

Chapter 01 리눅스 설치와 기본 사용법

목차

- 01 운영체제 개요
- 02 리눅스 기초
- 03 리눅스 실습 환경 구축
- 04 리눅스 윈도우 기본 사용법
- 05 리눅스 명령 사용법

학습목표



- 운영체제의 역할과 구성 요소가 무엇인지 설명할 수 있다.
- 리눅스의 탄생과 발전 과정을 설명할 수 있다.
- GNU 프로젝트, 리눅스 배포판, 가상머신이 무엇인지 설명할 수 있다.
- 가상머신을 설치하고 여기에 리눅스를 설치할 수 있다.
- 리눅스 윈도 환경의 메뉴 구조를 이해하고 필요한 응용 프로그램을 실행할 수 있다.
- 리눅스에서 터미널 창을 열고 기본 명령을 사용할 수 있다.

00 Preview

00 Preview

■ 1장의 내용 구성

- 리눅스 기초, 리눅스 실습 환경 구축, 리눅스 사용법의 기초로 구성
- 리눅스의 시작과 발전 과정 및 리눅스의 발전에 중요한 역할을 수행하고 있는 GNU 프로젝트를 학습
- 다양한 리눅스 배포판의 종류와 리눅스의 특징 및 구조를 살펴봄
- 실습 환경을 구축하고 리눅스의 윈도우 환경과 명령을 사용



01 운영체제 개요

01 운영체제 개요

■ 운영체제(OS)

- 컴퓨터 하드웨어와 응용 프로그램 사이의 인터페이스로, 컴퓨터 시스템의 핵심 소프트웨어

■ 운영체제의 목적

- 사용자 편의성 제공: 사용자가 쉽게 컴퓨터를 사용할 수 있도록 도움
- 자원 효율적 관리: CPU, 메모리 등의 자원을 효율적으로 운영

01 운영체제 개요

■ 운영체제의 기능

- 프로세스 관리: 프로세스 생성, 스케줄링, 관리
- 메모리 관리: 메모리 할당/해제, 가상 메모리
- 파일 시스템 관리: 파일/디렉터리 생성, 읽기, 쓰기, 삭제
- 입출력 관리: 데이터 입출력, 버퍼링, 스케줄링
- 자원 관리 및 보호: 하드웨어 자원 관리, 충돌 방지
- 사용자 인터페이스 제공: GUI 및 명령행 인터페이스 제공

01 운영체제 개요

■ 운영체제의 구성요소

- 커널: 시스템의 핵심 기능을 담당
- 파일 시스템: 데이터 저장 및 관리
- 디바이스 드라이버: 하드웨어와의 통신
- 시스템 호출: 운영체제 서비스 접근
- 인터페이스: 사용자와의 상호작용

■ CPU의 동작 모드

- 커널 모드: 하드웨어 자원에 직접 접근 가능
- 사용자 모드: 시스템 호출을 통해 자원 접근

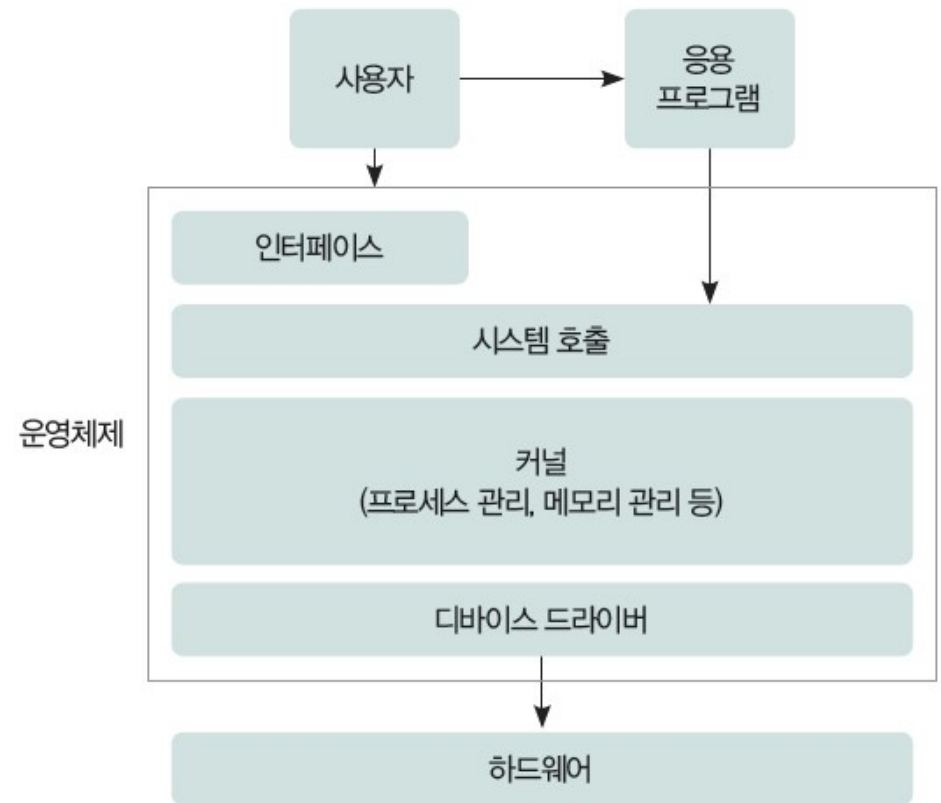


그림 1-1 운영체제 구성 요소

01 운영체제 개요

■ 커널의 구현 방식

- 모놀리식 커널: 대부분의 기능이 커널 모드에서 수행
- 마이크로 커널: 기본 기능만 커널 모드에서 수행, 나머지는 사용자 모드에서 수행

표 1-1 모놀리식 커널과 마이크로 커널 비교

	모놀리식 커널	마이크로 커널
특징	• 운영체제의 모든 기능을 포함한다.	• 운영체제의 핵심 기능만 포함한다(프로세스 관리, 메모리 관리 등). • 나머지 기능은 사용자 모드에서 제공한다.
장점	• 운영체제 구성 요소 간의 원활한 통신으로 처리 속도가 빠르고 효율성이 높다.	• 커널의 크기가 작다. • 운영체제에 기능을 추가하기 쉽다. • 일부 서비스에 문제가 있어도 운영체제 전체가 마비되지는 않는다.
단점	• 커널의 크기가 크다. • 일부 서비스에 문제가 생기면 운영체제 전체에 영향을 준다.	• 일부 시스템 기능이 사용자 모드에 있어, 통신 부담이 따르고 성능 저하 요소가 있으며 효율성이 떨어진다.
사례	리눅스, 유닉스, 윈도우 11	MacOS X

01 운영체제 개요

■ 운영체제의 역사

표1-2 운영체제의 역사

연대	주요 이슈
1940~1950년대	처음 등장한 컴퓨터인 에니악(ENIAC)에는 운영체제라고 할 만한 것이 없었으며 사용자가 직접 하드웨어를 제어하였다.
1960년대 초기	본격적인 운영체제라고 할 수 있는 일괄처리시스템이 등장하였다. 일괄처리시스템은 여러 응용 프로그램을 등록된 순서에 따라 순차적으로 실행하는 시스템이다.
1960년대 후반	시분할시스템(Time Sharing System)이 등장하였다. 시분할시스템은 CPU 시간을 작은 단위로 나누고 이를 여러 프로그램에 돌아가며 할당하여 실행하는 시스템으로, 사용자 입장에서는 동시에 여러 프로그램이 실행되는 것처럼 보인다.
1970년대	현대 운영체제의 기본 기술이 들어간 최초의 운영체제인 유닉스가 C 언어로 구현되어 빠르게 퍼져 나갔다.
1980년대	개인용 컴퓨터가 보급되고, MS-DOS가 등장하였다. 이 시기에 GUI 환경이 제공되기 시작하였다.
1990년대	네트워크 기술이 발전하면서 웹이 대중화되기 시작하였다. 리눅스가 등장하였으며 오픈소스 운동이 활성화되었다.
2000년대 이후	안드로이드, iOS 등 모바일 운영체제가 등장하였다. 또한 가상머신과 대용량 병렬처리 기술이 등장하였다.

