



리눅스 개요

2007. 09. 19

안효창

Contents

1

리눅스의 역사

2

리눅스의 특징

3

리눅스 배포판 설치

4

리눅스 시스템 디렉토리 구조

리눅스의 역사

- 일반적인 운영체제의 정의
 - 컴퓨터 사용자와 하드웨어간의 매개 역할을 하는 프로그램
- 목적
 - 사용자가 편리하게 프로그램을 수행할 수 있는 환경과 시스템을 효율적으로 사용할 수 있는 환경을 제공해 주는 것
- 커널 : 좁은 의미의 운영체제
 - 시스템 자원을 통제/관리하는 역할
 - 넓은 의미의 운영체제
 - 커널의 기능과 커널 위에서 수행되는 라이브러리, 명령어, 응용 프로그램들을 포함하여 지칭

리눅스의 역사

- 유닉스의 등장
 - 제 1버전 개발
 - 1969년 미국 AT&T사의 벨 연구소 켄 톰슨(Ken Thompson)이 DEC사의 미니 컴퓨터 PDP-7에서 어셈블리어로 단일 사용자용인 유닉스의 제1버전을 개발
 - C언어로 재작성
 - 1972년 데니스 리치(Dennis Ritchie)가 고급언어인 C로 다시 작성함
 - 일부분만의 수정 작업과 더불어 적용할 시스템의 환경에서 실행되는 컴파일러를 사용하여 쉽게 적용이 가능

리눅스의 역사

- 수많은 변종이 생산
 - 여러 대학들과 기업체들에 배포되어 독자적으로 발전하며, 현존하는 대부분의 운영체제에 영향을 끼침
 - 사업성을 이유로 코드와 기술을 공개하지 않음으로써 상호 호환성 결여 문제와 이식성이 떨어지는 문제 야기
 - 1984년에 유닉스 사용자 모임에서 표준화 위원회를 설립하여 유닉스 표준화를 시도
- 유닉스 버전들의 종류
 - Solaris, SunOS, FreeBSD, HP-UX, AIX, IRIX, AIX, SCO UNIX 등
 - 알려지지 않은 버전들을 합치면 수천 종

리눅스의 역사

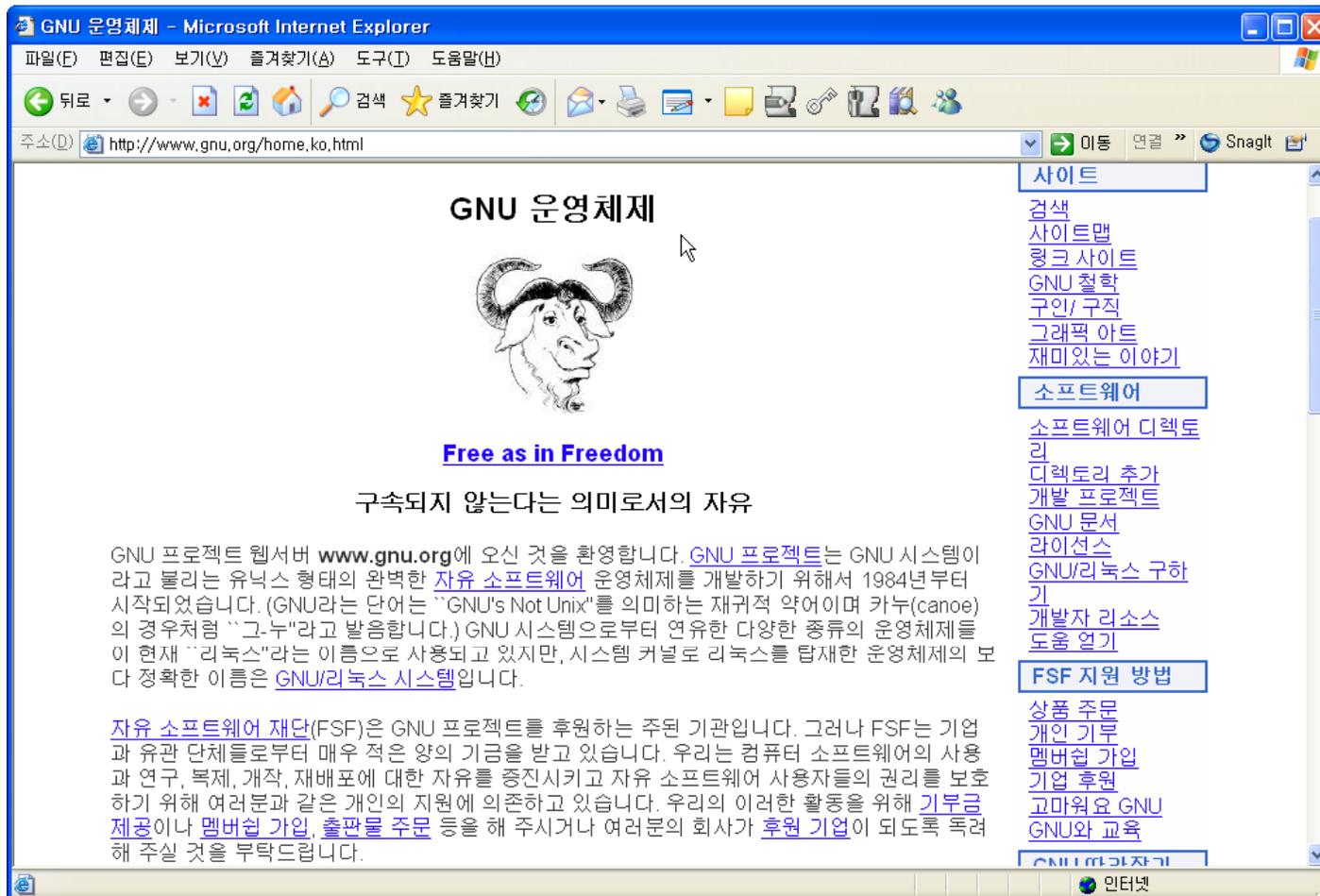
- 리눅스의 등장
 - 1987년 Andrew S.Tanenbaum에 의해 Minix가 개발됨
 - 유닉스의 일종인 오픈 소스 운영체제
 - 운영체제의 기본 기능만을 갖춘 small size
 - 교육용으로 보급
 - 리누스 토발즈에 의한 리눅스의 개발(현재 최신 버전 2.6)
 - GNU 프로젝트의 운영체제로 리눅스 선택
 - GNU/리눅스, 리눅스
 - 이후 레드햇, 칼데라, 데비안과 같은 다수의 리눅스 개발 회사 및 단체가 만들어짐

