어서 배포 패키지 설치 및 업그레이드가 쉽다는 장점이 있다. Dpkg, apt 등을 이용하여 소프트웨어의 설치나 업데이트에서 다른 패키지와의 의존성 확인, 보안관련 업데이트 등을 자동으로 해준다.

Ubuntu: 데스크톱에서 리눅스를 쉽게 사용할 수 있게 만든 리눅스 배포판이다. 데비안 GNU/리눅스(Debian GNU/Linux)를 바탕으로 만들어짐. 캐노니컬의 지원을 받는다

Fedora: 레드햇에서 후원하고 페도라 프로젝트에서 주관한다. 6개월 간격으로 새로운 버전이 배포되며 지원기간은 각 버전마다 13개월 SELinux (리눅스 보안 솔루션) 를 내장하고 있음. 주로 데스크탑에서 사용

CentOS: CentOS(The Community ENTerprise Operating System) 레드햇 엔터프라이즈 리눅스(상용)의 소스 코드를 그대로 가져와 사용함. 기본적으로 포함되는 소프트웨어와 업데이트되는 소프트웨어를 아울러 이진파일에 대해 상위판과 100%에 최대한 가까운 호환성을 유지

리눅스의 구조



Kernel & Shell

Kernel: 운영체제의 핵심적인 부분

- •사용자 프로그램과 하드웨어 장치 사이의 인터페이스, 프로세스 스케 줄링, 등 시스템의 여러 부분을 제어함
- •실제로 돌아가는 프로세스는 아니며 RAM 위에 커널의 코드가 올라간 상태로 사용됨
- •다른 프로세스에 의해 덮어쓰이지 않도록 고유의 메모리 영역을 사용
- •어려운 개념이므로 스스로 꼭 한 번 더 찾아볼것

Shell:운영체제의 외곽부분

- •유저와 직접 소통하며 유저에게 명령을 읽어 실행
- •그 외 멀티 프로세스 제어, 리다이렉션(표준 입출력의 방향을 바꿈),
- 쉘 스크립트 등 의 기능을 제공
- •주로 bash가 쓰이며 그 외에도 tcsh, ash, rc, zsh 등이 있다. => /etc/shells 에 사용가능한 쉘의 위치가 기록되어 있다

리눅스의 데이터 관리

리눅스에서 관리하는 모든 것은 파일로 인식 (일반 파일, 하드웨어, 현재 돌아가는 프로세스 등..)

리눅스에서는 '디렉토리' 라는 '파일'을 이용해서 전체 파일을 하나의 **트리**로 만듬

이 때 트리의 root는 / 로 나타내며 root directory 라 합니다.

디렉토리: 하위의 디렉토리들과 파일들의 정보를 저장하는 파일

한 프로세스는 하나의 작업 디렉토리를 가진다. (/proc 참조)

절대 경로 : / 를 기준으로 한 경로

상대 경로: /를 포함하지 않는 경로, 쉘의 작업 디렉토리가 상대 경로의 기준 디렉토리 자기 자신을 . 로 표현 바로 위의 디렉토리는 .. 로 표현

리눅스의 사용자 계층

1. root : 최고 관리자. 컴퓨터와 사용자들을 관리

2. user: root외의 일반 사용자들

Option) sudoer: sudo 라는 유틸리티에 의해 정의된 사용자 sudo는 root가 허가한 사용자에 대해 일정 시간(보통 5분) 동안 일부 명령을 root권한으로 사용하게 함

> /etc/sudoers 파일에 sudo를 사용할 수 있는 사용자와 각 사용자가 root권한으로 실행할 수 있는 명령이 정의 되어 있음

리눅스는 사용자들을 그룹으로 묶어서 관리함

모든 유저는 적어도 하나의 그룹에 속해있음

일반적으로 모든 파일은 파일의 소유주, 그룹, 그 외 사용자의 세의 단위로 분리되어 각 단위마다 권한을 다르게 지정할 수 있음

=>권한을 변경할 수 있는 권한은 파일의 소유주와 루트에게

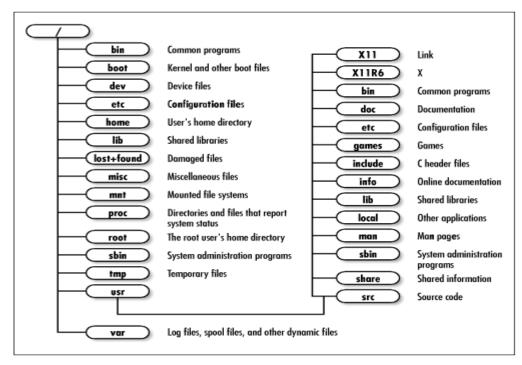
리눅스의 디렉토리 구조

specification 이 있음

문서화된 자료(영어):