02 리눅스 기초

02 리눅스 기초

■ 리눅스 점유율

- 데스크톱 PC: 주로 마이크로소프트의 윈도우 계열 운영체제를 사용
- 인터넷 서버와 슈퍼컴퓨터: 리눅스가 압도적인 점유율
- 모바일 기기: 리눅스 커널 기반의 안드로이드 운영체제가 70.8% 점유

■ 리눅스의 장점

- 유닉스와 호환되고, 공개 소프트웨어로 등장 당시부터 주목받음
- 다양한 환경에서 사용 가능: PC, 서버, 슈퍼컴퓨터, 임베디드 시스템, 모바일 기기
- 많은 개발자가 지원하는 소프트웨어를 포함하여 다양한 응용 프로그램 제공

02 리눅스 기초

■ 리눅스와 유닉스

- 리눅스: 유닉스 계열 운영체제, '리누스가 만든 유닉스'라는 의미
- 유닉스: 1969년 AT&T 벨연구소에서 개발, 1971년 C 언어로 재개발된 최초의 고급 프로그래밍 언어 기반 운영체제. 상용화 버전(시스템 계열)과 오픈소스 버전(BSD 계열)으로 발전
- 리눅스의 등장: 1980년대 후반, BSD 유닉스가 AT&T와 법적 공방 중 리누스 토르발스가 개발

■ 리눅스의 시작과 발전

- 핀란드 헬싱키대학교 학생인 리누스 베네딕트 토르발스가 교육용 운영체제 미닉스를 참고해 개발
- 오픈소스 운동에 합류되며 발전, 현재는 서버, 슈퍼컴퓨터, 임베디드 시스템, 모바일 기기에서 사용, 안드로이드 운영체제도 리눅스 기반
- 리눅스 커널: 1991년 8월 처음 공개, 현재 최신 버전 6.9(2024년 5월 25일 기준)

02 리눅스 기초

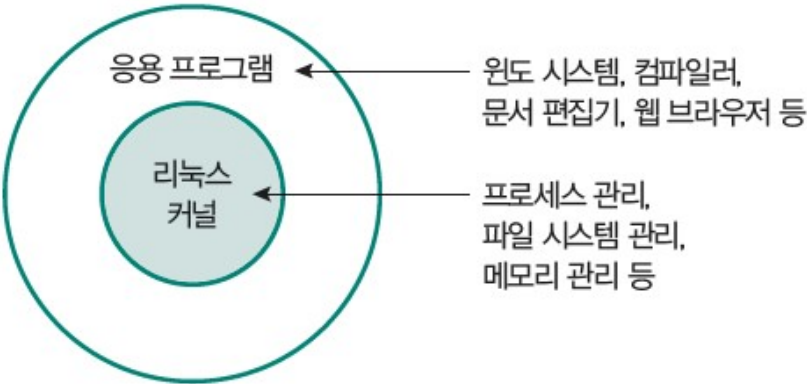


그림 1-2 리눅스 커널과 응용 프로그램

The Linux Kernel Archives

About Contact us FAQ Releases Signatures Site news

Protocol Location

HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release

6.9.2

mainline:	6.10-rc1	2024-05-26	[tarball]	[patch]	[view diff]	[browse]			
stable:	6.9.2	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	6.8.11	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	6.6.32	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	6.1.92	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.15.160	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.10.218	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.277	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.315	2024-05-25	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20240523	2024-05-23						[browse]	

그림 1-3 리눅스 커널 아카이브 웹 사이트

02 리눅스 기초

■ GNU 프로젝트와 리눅스

- 리처드 스톨먼이 1985년 <GNU 선언문>을 발표하며 시작
- 유닉스와 호환되는 자유 소프트웨어를 개발하는 프로젝트

■ GNU의 네 가지 자유

- 프로그램을 어떠한 목적으로도 실행할 수 있는 자유
- 프로그램이 어떻게 동작하는지 학습하고, 자신의 필요에 맞게 개작할 수 있는 자유. 이를 위해서는 소스 코드에 대한 접근 허용이 전제되어야 한다.
- 이웃을 도울 수 있도록 복제물을 재배포할 수 있는 자유
- 프로그램을 개선할 수 있는 자유와 개선된 이점을 공동체 전체가 누릴 수 있도록 발표할 자유. 이를 위해서도 역시 소스 코드에 대한 접근 허용이 전제되어야 한다.

02 리눅스 기초

■ GNU General Public License (GPL)

- 제정: 1989년, GNU 프로젝트 프로그램의 라이선스로 제정
- 의무
 - 프로그램을 자유롭게 사용할 수 있음
 - 실행 복사본과 함께 소스 코드를 배포
 - 소스 코드를 용도에 맞게 변경 가능
 - 변경된 프로그램의 소스 코드 공개 배포
 - 변경된 프로그램도 같은 라이선스(GPL) 적용
- GPLv1(1989), v2(1991), v3(2007): 자유 소프트웨어 라이선스, 프로그램 사용과 소스 코드 공개 등의 의무 포함

02 리눅스 기초

■ 리눅스 배포판

- 초기: 리눅스 커널과 GNU 응용 프로그램을 쉽게 구성할 수 있도록 배포판 제공
- 현재: 수백 개의 배포판 존재, 데비안, 슬랙웨어, 레드햇 계열로 구분
- 주요 배포판: 로키 리눅스(CentOS와 호환), RHEL(레드햇 엔터프라이즈 리눅스)

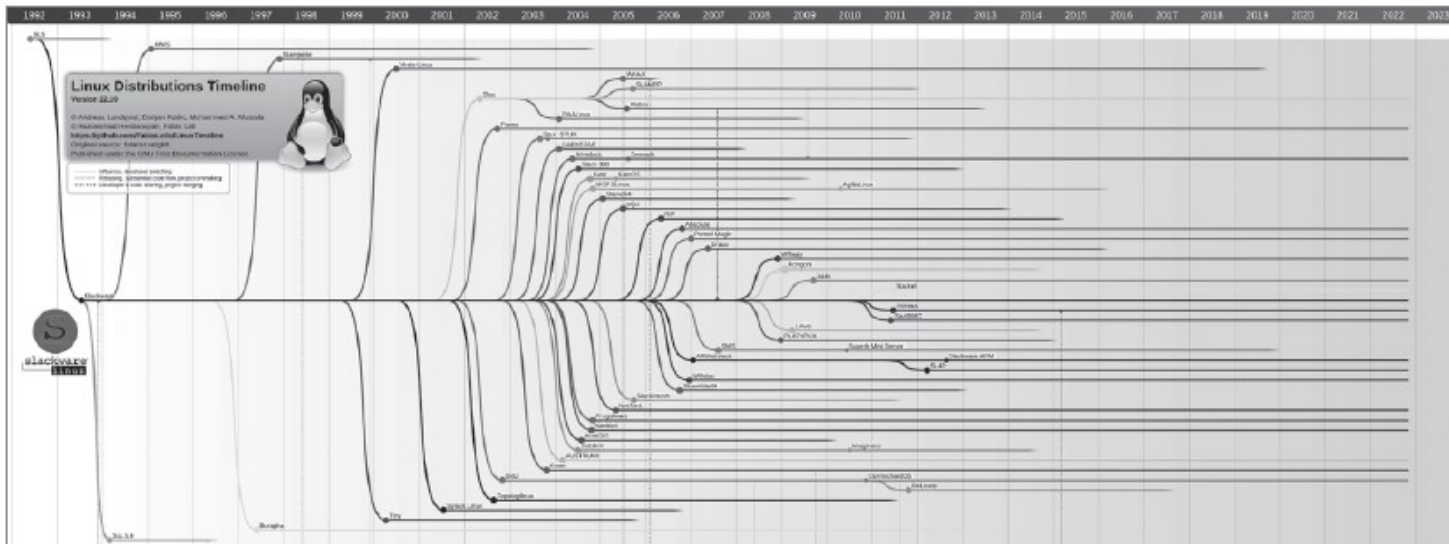


그림 1-4 리눅스 배포판의 계통도 일부(출처: <https://github.com/FabioLolix/LinuxTimeline>)