리눅스의 역사

- GNU(GNU is Not Unix), FSF(Free Software Foundation)
 - 창시자 : 리처드 스톨만
 - 모토 : 기술은 공유되어야 한다.
 - 모든 소스 코드는 오픈되어야 하며 다른 사람들에 의해 자유롭게 수정, 발전할 수 있어야 한다는 정신을 담고 있다.
 - 소프트웨어들의 소유와 독점에 대한 반대
 - GNU 프로젝트를 추진
 - Unix 운영체제와 다양한 프로그램들을 새롭게 작성하고 소스를 오픈하여 자유로운 수정 및 발전이 가능하도록 하기 위해 시작
 - 1985년 GNU 선언문(GNU Manifesto)을 발표
 - GNU 프로젝트의 목표를 설명하고 다른 사람들의 참여와 지원을 요청하기 위해 오픈 소스를 주장

리눅스의 역사

- <http://www.gnu.org/home.ko.html>



리눅스의 역사

■ 자유 소프트웨어의 요지

“자유 소프트웨어의 핵심은 구속되지 않는다는 관점에서의 자유라는 것이며 무료라는 금전적인 측면에 있는 것이 아니다”

“자유 소프트웨어는 사용자가 소프트웨어를 실행시키거나 이를 복제 및 배포할 수 있는 자유와 함께 소스 코드에 대한 접근을 통해서 이를 학습하고 수정, 개선시킬 수 있는 원천적인 자유까지를 모두 포괄하는 것이다”

1. 프로그램을 어떠한 목적을 위해서도 실행할 수 있으며
2. 프로그램의 동작 원리를 연구하고 필요에 따라 변경할 수 있으며
3. 공익을 위해 소스 코드를 복제하고 배포 및 이익을 환원시킬 수 있는 자유를 포괄

리눅스의 역사

- 초기 GNU 프로젝트의 난관
 - 운영체제의 개발 미흡
 - 1983년 유닉스 오픈 소스 버전을 만들기 위해 GNU 운영체제(HURD) 개발 시작
 - HURD : 마이크로 커널 구조로 제작
 - 프로세스들간의 조합 시 여러 문제점들이 발생하여 불안정화
 - 기존의 프로그램을 구동시킬 운영체제의 개발 부진은 GNU 프로젝트의 추진에 큰 걸림돌이 됨
- GNU 시스템의 운영체제로 리눅스를 채택
 - 1992년 무렵에 리눅스 커널과 GNU 시스템이 결합됨으로써 하나의 완성된 자유 운영체제의 모습을 갖추

리눅스의 특징

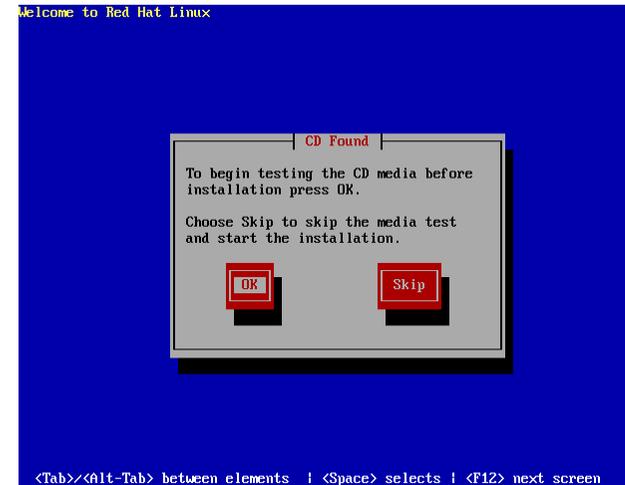
- 특징
 - 멀티태스킹(True Multi-tasking), 멀티유저(Multi-user) 지원
 - 다양한 하드웨어 플랫폼의 지원
 - 공개 소프트웨어
 - 가상 메모리(Virtual Memory) 지원
 - 공유 라이브러리(Shared Libraries) 사용
 - 실시간 페이지 적재 기능(demand loading executables)
 - 강력한 네트워크 환경을 제공
 - 다른 운영체제와의 공존 가능
 - 유닉스와 완벽한 호환(Full Compatible with Unix)
 - 다양한 파일시스템의 지원
 - 가상 콘솔(Virtual Console) 지원
 - 뛰어난 안정성과 빠른 업그레이드

리눅스의 배포판 설치

- 리눅스 배포판
 - 와우 리눅스, 레드햇 리눅스, 맨드레이크 리눅스, 수세 리눅스 등
 - 리눅스 배포판
 - 리눅스 토발즈가 개발한 커널에 여러 가지 유용한 유틸리티들이 안정적으로 작동할 수 있도록 패키징된 리눅스를 의미함