http://sparcs.org/~grandmarnier/fhs-2.3.pdf

⇒원문은 <u>http://www.pathname.com/</u>



1) root & home

- •home 디렉토리는 리눅스 사용자들의 개인 공간
- •보통 일반 사용자는 home 밑에 별도의 설정이 없는 경우 자신의 아이디로 된 디렉토리가 배정됨 => 로그인 시 처음 작업 디렉토리가 됨
- •/root는 root 의 홈 디렉토리

2) bin & sbin

- •리눅스 사용에 필수적인 명령어들을 모아놓은 디렉토리
- •sbin 은 시스템 관리를 위한 명령들 => root에게만 허용됨
- •이렇게 명령어들을 모아놓은 디렉토리의 위치는 PATH 환경 변수에 저장되어 있음
- => 쉘에서 \$echo \$PATH 를 쳐보세요



3) lib

- •리눅스 상에서 자주 사용되는 라이브러리를 모아놓음
- •Shared object file format(.so)
- •윈도우에서의 Dynamic Linked Library(DLL)을 모아놓은 폴더로 생각할 수 있음

```
_ | D | X
grandmarnier@sparcs: /lib
grandmarnier@sparcs:/lib$ ls
                                        libnss_files-2.7.so
срр
                                        libnss_files.so.2
                                        libnss_hesiod-2.7.so
i486-linux-gnu
                                        libnss_hesiod.so.2
                                        libnss_mdns.so.2
klibc--lowhOVR87LX1LY95rmnFLc1vuY.so
                                       libnss_mdns4.so.2
Id-2.7.so
                                        libnss_mdns4_minimal.so.2
ld-linux.so.2
                                        libnss_mdns6.so.2
                                        libnss_mdns6_minimal.so.2
libBrokenLocale-2.7.so
libBrokenLocale.so.1
                                        libnss_mdns_minimal.so.2
libSegFault.so
                                        libnss_nis-2.7.so
                                        libnss_nis.so.2
libacl.so.1
                                        libnss_nisplus-2.7.so
libanl-2.7.so
                                        libnss_nisplus.so.2
libanl.so.1
                                        libpam.so.O
libattr.so.1
                                        libpam.so.0.81.12
libattr.so.1.1.0
                                        libpam_misc.so.O
libblkid.so.1
                                        libpam_misc.so.0.81.3
 ibblkid.so.1.0
                                        libpame.so.0
 ibbz2.so.1
                                        libpamc.so.0.81.0
 ibbz2.so.1.0
                                        libpcprofile.so
 ibbz2.so.1.0.4
                                        libpopt.so.0
                                        libpopt.so.0.0.0
```

4) etc

리눅스 내의 설정 파일들이 위치함

주요 설정 파일들

•motd : 로그인 성공 시 처음에 띄워주는 문구

•passwd: 리눅스 사용자에 대한 정보 (예전에는 사용자의 비밀번호(!) 도 있었음)

•shadow : 리눅스 사용자 패스워드 정보 (root만 볼수 있음)

•sudoers: sudo 권한을 가진 사용자와 권한에 대한 정보

•fstab : 파일 시스템과 마운트 포인터

•hosts: 도메인 네임

•issue : 리눅스 배포반 및 버전

•skel/: 새로운 사용자가 추가될 때 홈 디렉토리에 기본적으로 복사

해줄 파일들을 포함한 디렉토리

•init.d/: 컴퓨터가 부팅될 때 실행될 파일들을 넣어둔 디렉토리

이 외에도 많은 파일들이 중요한 설정을 가지고 있음

5) proc

- •RAM 위에 존재하며 커널에 의해 제어됨
- •현재 실행되는 프로세스와 실제로 사용되는 장치, 커널이 수집한 하드웨어 정보가 저장됨
- •사용자가 /proc이나 하위 파일에 접근할 때마다 커널에서 파일 내용을 동적으로 만들어냄
- •각 프로세스는 고유의 ID를 가지고 있으며(숫자) 이 아이디를 가 진 디렉토리 밑에 각종 정보를 저장함