02 리눅스 기초

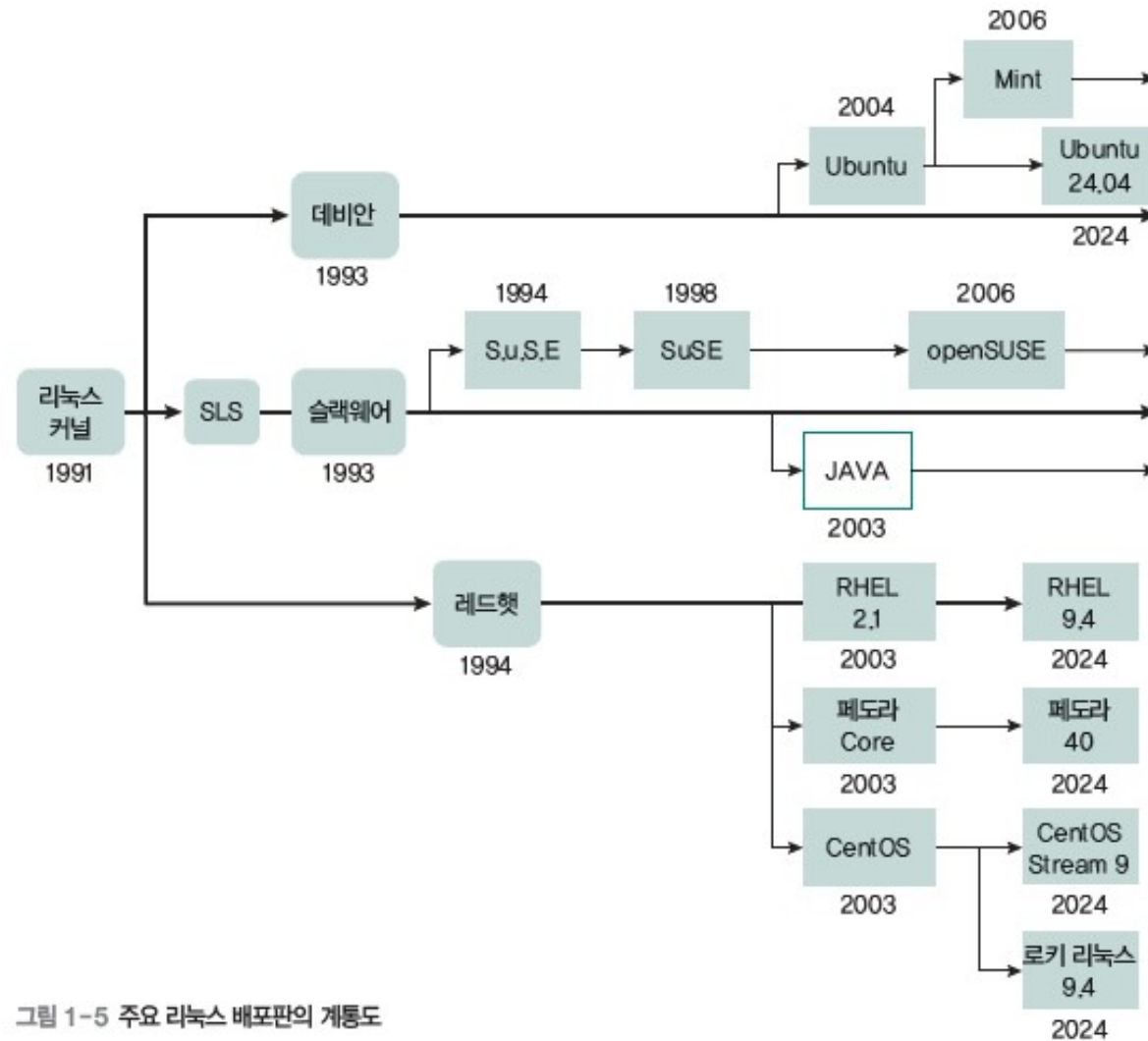


그림 1-5 주요 리눅스 배포판의 계통도

02 리눅스 기초

■ 리눅스의 특징

- 리눅스는 공개 소프트웨어이며 무료로 사용할 수 있다.
- 유닉스와 완벽한 호환성을 유지한다.
- 서버용 운영체제로 많이 사용된다.
- 편리한 GUI 환경을 제공한다.

■ 리눅스의 구조

- 커널: 프로세스, 메모리, 파일 시스템, 장치 관리
- 셸: 사용자 인터페이스, 커널과 사용자 간의 중간자 역할
- 응용 프로그램: 프로그래밍 도구, 문서 편집 도구, 네트워크 관련 도구 등

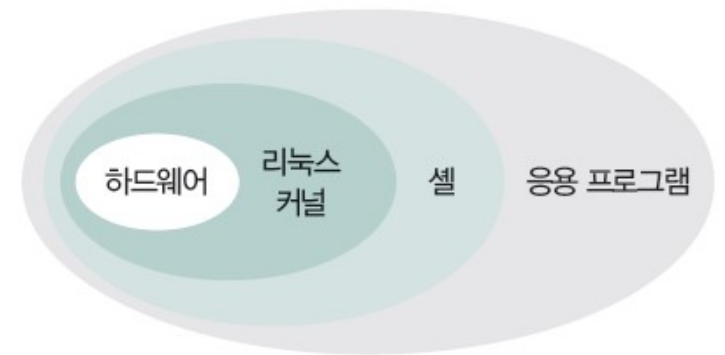


그림 1-6 리눅스의 구조

03 리눅스 실습 환경 구축

03 리눅스 실습 환경 구축

■ 리눅스 설치 개요

- 별도의 PC 없이 현재 사용하는 PC에 가상머신을 설치해 리눅스를 사용할 수 있음

■ 가상머신 개념과 종류

- 가상머신은 호스트 OS 위에 가상의 시스템을 생성해 다른 운영체제(게스트 OS)를 설치할 수 있는 응용 프로그램으로 여러 가상머신을 실행하려면 충분한 메모리가 필요함

표 1-3 가상머신의 종류

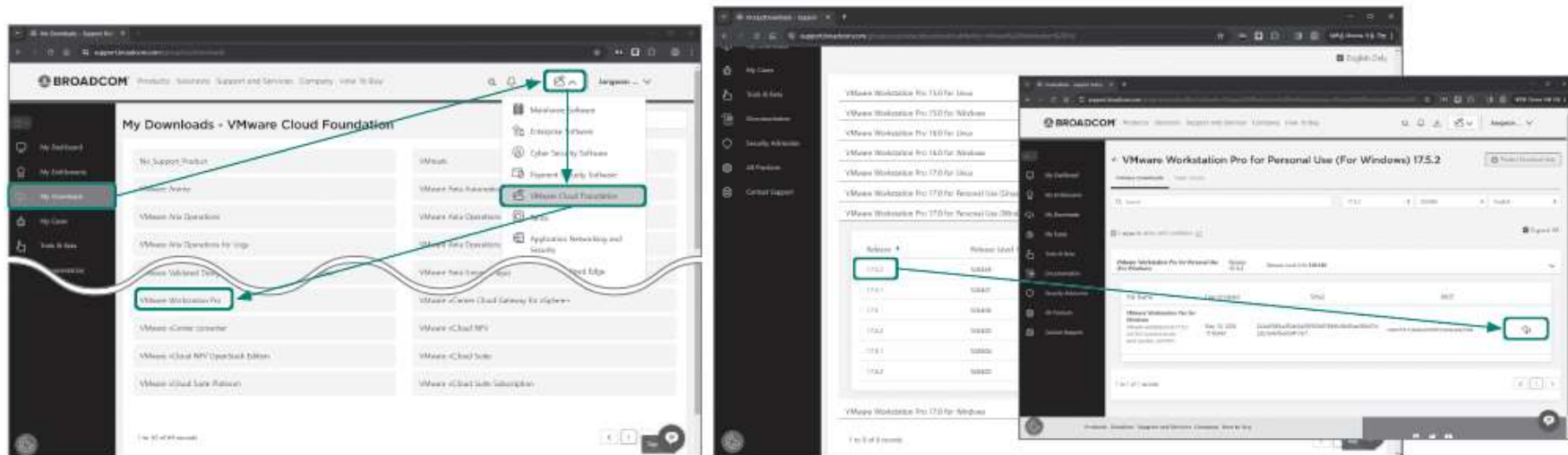
가상머신	호스트 OS	게스트 OS
VMware	윈도 계열 운영체제, 대부분의 리눅스, Mac OS	윈도 계열 운영체제, 대부분의 리눅스 배포판, 솔라리스, Mac OS
버추얼 PC	윈도 계열 운영체제	윈도 계열 운영체제, 일부 리눅스, 솔라리스
버추얼 박스	윈도 계열 운영체제, 대부분의 리눅스, Mac OS, 솔라리스	윈도 계열 운영체제, 대부분의 리눅스 배포판, 솔라리스, Mac OS, OpenBSD

03 리눅스 실습 환경 구축

■ 가상머신의 설치

① VMware 다운로드하기

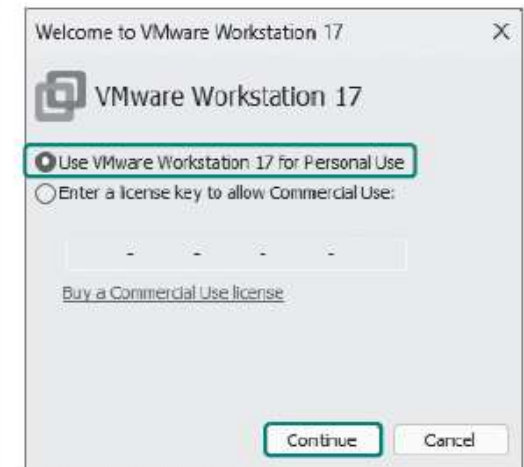
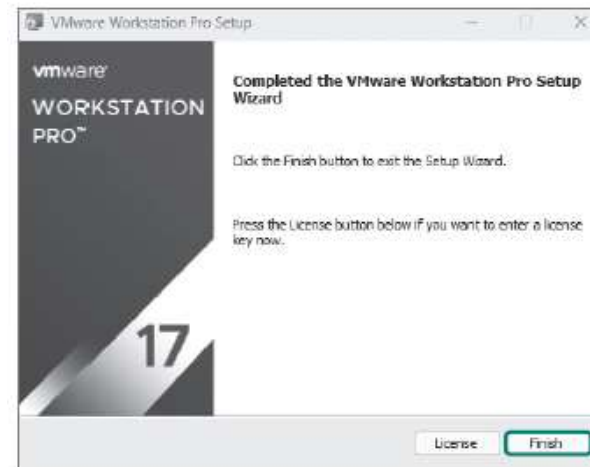
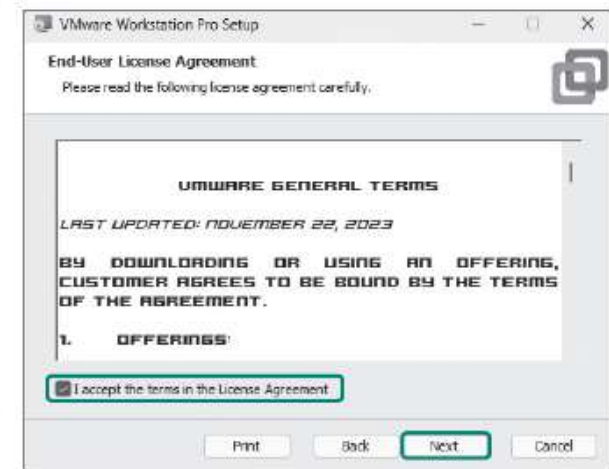
- Broadcom 사이트(<https://login.broadcom.com/>)에 접속해 계정을 등록
- 로그인 후 My Downloads에서 구름 모양 버튼 클릭, VMware Cloud Foundation 선택, VMware Workstation Pro 다운로드