리눅스의 배포판 설치

- CD-ROM 무결성 검사
 - [OK] 버튼을 눌러 검사 수행



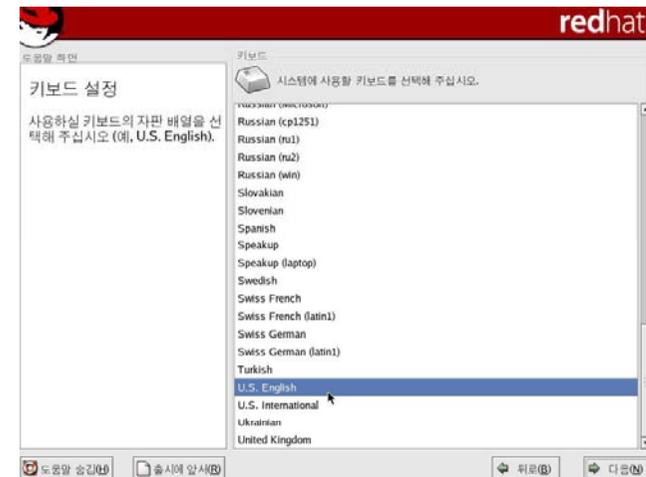
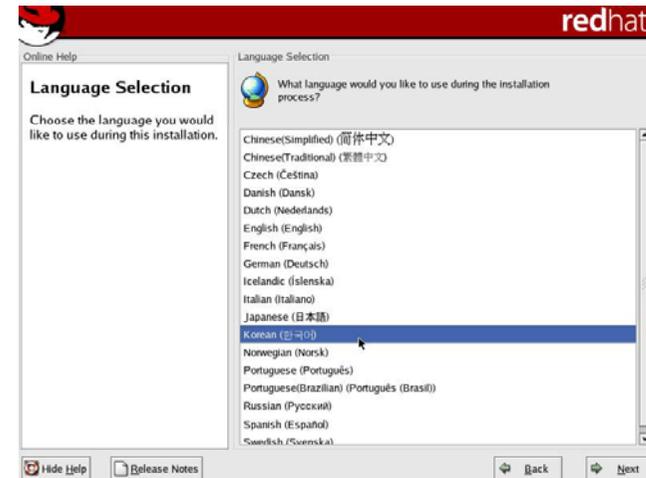
- 레드햇 설치 화면
 - 필요한 옵션 설정
 - [Next] 버튼을 눌러가며 설정



리눅스의 배포판 설치

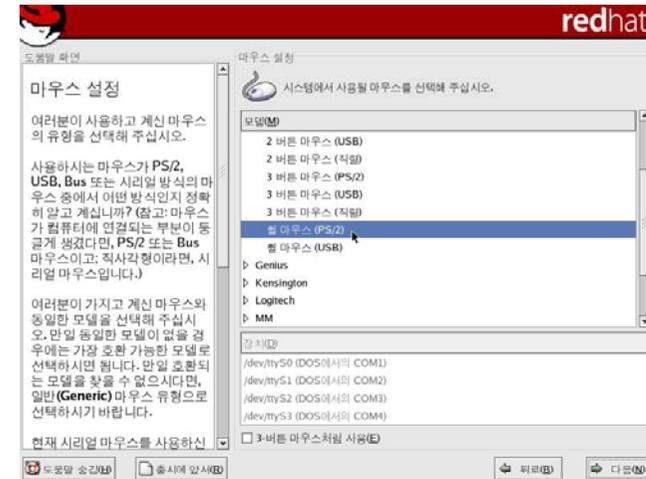
- 언어 선택
 - 'Korea(한국어)' 선택 후 [Next] 버튼

- 키보드 설정
 - 현재 사용중인 키보드 설정
 - 우리나라는 U.S 키보드 레이아웃 사용



리눅스의 배포판 설치

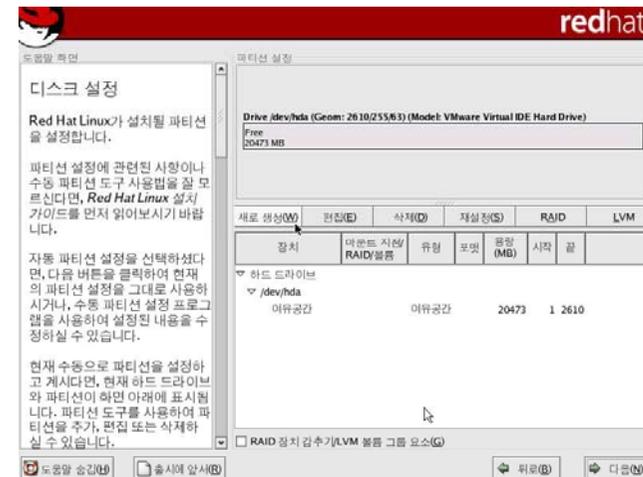
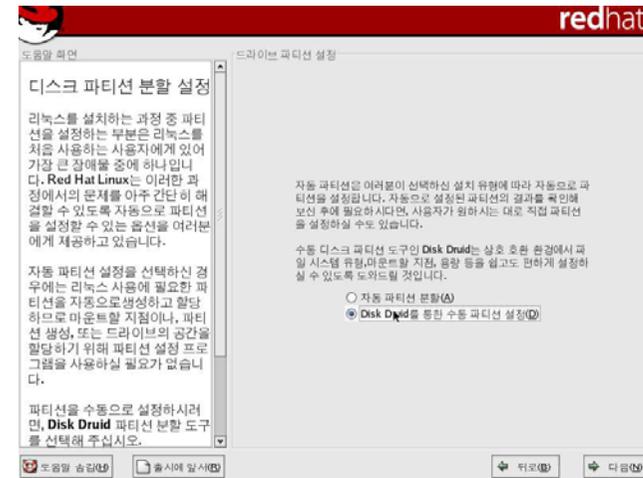
- 마우스 설정
 - 사용할 마우스 유형 설정
 - 호환 모델 없을 경우, 일반(Generic) 선택
- 설치 유형
 - 개인 데스크탑
 - 리눅스를 처음 접하는 사람
 - 워크스테이션
 - 그래픽 데스크탑 환경
 - 소프트웨어 개발 도구 제공
 - 서버
 - 서버 사용 시 필요한 패키지 설치
 - 사용자 설치
 - 부트로더, 패키지를 사용자가 직접 선택