=>

cmdline: 프로세스를 시작한 명령행 내용

status : 프로세스의 내부 상태 정보 cwd : 프로세스의 현재 작업 디렉토리

/proc/meminfo : 총 메모리 사용 현황 /proc/stat : 시스템의 상태에 관한 정보

/proc/uptime : 시스탬이 부팅된 후 흐른 시간

/proc/version : 현재 실행되는 커널 버전

6) dev

- •실제 로 하드 디스크에 존재하지는 않는 정보를 저장함
- 모든 하드웨어(hdd,cdrom, 마우스, 가상 장치,etc...) 를 파일로 인식 하며 이를 장치 파일이라 함
- •psaux PS/2 마우스 장치
- •tty : teletypewritter, 사용자와 시스템 사이를 중계하는 역할 사용자와 시스템 사이의 입출력은 모두 여기를 통해 이루어짐 서버 시스템에 직접 연결된 장치에서 사용함
- •pts : telnet, ssh 등을 이용해 원격으로 접속할 경우 (pseudo TTY slave)

유사 장치:실제 장치와 관련이 없음

•null: 이 장치로 들어간 데이터는 모두 사라짐

•zero : 이 장치에서는 항상 null을 반환

7) mnt & media

둘 다 외부 저장 매체가 마운트 되는 디렉토리

마운트: 한 파일 시스템의 루트 디렉토리를 다른 디렉토리에 붙임으로써 디렉토리를 사용할 수 있게 만들어주며, 모든 파일 시스템들을 마치 그들이 속해있는 파일시스템의 서브 디렉토리인 것처럼 사용 가능하게 만드는 것

문서에는 media 는 주로 플로피 디스크, CD 등이 마운트 되고 mnt 는 수동적으로 마운트 시켜야 하는 저장 매체를 마운트 되는 곳으로 쓰도록 되어있다.

하지만, 이제는 많은 저장 매체들이 자동으로 마운트 되기 때문에 주로 /media(특히 우분투) 를 사용하는 경우가 많지만 mnt를 사용하는 것은 자유

8) tmp

- •리눅스 상에서 실행되는 프로세스들의 임시 파일들을 저장 하는 위치
- •종료 시 이 위치에 있는 파일들은 모두 삭제됨

9) boot

- •부팅에 필요한 필수 파일들이 저장됨
- •lilo, grub 등의 부트 로더와 커널이 위치함

부트 로더: 컴퓨터를 켰을 때 가장 먼저 실행되는 프로그램 => OS의 커널을 로드하고 몇몇 커널 파라메터를 커널에 넘김

10) var

- •리눅스 상에서 자주 변경되는 데이터들을 모아놓음
- •/var/lib 일반적인 시스템 운용시 계속 갱신되는 파일들을 위한 공간
- •/var/local
- •/usr/local 아래에 설치된 프로그램들의 다양한 데이터가 보관
- •/var/lock : 잠금 파일(lock file)이 있는 곳이다.
- •/var/log: 다양한 프로그램들의 로그 파일
 - •/var/log/wtmp : 시스템의 모든 로그인, 로그아웃 정보를 기록
 - •/var/log/messages: 커널과 시스템 프로그램들의 모든 메시지
- •/var/run : 시스템의 현재 정보, 부팅 시 리셋
 - •/var/run/utmp : 현재 로그인한 사용자들에 대한 정보
- •/var/spool : 대기 상태에 있는 작업들을 위한 디렉토리
- •/var/tmp:/tmp에 있는 임시 파일들보다는 좀 더 오래 유지될 필요가 있는 임시 파일들, 부팅 시 지워지지 않는다.