03 리눅스 실습 환경 구축

② VMware 설치하기

- 일반 윈도 프로그램처럼 설치 진행
- 라이선스 동의 화면에서 'I accept the terms in the License Agreement' 체크 후 [Next] 클릭
- 설치 완료 화면에서 [Finish] 클릭.
- VMware Workstation Pro 실행 후 'Use VMware Workstation 17 for Personal Use' 선택



03 리눅스 실습 환경 구축

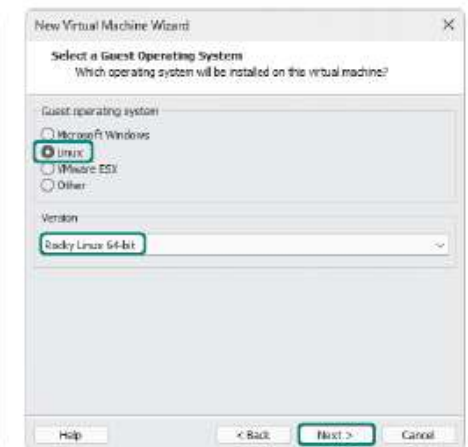
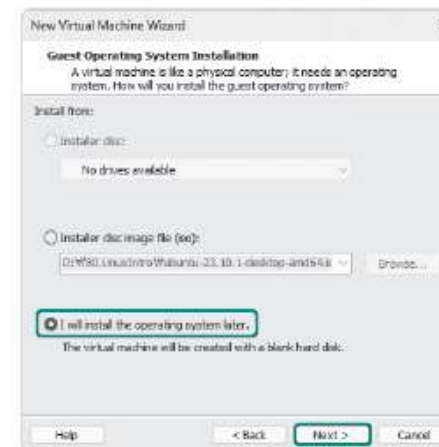
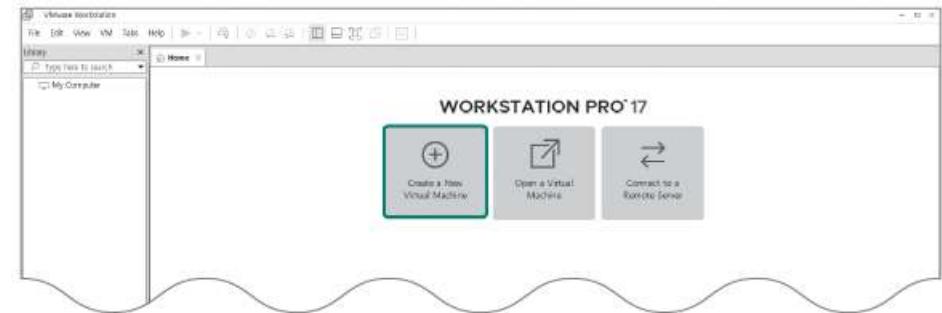
■ 가상머신 생성하기

① VMware 실행하기

- Workstation Pro 실행 후 [Create a New Virtual Machine] 선택

② 게스트 OS 설치 방법 선택하기

- 세 가지 방법 중 선택: CD/DVD 드라이브 이용 / ISO 파일 이용 / 빈 하드디스크 생성 (나중에 설치)
- 'I will install the operating system later' 선택 후 [Next] 클릭
- 설치할 게스트 OS로 'Linux' → 'Rocky Linux 64-bit' 선택



03 리눅스 실습 환경 구축

③ 가상머신 이름과 디스크 파일의 위치 지정하기

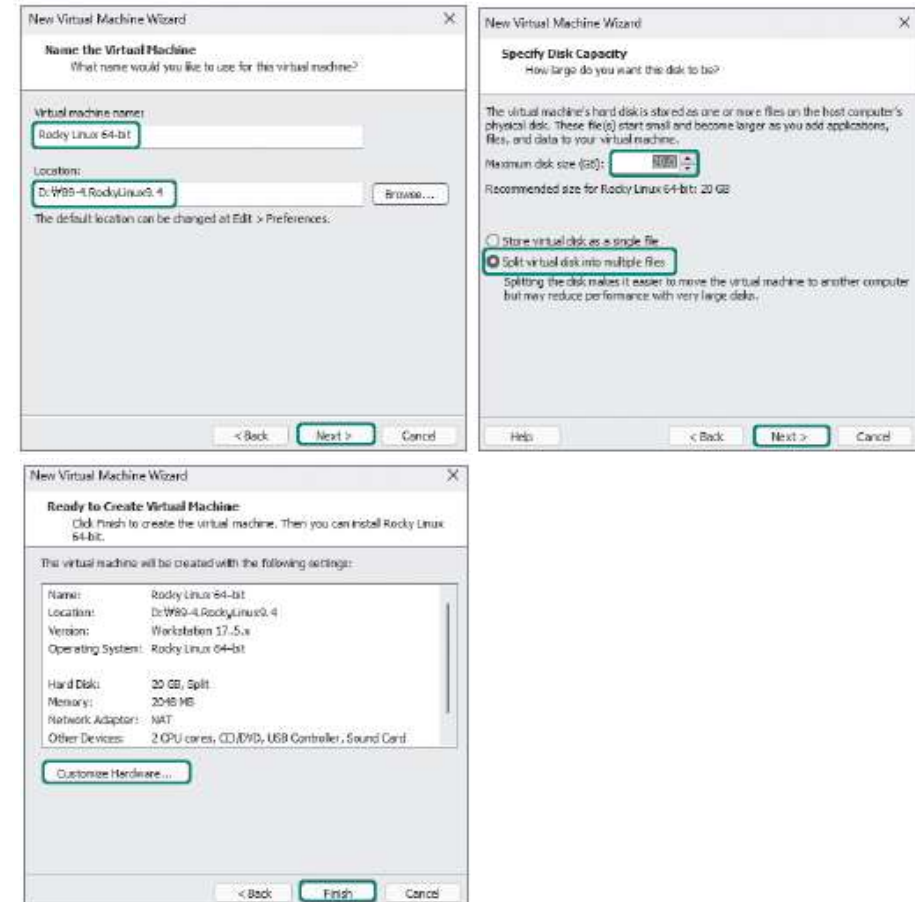
- 가상머신 이름과 폴더 지정 (예: 'Rocky Linux 64-bit')

④ 디스크 파일의 크기와 저장 방식 지정하기

- 디스크 용량 20GB로 설정, 여러 개의 파일로 저장 선택

⑤ 하드웨어 정보 수정하기

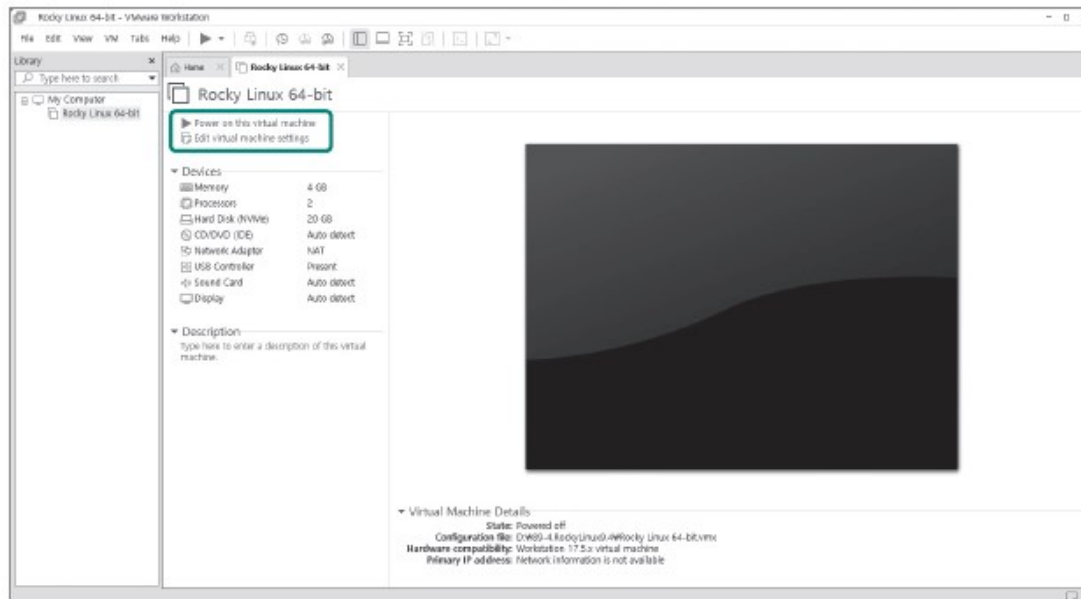
- [Customize Hardware...] 클릭, 메모리 4GB로 수정 후 [Finish] 클릭