리눅스의 배포판 설치

- 파티션 설정
 - 자동 파티션 설정
 - 수동 파티션 설정
 - Disk Druid 를 통한 수동 파티션 설정 선택

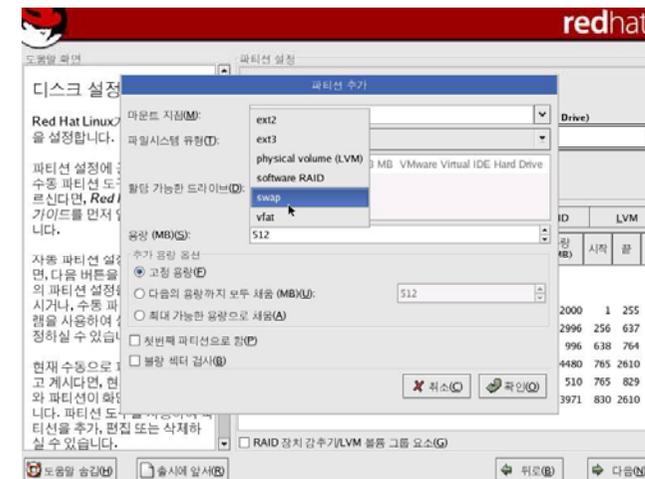
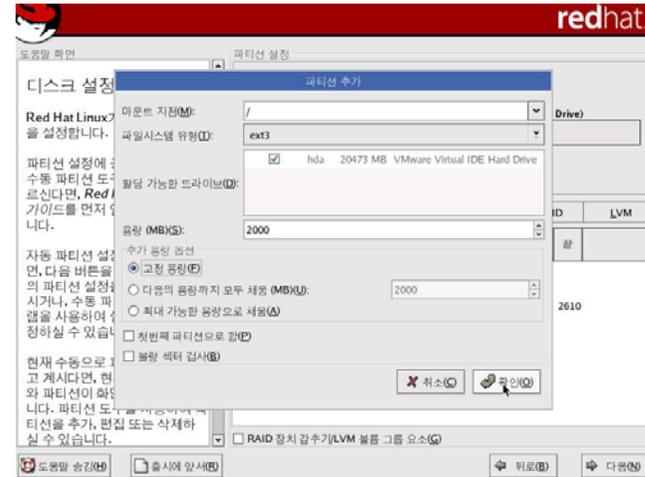
- Disk Druid를 통한 수동 파티션 설정
 - 설치 시 루트(/) 파티션 설치 필수
 - 설치 프로그램에 설치 위치를 알려줌



리눅스의 배포판 설치

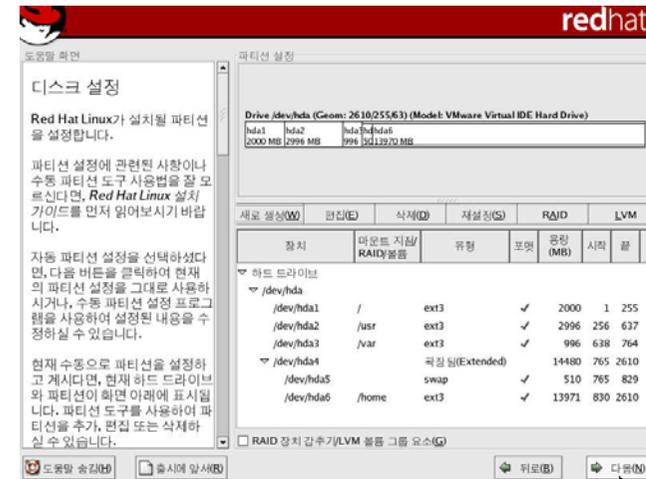
- 루트 파티션 설정
 - [새로 생성] 버튼 클릭
 - 필요 정보 입력 및 선택
 - 마운트 지점
 - 파일시스템 유형
 - 파티션 설정될 드라이브
 - 용량