11) usr

- •리눅스 상에서 가장 큰 공간을 사용하는 디렉토리
- •리눅스 배포판에 따라 많은 차이를 보임
- •어플리케이션 수준의 프로그램들이 여기에 있음
- usr 밑의 bin, lib,... 등은 사용자가 자유롭게 다룰 수 있으며 루트 디렉토리의 같은 이름을 가진 디렉토리와 같은 역할을 한다. (대신 우선순위가 떨어진다.)

usr / 밑의 디렉토리 설명

- •usr/bin : 실행 가능한 명령어, 주로 콘솔이나 X에서 사용되는 명령어들
- •usr/sbin: 시스템 관리를 위한 명령어, 주로 서버 관리용
- •usr/lib: 프로그램과 하위 시스템을 위한 라이브러리
- •/usr/man, /usr/info, /usr/doc :각각 매뉴얼 페이지, GNU Info 문서들, 그리고 기타 다른 문서
- •usr/include: C 를 위한 헤더 파일
- •usr/X11R6 : X window
- •usr/src : 시스템에 빌드하는 프로그램의 소스
- •usr/share: 읽기 전용 자료, 주로 매뉴얼이나 문서 자료

usr/local : 리눅스에서 필수적인 파일이나 배포판에서 확장되는 파일들을 제외한 응용 프로그램 파일들을 저장

응용 프로그램 설치 시 프로그램이 독자적으로 명령어를 추가하 거나 라이브러리가 추가적으로 필요로 할 경우 등 추가적인 설 정이 필요할 때, 이 위치에 저장하기 때문에 usr과 흡사한 구조 로 되어있다.

```
grandmarnier@sparcs:/usr/local$ Is /usr /usr/local /usr:

X11R6 bin games include lib lib64 local sbin share src

/usr/local:
bin etc games include lib man sbin share src

grandmarnier@sparcs:/usr/local$
```

/bin : 문서에 명시되어 있는 명령어 및 필수적인 명령어

/usr/bin : 주로 콘솔과 X에서 사용되는 명령어

/usr/local/bin : 그 외 어플리케이션에 의해 추가되는 명령어

sbin, lib, etc, 등도 이와 비슷한 구조를 가짐

12) opt

- •주로 규모가 있는 소프트웨어 패키지가 들어있음
- gnome, kde, 오픈오피스, 파이어폭스 등등 ...

출ᄎ

```
러닝 리눅스 5판
위키피디아 http://en.wikipedia.org/
텀즈 http://terms.co.kr/
그외:
http://www.dal.kr/chair/cgj/cgj0907.html
http://www.dal.kr/chair/wh/who902.html
http://4ellene.net/tt/193
http://coffeenix.net/doc/kuls/file_system.html
http://futurist.se/gldt/wp-content/uploads/11.04/gldt1104.png
http://www.levenez.com/
http://blog.wowpc.net/15
http://mylucky8.tistory.com/206
http://www.debian.org/releases/sarge/powerpc/apcso2.html.ko
http://wiki.kldp.org/Translations/html/SysAdminGuide-
KLDP/x642.html
http://www.pln.kr/support/tech/
http://www.test104.com/kr/tech/1248.html
http://www.pathname.com/
http://kimgisa.net/tag/tty
http://www.lug.or.kr/docs/LINUX/others/01-05-5.htm
```

X GNU License, GPL

GNU 일반 공중 사용 허가서(GNU General Public License, GNU GPL 또는 GPL) :자유 소프트웨어 재단에서 만든 자유 소프트웨어 라이선스이 허가를 가진 프로그램을 사용하여 새로운 프로그램을 만들게 되면 파생된 프로그램 역시 같은 카피레프트를 가져야 한다.

GPL은 다음의 다섯 가지의 의무를 저작권의 한 부분으로서 강제한다.

- 1. 컴퓨터 프로그램을 어떠한 목적으로든지 사용할 수 있다. 다만 법으로 제한하는 행위는 할 수 없다.
- 2. 컴퓨터 프로그램의 실행 복사본은 언제나 프로그램의 소스 코드와 함께 판매 하거나 소스코드를 무료로 배포해야 한다.
- 3. 컴퓨터 프로그램의 소스 코드를 용도에 따라 변경할 수 있다.
- 4. 변경된 컴퓨터 프로그램 역시 프로그램의 소스 코드를 반드시 공개 배포해야 한다.
- 5. 변경된 컴퓨터 프로그램 역시 반드시 똑같은 라이선스를 취해야 한다. 즉 GPL 라이선스를 적용해야 한다.

X POSIX

Portable Operating System Interface

유닉스 운영체계에 기반을 두고 있는 표준 운영체계 인터페이스