03 리눅스 실습 환경 구축

⑥ 가상머신 생성 완료하기

- 가상머신 생성 후 [Play on this virtual machine] 클릭하여 부팅
- [Edit virtual machine settings] 클릭하여 설정 수정 가능

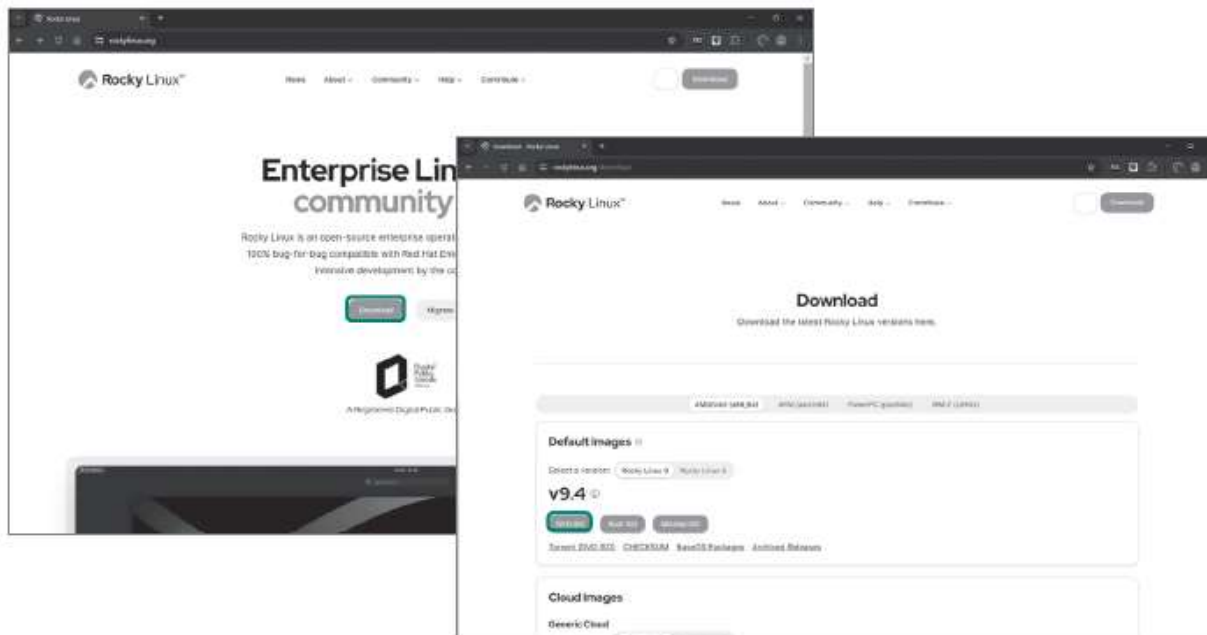


03 리눅스 실습 환경 구축

■ 리눅스의 설치

■ 로키 리눅스 다운로드하기

- <https://rockylinux.org>에서 최신 버전 다운로드



03 리눅스 실습 환경 구축

■ 리눅스의 설치

■ 로키 리눅스 설치하기

① ISO 파일 설정하고 VMware 시작하기

- VMware에서 [Edit virtual machine settings] 클릭
- 'CD/DVD(IDE)' 항목에서 'Connect at power on' 선택
- 'Use ISO image file:' 선택 후 ISO 파일 경로 지정
- [OK] 클릭 후 [Play on this virtual machine] 클릭

② 리눅스 설치 화면에서 [Install Rocky Linux 9.4] 선택



03 리눅스 실습 환경 구축

③ 설치 과정에서 사용할 언어 선택하기

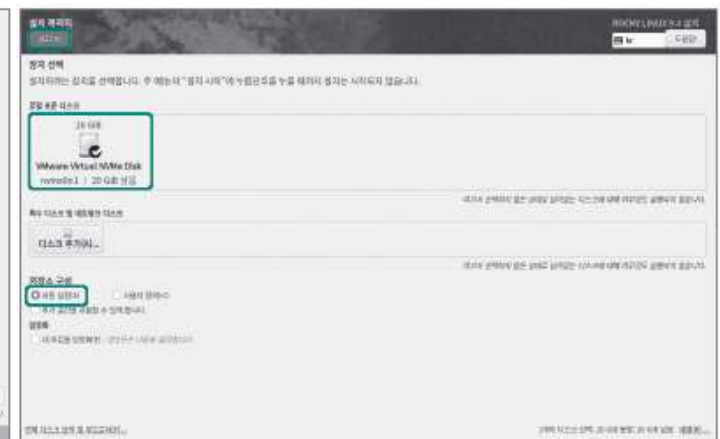
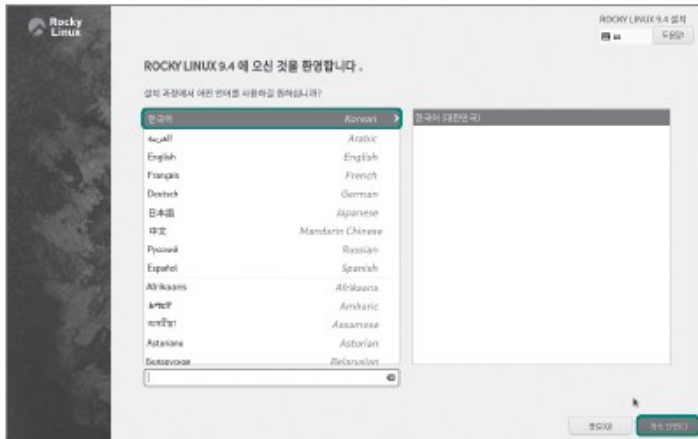
- '한국어' 선택 후 [계속 진행(C)] 클릭