- 스왑 파티션 추가
 - 파일 시스템 유형에서 swap 선택

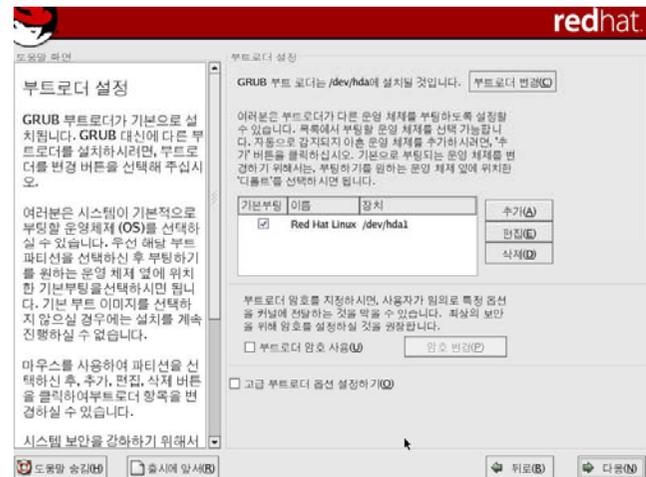


리눅스의 배포판 설치

- 파티션 설정 완료
 - 수정하려면 해당 버튼 선택

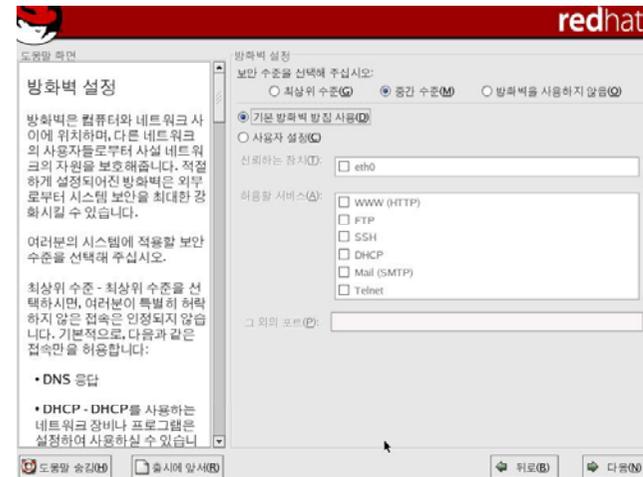
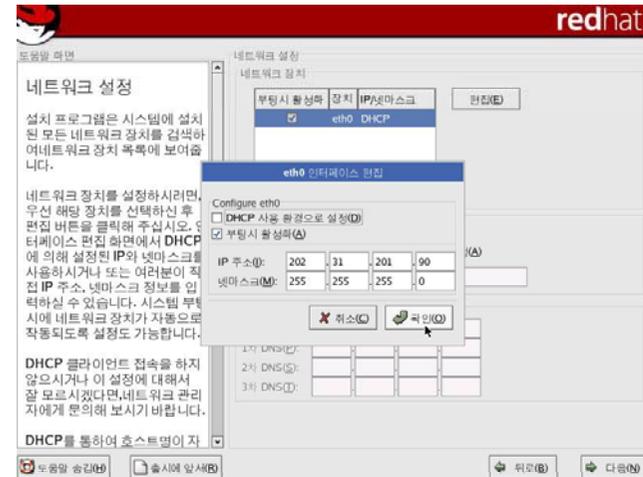


- 부트로더 설정
 - 원하는 부트로더 선택(GRUB, LILO)
 - GRUB 선택
 - 멀티 부팅 환경 제공
 - Window XP 설치 시 선택적 부팅 가능



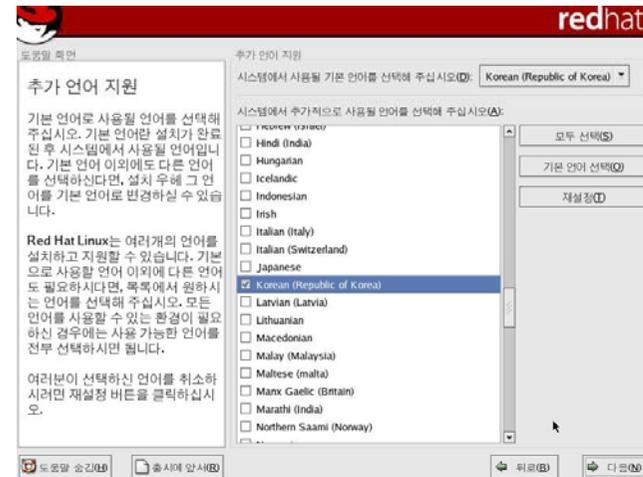
리눅스의 배포판 설치

- 네트워크 설정
 - IP 주소와 넷마스크 입력
 - [편집] 버튼 선택
 - 부팅 시 활성화 선택
 - 호스트명에 시스템 이름 입력
 - 미 입력시 localhost
 - 게이트웨이 주소 및 DNS 주소 입력
- 방화벽 설정
 - 시스템에 적용할 보안 수준 선택