④ 설치 요약 확인하기

- ! 표시가 있는 [설치 목적지D(D)] 클릭

⑤ 설치 목적지 확인하고 디스크 파티션 설정하기

- '자동 설정(A)'을 그대로 두고 [완료(D)] 클릭



03 리눅스 실습 환경 구축

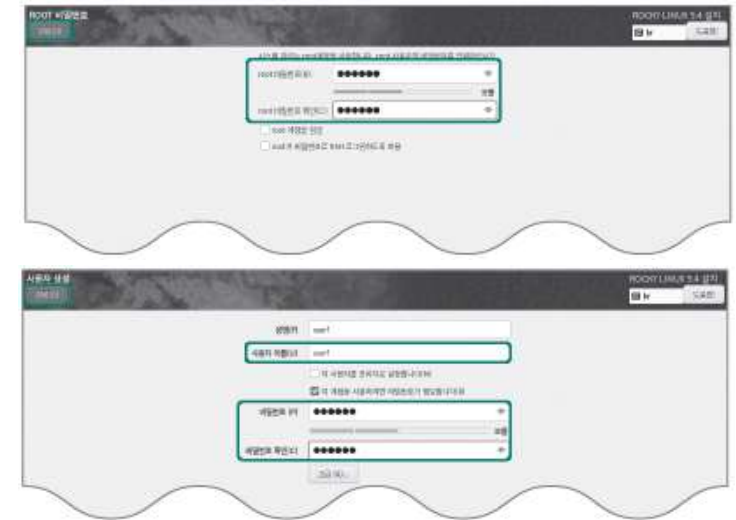
⑥ root 암호 설정하기

⑦ 사용자 계정 생성하기

- 사용자 이름에 'user1' 입력, 암호 설정 후 [완료(D)] 클릭

⑧ 설치 시작하기

- '설치 요약' 화면에서 [설치 시작(B)] 클릭



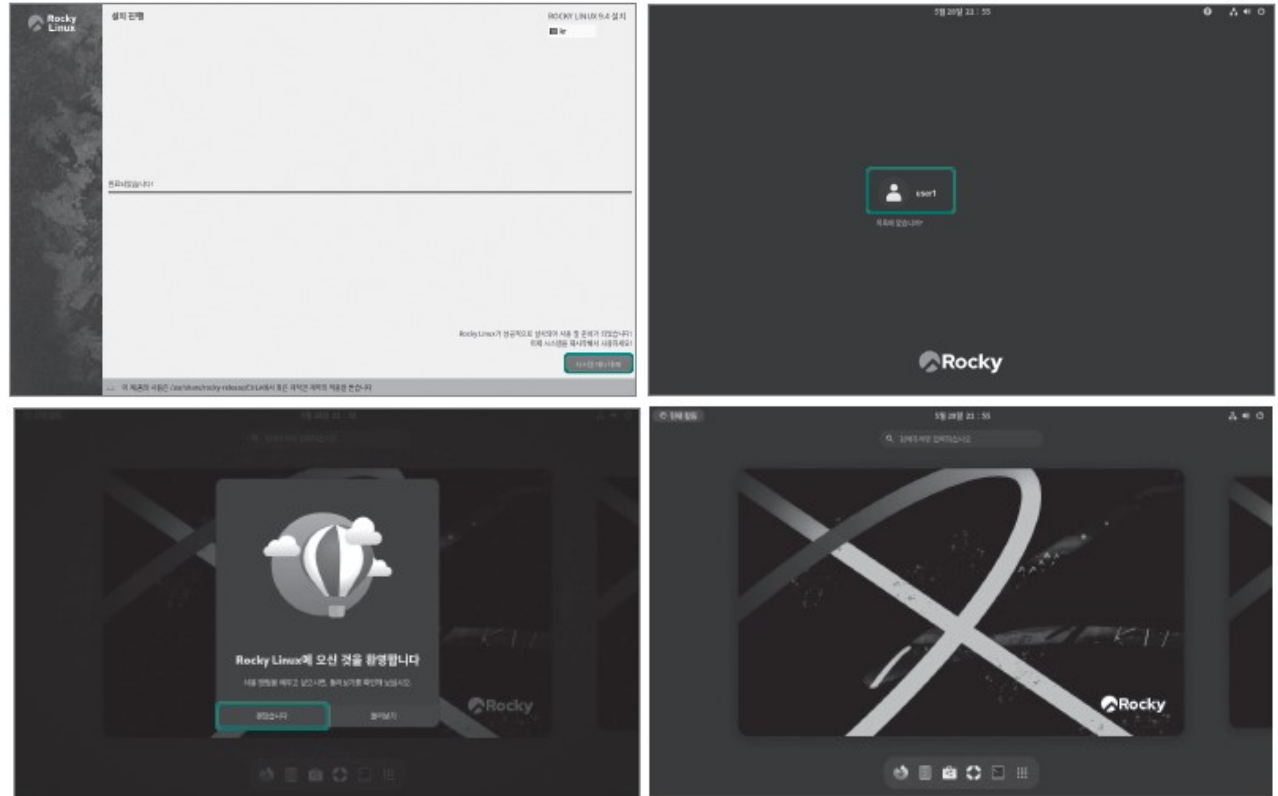
03 리눅스 실습 환경 구축

⑨ 설치 완료하기

- 패키지 설치 후 [시스템 재시작(R)] 클릭

⑩ 로키 리눅스 재시작하기

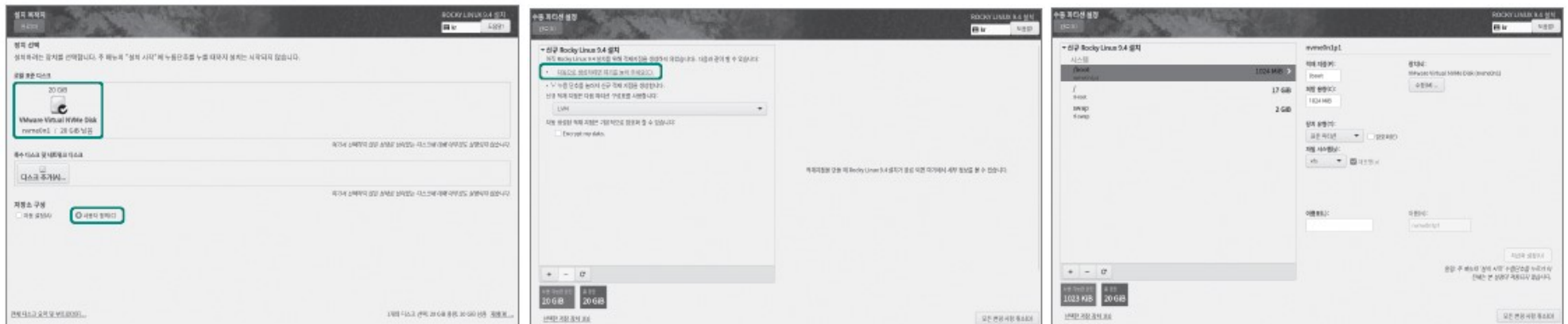
- 부팅 후 로그인 화면에서 계정 클릭, 암호 입력
- 환영 메시지에서 [괜찮습니다] 선택 후 로키 리눅스 사용 시작



03 리눅스 실습 환경 구축

■ 파티션을 수동으로 설정하기

- '설치 목적지' 화면에서 로컬 표준 디스크 선택 후 '사용자 정의(C)' 선택
- [완료] 클릭하여 '수동 파티션 설정' 화면으로 이동
- [자동으로 생성하려면 여기를 눌러 주세요(C).] 클릭하여 필수 파티션 자동 생성
- 필요에 따라 자동 생성된 파티션 수정 가능



03 리눅스 실습 환경 구축

- 리눅스 설치 시 기본적으로 필요한 파티션: root 파티션(/), boot 파티션(/boot), swap 파티션

표 1-4 리눅스 파티션의 구분 예

마운트 포인트	내용
필수	/ root 파티션으로 시스템 설정 등 주요 내용이 저장된다.
	/boot boot 파티션으로 리눅스 부팅 커널이 저장된다. 대략 500MB를 할당한다.
	swap 메모리(RAM)가 부족할 때 사용되는 영역으로 대개 RAM 크기의 2배 정도로 설정한다.
선택	/usr 명령어, 응용 프로그램 등이 주로 저장된다.
	/var 로그 파일 등이 주로 저장된다.
	/tmp 시스템 사용 중에 발생하는 임시 파일이 저장된다
	/home 사용자 홈 디렉터리가 위치한다.

- 추가 파티션 설정 (선택사항)
 - [+] 클릭하여 신규 적재 지점 추가
 - '신규 적재 지점 추가' 팝업 창에서 원하는 파티션 설정

신규 적재 지점 추가
아래 적재 지점을 생성한 후 더 많은 사용자 정의 선택을 사용 할 수 있습니다.
적재 지점(P):
희망 용량(C):

03 리눅스 실습 환경 구축

- 파티션 설정 완료 후 [완료(D)] 클릭
- [취소하고 사용자 정의 파티션 설정으로 되돌아갑니다 (C)] 클릭 → 설정 화면으로 돌아감
- [변경 적용 (A)] 클릭 → 설치 요약 화면으로 돌아감, 나머지 설치 계속 진행

변경 요약			
주 메뉴로 돌아가 설치를 시작하면 설정이 다음과 같이 적용됩니다:			
순서	동작	유형	장치
1	포맷 삭제	Unknown	VMware Virtual NVMe Disk (nvmeOn
2	포맷 생성	파티션 테이블 (MSDOS)	VMware Virtual NVMe Disk (nvmeOn
3	장치 생성	partition	VMware Virtual NVMe Disk 상의 nvme
4	장치 생성	partition	VMware Virtual NVMe Disk 상의 nvme
5	포맷 생성	physical volume (LVM)	VMware Virtual NVMe Disk 상의 nvme
6	장치 생성	lvmvg	rl
7	장치 생성	lvmiv	rl-root
8	포맷 생성	xfs	rl-root
9	장치 생성	lvmiv	rl-swap
10	포맷 생성	swap	rl-swap
11	포맷 생성	xfs	VMware Virtual NVMe Disk 상의 nvme
취소하고 사용자 정의 파티션 설정으로 되돌아갑니다(C)		변경 적용(A)	