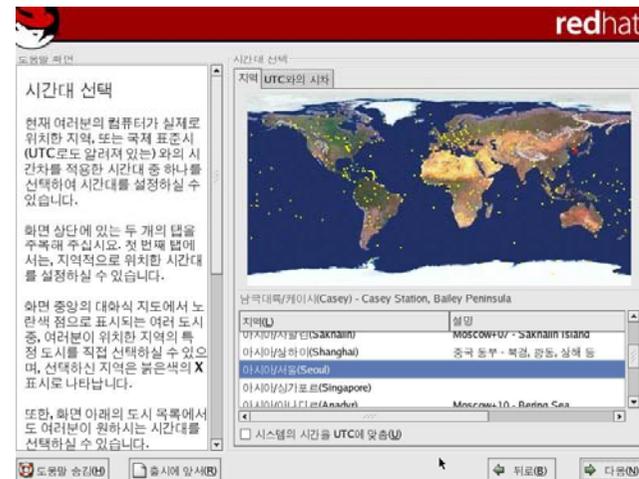


리눅스의 배포판 설치

- 추가 언어 지원



- 시간대 선택
 - '아시아/서울(Seoul)' 선택

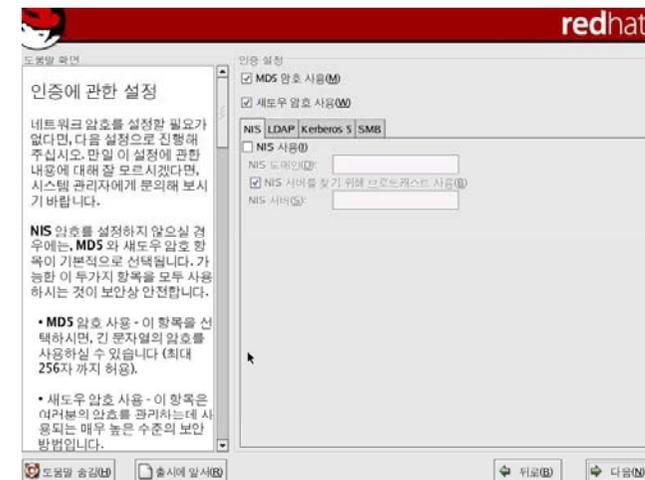


리눅스의 배포판 설치

- 루트 암호 설정
 - 관리자 암호 입력
 - 대소문자에 주의

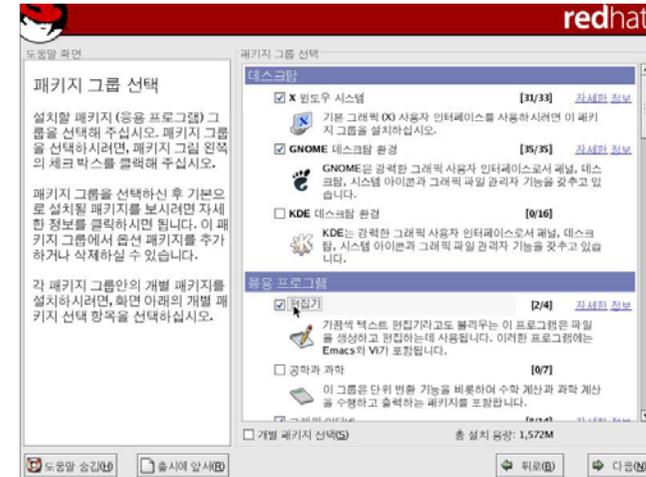


- 인증에 관한 설정
 - 보안을 위해 기본 설정 항목 그대로 사용



리눅스의 배포판 설치

- 패키지 그룹 선택
 - 필수 설치 패키지
 - X 윈도우 시스템
 - 편집기
 - 개발용 도구
 - 커널 개방



- 설치 시작

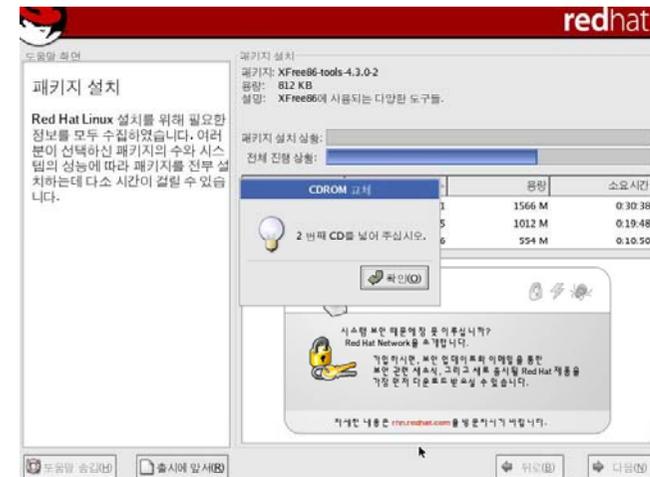


리눅스의 배포판 설치

- 패키지 설치



- 기본 패키지 외 별도 패키지 추가 시
 - 두 번째 CD 요청

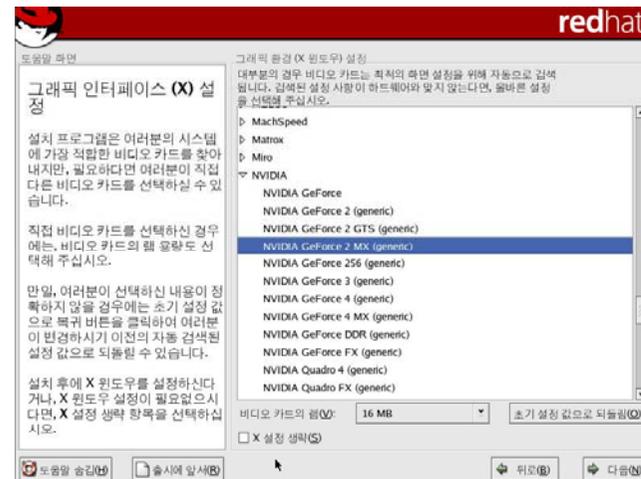


리눅스의 배포판 설치

- 부팅 디스켓 제작



- 그래픽 환경 설정
 - 최적의 화면 설정으로 자동 검색



리눅스의 배포판 설치

- 모니터 설정
 - 최적의 화면 설정을 위해 자동 검색



- 그래픽 사용자 설정
 - X 윈도우에서 사용할 색상 수, 해상도 설정
 - 로그인 유형을 그래픽 환경으로 선택
 - 로그인 유형 /etc/inittab 파일에서 변경



리눅스의 배포판 설치

- 설치 완료
 - [종료] 버튼을 눌러 재부팅

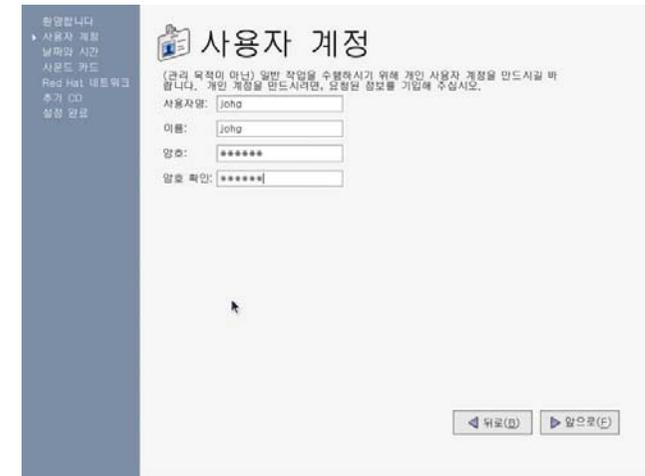


- 레드햇 리눅스 시작하기



리눅스의 배포판 설치

- 사용자 계정 추가
 - 리눅스 접속을 위한 계정 추가
- 기타 설정
 - 날짜
 - 시간
 - 사운드 카드 설정
 - 네트워크 시스템 등록
- 설정 완료