04 리눅스 윈도우 기본 사용법

04 리눅스 원도 기본 사용법

■ 로키 리눅스 원도 환경의 구성



그림 1-7 로키 리눅스 원도의 상단 구성 요소

04 리눅스 윈도 기본 사용법

■ 로키 리눅스 윈도 환경의 구성

■ 현재 활동

- 자주 사용하는 프로그램 아이콘
- 프로그램 표시 아이콘(전체 프로그램)
- 실행 중인 프로그램 아이콘

■ 날짜와 시간 및 알림 사항 관리

■ 기타 설정

- 소리 설정, 네트워크 설정, 사용자 설정, 시스템 설정, 시스템 종료 등



그림 1-8 현재 활동

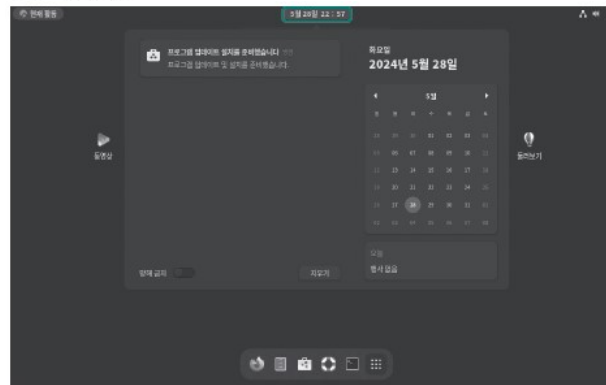


그림 1-9 날짜와 시간 설정 및 일정 확인

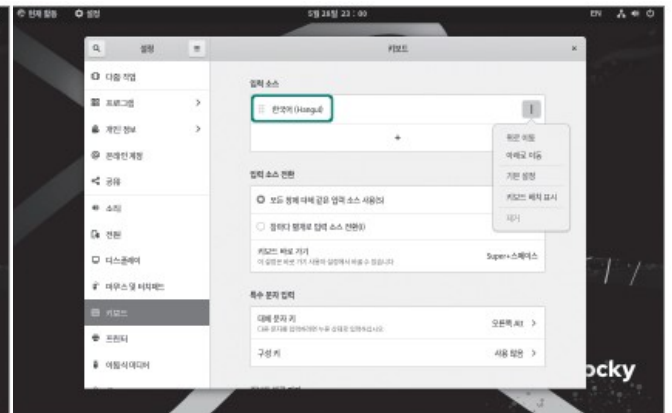
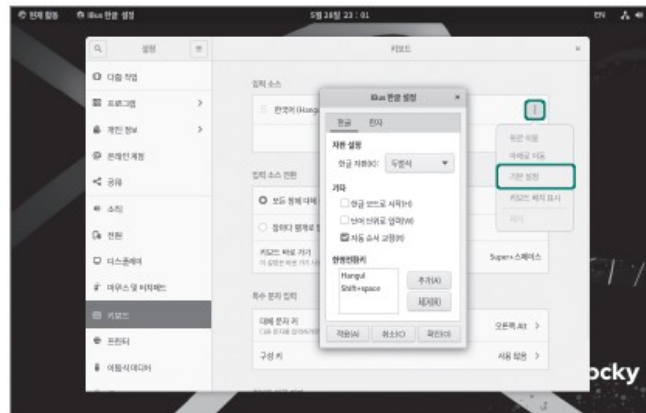
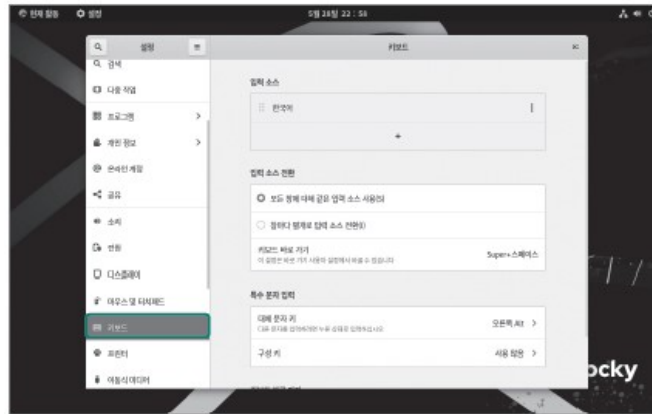