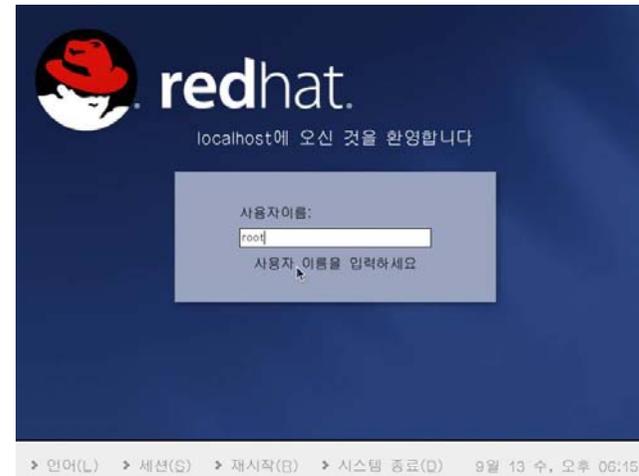


리눅스의 배포판 설치

- 온라인 매뉴얼 사용 명령어
 - man
 - 기능
 - 리눅스 명령어나 함수, 유틸리티 사용법 등에 관한 매뉴얼을 보여주는 명령으로, 시스템 개발자가 자주 사용한다.
 - 사용법 : man [옵션] 리눅스 명령
 - 옵션
 - -a : 모든 매뉴얼 페이지 출력
 - -d : 디버깅 정보 출력
 - -f : whatis와 같음
 - -h : 도움말 출력
 - -k : apropos와 같음
 - -t : groff(troff)를 사용하여 프린트를 위한 형식으로 출력

리눅스의 배포판 설치

- 계정 로그인



- 부트로더 화면
 - 시스템 부팅 시 나오는 화면

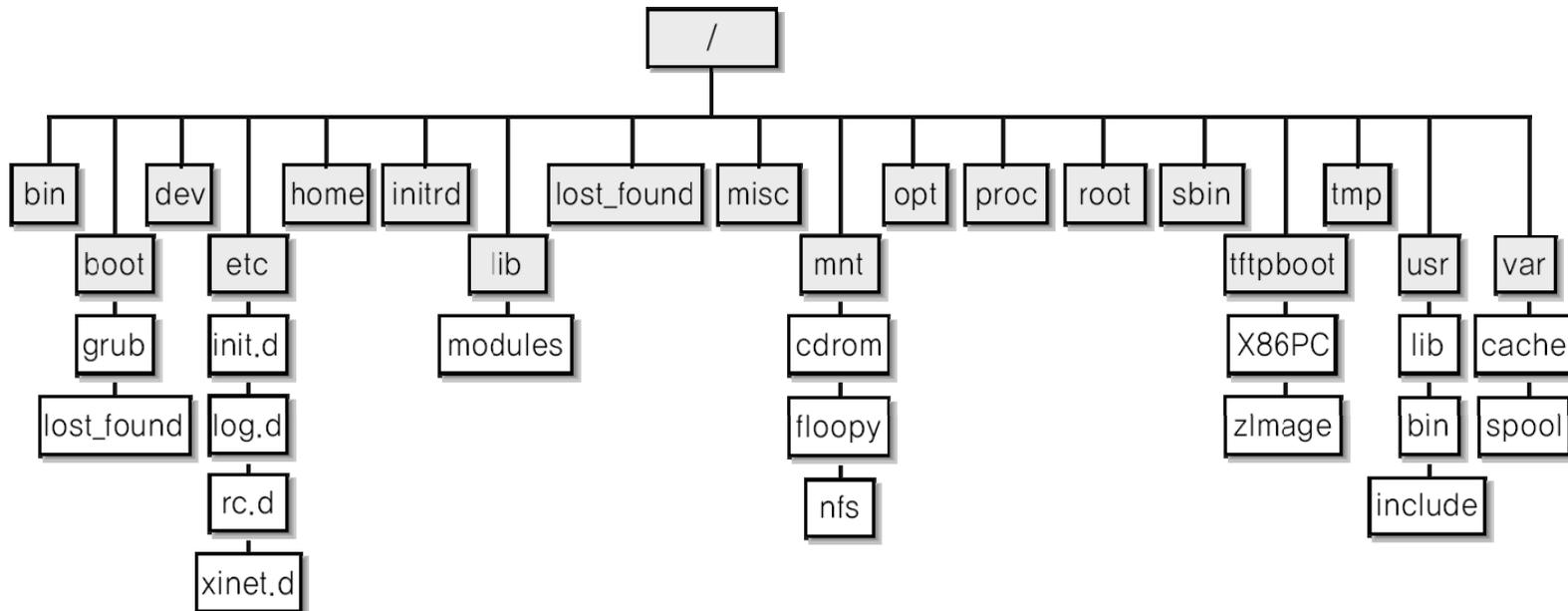


리눅스 시스템의 디렉터리 구조

- 계층적인(hierarchical) 트리(tree) 형태로 구성됨
 - 트리 구조
 - 모든 하위 파일 시스템(하위 디렉토리)이 하나의 단일 파일 구조(root directory)에 연결되어 마치 나무의 가지처럼 확장되는 형태
- 윈도우와 달리 파티션과 마운트 개념 사용
 - 루트 디렉토리부터 시작하여 각 하위 디렉토리를 계층적으로 구성하며, 각 디렉토리의 역할들은 다른 디렉토리의 역할과 확실하게 구분
 - 시스템을 설치 시 각 디렉토리마다 따로 파티션을 분할하여 리눅스의 디렉토리 구조에 마운트시켜 관리도 가능

리눅스 시스템의 디렉터리 구조

- 리눅스 커널의 소스 구조와는 다르며 리눅스 파일 구조라고 불림



리눅스 시스템의 디렉터리 구조

- / (root)
 - 루트 디렉토리(최상위 디렉토리)
 - 이 디렉토리를 기준으로 모든 디렉토리가 생성됨
 - 모든 디렉토리의 출발점이자 다른 파티션의 연결점
 - 일반적으로 데이터를 저장하지 않음
 - CD-ROM과 같은 장치의 경우 윈도우에서는 D 드라이브와 같이 순차적으로 드라이브 이름이 할당되지만, 리눅스에서 CD-ROM을 사용하려면 마운트라는 절차를 통해 / 디렉토리 아래에 붙여서 사용
 - 예) #mount /mnt/cdrom/
- /bin
 - 시스템을 사용하기 위한 기본적인 명령어 포함
 - 예) cp, mv, vi, …..