그림 1-10 소리, 블루투스, 네트워크 설정 메뉴

04 리눅스 윈도우 기본 사용법

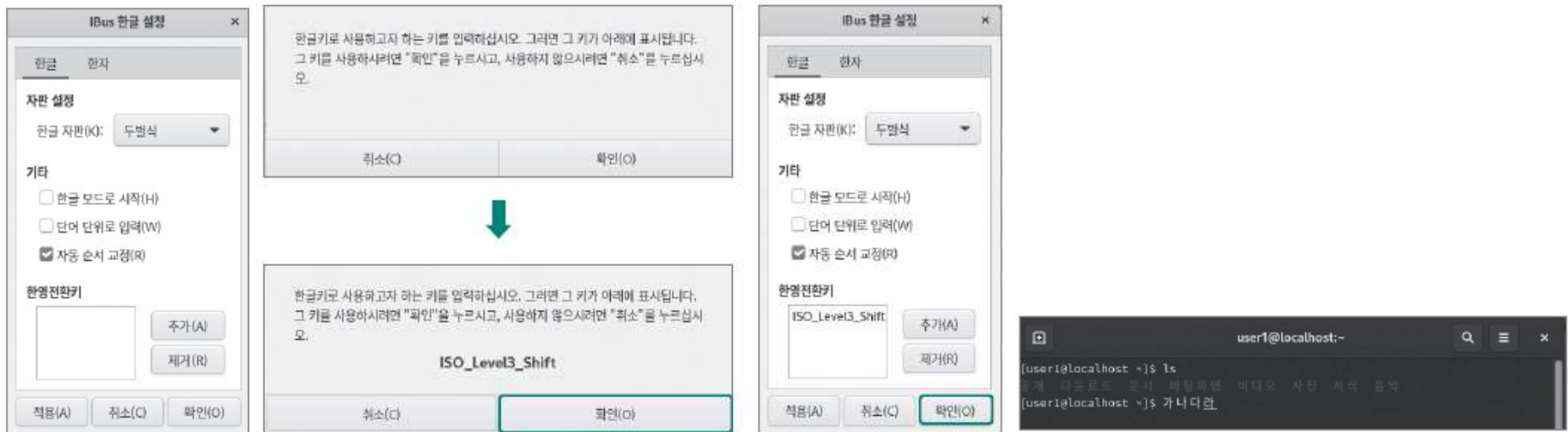
■ 한글 입력 설정하기

- ① 설정에서 [키보드] 선택하기
- ② 한국어(Hangul) 추가하기
- ③ 입력 소스 정리하기
- ④ [기본 설정] 선택하기



04 리눅스 윈도우 기본 사용법

- ⑤ 기존 한영전환키 삭제하기
- ⑥ 한영전환키 추가하고 설정 완료하기
- ⑦ 한글 입력 완료하기



04 리눅스 윈도우 기본 사용법

■ 로키 리눅스 윈도우의 사용

■ 파일

- 로키 리눅스의 파일 관리 프로그램
- [현재 활동]에 기본적으로 포함
- 윈도우 탐색기와 유사한 인터페이스 제공

■ 이미지 보기 프로그램

- 유틸리티 그룹에 속해 있음
- [현재 활동] → [프로그램 표시] → [유틸리티] → [이미지 보기] 클릭

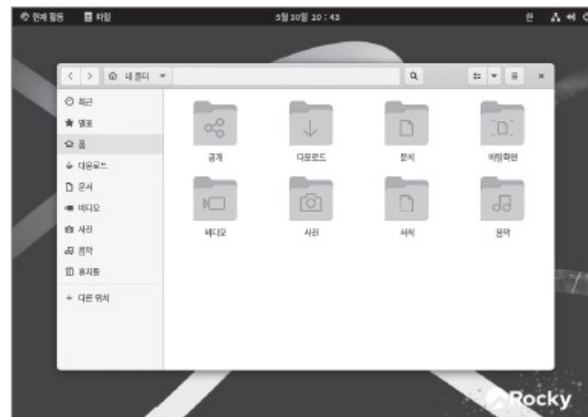


그림 1-11 파일 프로그램 창

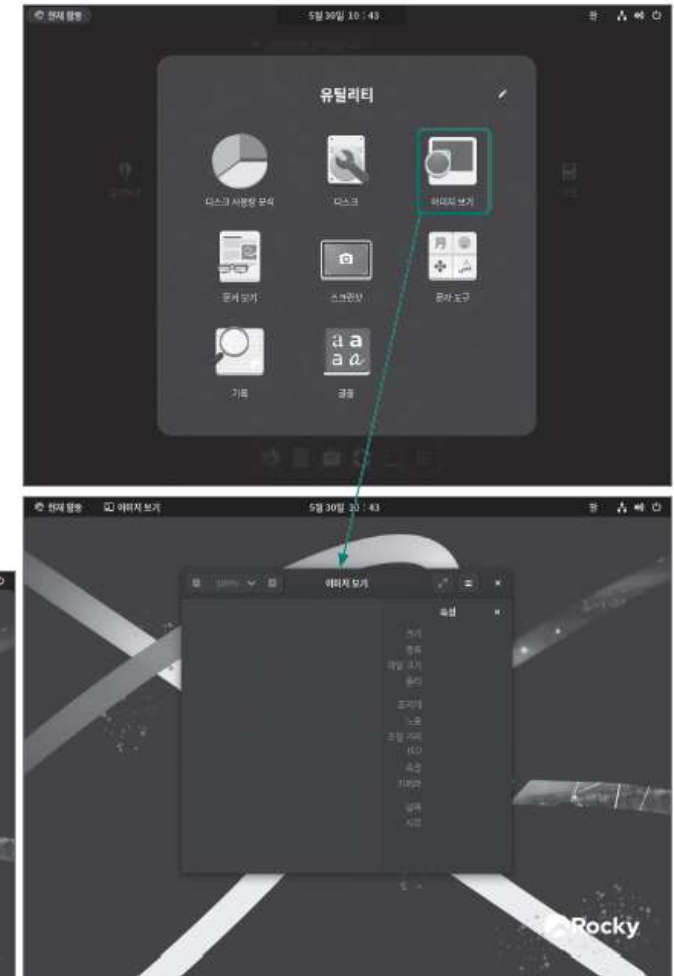


그림 1-12 이미지 보기 실행 화면

04 리눅스 윈도우 기본 사용법

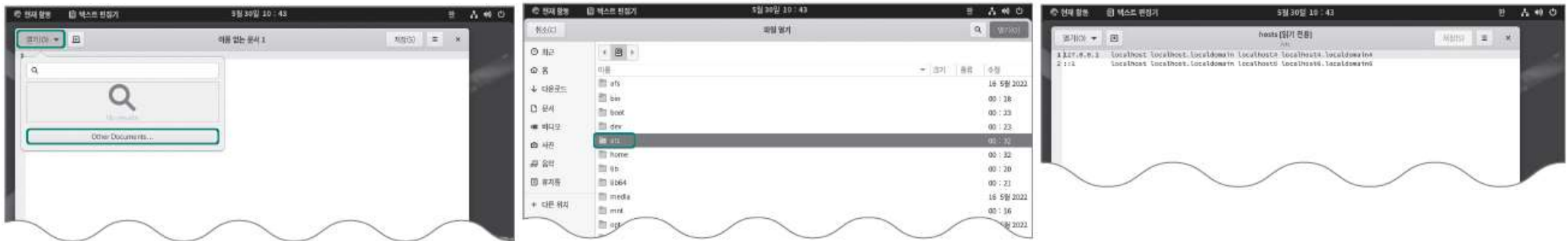
■ 텍스트 편집기로 문서 열기

① 텍스트 편집기 실행하기

- [현재 활동] → [프로그램 표시] → [텍스트 편집기] 클릭
- 텍스트 편집기(gedit)는 그놈에서 제공하는 기본 프로그램으로, 윈도우의 메모장과 유사함

② 텍스트 편집기에서 디렉터리 선택하기

③ 텍스트 편집기에서 파일 열기



05 리눅스 명령 사용법

05 리눅스 명령 사용법

■ 터미널의 시작과 종료

- 터미널 시작: [현재 활동] → [터미널] 클릭
- 터미널 종료

- 터미널 창의 오른쪽 위에 있는 [x] 클릭

- exit 명령 사용 `[user1@localhost ~]$ exit`

- [Ctrl]+d 사용 `[user1@localhost ~]$ ^d`

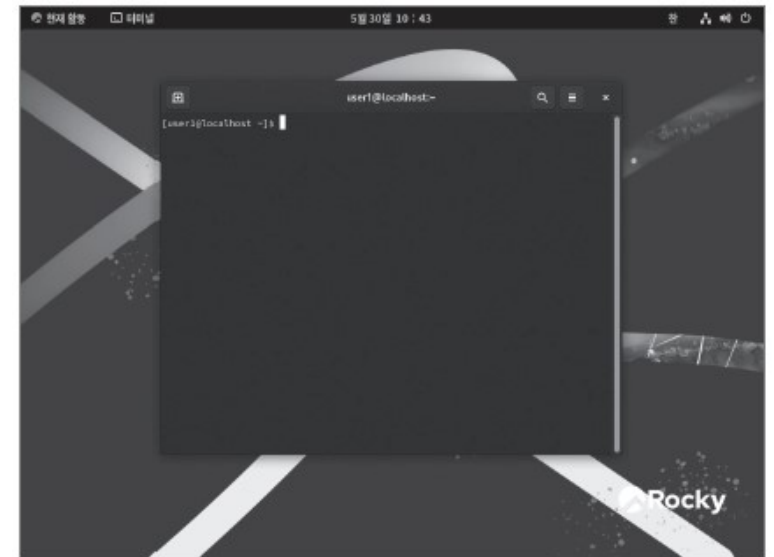


그림 1-13 터미널 창

■ 프롬프트 기호와 홈 디렉터리

- 프롬프트: 사용자의 명령 입력을 기다리는 표시
- 홈 디렉터리: 사용자 계정의 기본 디렉터리, 사용자가 파일을 생성하고 작업할 수 있는 공간

05 리눅스 명령 사용법

■ 명령행 편집

- 문자 지우기: [Backspace]나 [Delete] 사용
- 단어 지우기: [Ctrl]+w

```
[user1@localhost ~]$ linux Rocky → Rocky를 지우려면 Ctrl+w를 누른다.  
[user1@localhost ~]$ linux
```

- 행 지우기: [Ctrl]+u

```
[user1@localhost ~]$ linux Rocky → 모두 지우려면 Ctrl+u를 누른다.  
[user1@localhost ~]$
```

05 리눅스 명령 사용법

■ 명령의 구조와 사용

명령의 구조

- **형식** 명령 [옵션] [인자]
- **명령** 리눅스를 사용하기 위해 사용자가 입력하는 명령에는 `date`, `man`, `ls`, `cp`, `mv` 등 수백 가지가 있다.
- **옵션** 옵션을 사용하여 명령의 세부 기능을 선택할 수 있다. 리눅스의 기능을 풍부하게 하는 중요한 특징으로, 명령에 따라 옵션이 있는 경우도 있고 없는 경우도 있다. 옵션은 `-`나 `--` 기호로 시작하며 영문 소문자나 대문자로 구성된다. 명령에 따라 어떤 옵션이 있고 그 기능이 무엇인지를 알려면 해당 명령의 사용법을 참조해야 한다.
- **인자** 인자는 명령으로 전달되는 값이며 주로 파일명이나 디렉터리명이 사용된다. 명령에 따라 인자가 없을 수도 있고, 각 명령에 필요한 인자도 각기 다르므로 사용법을 참조해야 한다.

■ 명령만 사용한 경우

```
[user1@localhost ~]$ ls
```

```
공개 다운로드 문서 바탕화면 비디오 사진 서식 음악
```

05 리눅스 명령 사용법

- 명령과 옵션을 사용한 경우

```
[user1@localhost ~]$ ls -a
.          .bash_logout  .cache      .mozilla    문서        사진
..         .bash_profile .config     공개        바탕화면    서식
.bash_history .bashrc      .local      다운로드    비디오      음악
```

- 명령과 인자를 사용한 경우

```
[user1@localhost ~]$ ls /tmp
systemd-private-d0ad5b2340d74428be4e2e0c011c3adc-ModemManager.service-0WbbZt
systemd-private-d0ad5b2340d74428be4e2e0c011c3adc-bluetooth.service-RuAIVV
systemd-private-d0ad5b2340d74428be4e2e0c011c3adc-chronyd.service-i5BVMR
(생략)
```

05 리눅스 명령 사용법

- 명령, 옵션, 인자를 사용한 경우

```
[user1@localhost ~]$ ls -a /tmp
.
..
.ICE-unix
.X0-lock
.X1-lock
.X1024-lock
.X1025-lock
(생략)
```


05 리눅스 명령 사용법

■ 기초 명령의 사용법

■ date 명령

date

- 기능 날짜와 시간을 출력한다.
- 형식 date

```
[user1@localhost ~]$ date  
2024. 06. 08. (토) 16:32:34 KST
```

■ clear 명령

clear

- 기능 화면을 지운다.
- 형식 clear

```
[user1@localhost ~]$ clear
```

05 리눅스 명령 사용법

man 명령

man

- 기능 명령 사용 방법을 화면에 출력한다.
- 형식 man 명령

```
[user1@localhost ~]$ man clear
```

```
clear(1)                General Commands Manual                clear(1)

NAME                    → 명령에 대한 간략한 설명
    clear - clear the terminal screen

SYNOPSIS                → 명령의 사용 방법 요약
    clear [-Ttype] [-V] [-x]

DESCRIPTION            → 명령에 대한 상세한 설명
    clear clears your screen if this is possible, including its scrollback
    buffer (if the extended "E3" capability is defined).  clear looks in
    the environment for the terminal type given by the environment variable
    TERM, and then in the terminfo database to determine how to clear the
    screen.

    clear writes to the standard output.  You can redirect the standard
    output to a file (which prevents clear from actually clearing the
    screen), and later cat the file to the screen, clearing it at that
    point.
```

```
OPTIONS                → 명령의 옵션 설명
    -T type
(생략)

PORTABILITY            → 명령의 호환성과 관련된 사항
    Neither IEEE Std 1003.1/The Open Group Base Specifications Issue 7
    (POSIX.1-2008) nor X/Open Curses Issue 7 documents tset or reset.

    The latter documents tput, which could be used to replace this utility
    either via a shell script or by an alias (such as a symbolic link) to
    run tput as clear.

SEE ALSO               → 명령과 관련된 참고 사항
    tput(1), terminfo(5)

    This describes ncurses version 6.2 (patch 20210508).

clear(1)
Manual page clear(1) line 78/100 (END) (press h for help or q to quit)
```

05 리눅스 명령 사용법

▪ passwd 명령

passwd

- **기능** 사용자 계정의 비밀번호를 변경한다.
- **형식** passwd [인자]

```
user1@localhost ~]$ passwd
user1 사용자의 비밀 번호 변경 중
Current password:
새 암호:
새 암호 재입력:
passwd: 모든 인증 토큰이 성공적으로 업데이트 되었습니다.
```

Thank you!