리눅스 시스템의 디렉터리 구조

- /boot
 - 부팅에 필요한 커널과 부팅할 때 필요한 핵심 파일이 들어있음
 - 부팅할 때 부트로더가 커널의 위치를 참조
- /dev
 - 장치 파일이 위치함
 - 일반 파일을 저장하지 않으며 장치 파일은 아주 특별한 경우가 아니라면 스스로 생성하지 않으므로 디렉토리의 내용은 거의 변경되지 않음.
- /etc
 - 리눅스에 사용되는 응용 프로그램과 서버 프로그램의 환경설정에 필요한 설정 파일이 들어있음
 - 실행 파일은 존재하지 않으며, 편집기를 통해 해당 파일의 변경을 통해 시스템의 설정을 변경하는 디렉토리

리눅스 시스템의 디렉터리 구조

- /home
 - 일반 사용자의 홈 디렉토리가 생성되는 곳
 - ftp, mysql, tomcat 등 별도의 공간을 요구하는 서비스를 위한 디렉토리로 사용 가능
- /lib
 - 시스템 운영 및 프로그램 구동할 때 필요한 공유 라이브러리
 - 커널 모듈
 - 일반 사용자 입장에서 /lib 내부의 파일을 함부로 삭제하거나 변경하면 안됨
- /root
 - 시스템 관리자, 즉 root 사용자의 홈디렉토리

리눅스 시스템의 디렉터리 구조

- /mnt
 - CD-ROM과 플로피 장치에 대한 기본 마운트 포인트를 제공
 - 예) #mount /mnt/cdrom (CD-ROM일 경우)
#mount /mnt/floppy (플로피의 경우)
#mount /dev/sda /mnt/usb (usb 메모리의 경우)
- /proc
 - 프로세스와 시스템 정보를 제공하기 위한 목적으로 설계된 가상 파일 시스템을 사용하는 디렉토리
 - 프로그램을 구동하게 되면 /proc에 해당 프로세스의 PID 번호에 해당하는 디렉토리를 생성하여 프로세스의 진행 상황을 추적하고 관리
 - 문자로 이루어진 이름의 디렉토리와 파일 : 시스템 및 커널 정보
 - 숫자로 이루어진 이름의 디렉토리 : 현재 실행중인 프로세스의 정보

리눅스 시스템의 디렉터리 구조

- /sbin
 - 관리자가 사용하는 시스템 운영에 필요한 명령어가 존재하는 디렉토리
 - fdisk, reboot 등
- /tmp
 - 프로세스 생성 과정에서 발생하는 임시 파일을 저장하는 디렉토리
 - 수시로 파일이 생성되고 삭제
 - 시스템을 재구동하는 경우 존재하는 파일은 모두 삭제

리눅스 시스템의 디렉터리 구조

- /lost+found
 - 개별 파일 시스템의 최상위 디렉토리마다 존재
 - 일반적인 상황에서는 이 디렉토리에 파일이 존재하지 않는 것이 정상
 - fsck(File System Check) 명령어가 사용되는 디렉토리
 - 시스템 부팅 시 파일시스템의 이상 유무를 진단하여 문제가 있으면 구동
 - 시스템을 점검 시 파일이 손상된 경우
 - 해당 파일시스템에 존재하는 lost+found 디렉토리를 이용하여 해당 데이터를 복구
 - 손상된 데이터를 복구하지 못한 경우
 - 문제가 발생한 파일시스템에 존재하는 lost+found 디렉토리에 복구하지 못한 데이터 파일이 복사됨

리눅스 시스템의 디렉터리 구조

■ /usr

- 프로그램을 설치할 때 패키지에 포함된 대부분의 파일이 이곳에 설치
- X 윈도우 관련 프로그램, 사용자 명령어, 프로그램 관련 문서, 게임, C 프로그램의 헤더파일, GNU info 파일, 라이브러리, 매뉴얼, 시스템 명령어, 커널과 패키지 소스 등

■ /opt

- add-on 패키지가 설치되는 디렉토리
- 표준 레드햇 리눅스 배포판에 존재하지 않는 패키지가 설치될 경우 이 디렉토리에 설치함
- 거의 사용하지 않지만 임베디드 시스템의 툴 체인을 보통 여기에 설치

리눅스 시스템의 디렉터리 구조

■ /var

- 내용이 자주 변경되는 가변 자료가 저장
 - 시스템 운영할 때 발생하는 로그(log)
 - 메일 송/수신할 때 임시로 저장되는 스푼(spool)
 - 프린터로 전송하기 위해 임시로 저장하는 프린트 스푼 데이터
 - 레드햇 배보판에서 패키지로 제공하는 아파치(apache) 웹 서버 관련 자료 (/var/www)
 - MySQL의 데이터 베이스
 - FTP의 최상위 디렉토리 등
 - 이러한 서비스를 제공하는 서버로 리눅스를 운영한다면 /var 디렉토리를 만드는 것이 바람직

■ /misc

- 시스템 아키텍처에 독립적인 자료들이 저장되는 디렉토리
- 거의 사용되지 않음.

참고 사항

- 리눅스와 관련된 유용한 사이트
 - <http://www.gnome.or.kr> 리눅스 그놈 사용자 한국 모임
 - <http://www.kde.or.kr> 리눅스 KDE사용자 한국 모임
 - <http://www.linux.co.kr> 리눅스 포털
 - <http://usr.gentoo.or.kr> 젤투 리눅스 한국 유저 모임
 - <http://kldp.org> 리눅스 문서 한글화 프로젝트



■ *Any Questions ?*

