

Linux & git 보고서

20기 인턴 송수민

목차

1. 리눅스

1.1 리눅스란?

1.2 리눅스와 윈도우 차이

1.3 cmd 명령어와 단축키

2. 우분투

2.1 우분투란?

3. git

3.1 git이란?

3.2 git 명령어

1. 리눅스

1.1 리눅스 란?

- 개요

리눅스(Linux)는 유닉스(Unix) 계열 운영체제(OS) 중 하나

1991년, 핀란드 헬싱키 대학교의 학생 **리누스 토르발스(Linus Torvalds)**가 개인 프로젝트로 개발 시작

처음에는 “자신의 386 PC에서 사용할 수 있는 간단한 운영체제 커널”을 목표로 함

이후 전 세계 오픈소스 커뮤니티와 협력하면서 발전 → 현재는 가장 널리 사용되는 오픈소스 OS

안드로이드(Android), 크롬OS(ChromeOS) 등 현대적 운영체제의 기반 기술로 확산됨

- 주요 특징

1) 오픈소스(Open Source)

GNU 일반 공중 사용 허가서(GPL, General Public License)에 따라 누구나 소스코드 사용·수정·배포 가능

특정 기업이 독점하지 않고, 전 세계 개발자 커뮤니티가 공동 관리

→ 혁신 속도가 빠르고, 필요에 따라 맞춤형 시스템 제작 가능

2) 무료(Free)

상용 OS(Windows, macOS)와 달리 무료로 다운로드 및 설치 가능

기업·교육 기관·개인 사용자 모두 자유롭게 활용 가능

일부 기업용 배포판(예: Red Hat Enterprise Linux)은 유료 지원 서비스 제공

3)멀티태스킹(Multitasking)

여러 프로세스를 동시에 실행 가능

CPU 스케줄링과 자원 관리가 효율적으로 설계됨
서버 환경, 연구용 컴퓨터, 개발 환경에서 뛰어난 성능 발휘

4) 다중 사용자(Multi-user)

여러 사용자가 동시에 같은 시스템에 접속해도 자원을 안전하게 분배
사용자 계정별 권한 체계가 철저히 구분됨
→ 서버 및 대형 컴퓨팅 환경에서 신뢰도 높음

5) 안정성 및 보안성

높은 보안성과 신뢰성을 제공 → 금융권, 공공기관, 연구소, 서버 환경에서 선호
악성 코드와 바이러스의 공격 대상이 상대적으로 적음
보안 패치와 업데이트가 활발하게 이루어짐

- 사용 예시

1) 웹 서버(Web Server)

아파치(Apache), Nginx 등 세계적으로 가장 많이 쓰이는 웹 서버 프로그램이 리눅스 기반
에서 동작
전 세계 인터넷 서버의 70% 이상이 리눅스 환경에서 운영됨

2) 슈퍼컴퓨터(Supercomputer)

2024년 기준 세계 TOP500 슈퍼컴퓨터의 100%가 리눅스 기반
고성능 컴퓨팅(HPC)에 최적화된 구조

3) 모바일 OS

구글의 안드로이드(Android)는 리눅스 커널 기반
전 세계 스마트폰의 70% 이상이 리눅스 커널을 활용

4) IoT 및 임베디드 장치

라즈베리파이(Raspberry Pi), 네트워크 라우터, 스마트 TV 등 소형 장치에도 활용
경량화된 배포판을 통해 다양한 임베디드 시스템 구현 가능

5) 데스크톱 환경

우분투(Ubuntu), 리눅스 민트(Mint), 페도라(Fedora) 등 사용자 친화적인 배포판 존재
오피스, 웹 브라우저, 멀티미디어 작업도 가능

- 개발 언어

C 언어: 리눅스 커널의 대부분이 작성됨

어셈블리어(Assembly): 하드웨어 제어 및 성능 최적화를 위해 부분적으로 사용

이러한 구조 덕분에 하드웨어 자원 활용이 효율적이고, 다양한 아키텍처(x86, ARM, RISC-V 등)를 지원

- 역사 및 발전

1991년: 리누스 토르발스가 처음으로 리눅스 커널 개발, 인터넷 뉴스그룹에 공개

1992년: GPL 라이선스로 배포 → GNU 프로젝트와 결합하여 본격적인 오픈소스 OS로 성장

1990년대 후반: 서버 시장에서 유닉스를 대체하며 빠르게 확산

2000년대: 데비안(Debian), 레드햇(Red Hat), 수세(SUSE) 등 주요 배포판 등장

2010년대 이후:

안드로이드 스마트폰 운영체제의 기반으로 자리 잡음

클라우드 서비스(AWS, 구글 클라우드, MS Azure 등)에서 사실상 표준 OS가 됨

현재:

데스크톱 환경에서도 사용 가능하지만, 주로 서버·슈퍼컴퓨터·임베디드 기기·모바일 등에서 압도적 점유율

세계적인 IT 기업들(구글, 아마존, IBM, 메타 등)이 리눅스 기반 시스템 운영

1.2 리눅스와 윈도우의 차이

구분	리눅스(Linux)	윈도우(Windows)
기반	유닉스 계열	독자적인 OS
커널	모놀리식 커널(Monolithic Kernel), 주로 C 언어	하이브리드 커널(Hybrid Kernel)
라이선스	오픈소스, 무료(GPL 등)	상용, 유료
소스 코드 공개 여부	공개	비공개
사용자 환경	개발자, 서버, 임베디드, 슈퍼 컴퓨터	개인용 PC, 사무용, 게임 등
보안성	상대적으로 강함, 멀웨어 적음	멀웨어 및 바이러스 취약
패키지 관리	패키지 매니저(APT, YUM 등) 사용	설치 파일(.exe, .msi) 사용
커스터마이징	자유롭게 수정 가능	제한적, 일부 UI만 변경 가능
시스템 요구 사항	가벼운 배포판 가능, 저사양에서도 실행 가능	상대적으로 높은 사양 필요

먼저, 리눅스는 유닉스 계열의 오픈소스 운영체제로서 누구나 자유롭게 소스 코드를 열람하고 수정할 수 있다. 이 때문에 전 세계 개발자들이 협력하여 꾸준히 개선해 왔으며, 서버, 슈퍼컴퓨터, 임베디드 장치 등 다양한 환경에서 활용되고 있다. 반면, 윈도우는 마이크로소프트에서 개발한 상용 운영체제로, 소스 코드가 공개되지 않으며 사용자가 라이선스를 구입해야만 사용할 수 있다. 따라서 윈도우는 개인용 컴퓨터와 기업 사무 환경에서 표준처럼 사용되고 있다.

또한, 보안성과 안정성 측면에서도 차이가 존재한다. 리눅스는 다중 사용자와 권한 관리 체계가 엄격하게 설계되어 있어 해킹이나 바이러스에 상대적으로 강한 편이다. 반대로 윈도우는 사용 편의성을 우선시하면서 다양한 소프트웨어와 호환되도록 설계되었기 때문에, 악성 코드의 표적이 되는 경우가 많다.

사용자 환경에서도 차이가 드러난다. 리눅스는 명령어 기반의 터미널 사용이 기본적이지만, 다양한 배포판을 통해 그래픽 사용자 인터페이스(GUI)도 지원한다. 그러나 초보자에게는 진입 장벽이 다소 높은 편이다. 윈도우는 직관적인 GUI를 중심으로 설계되어 있어 일반 사용자들이 쉽게 접근할 수 있으며, 오피스, 게임, 디자인 소프트웨어 등 폭넓은 프로그램과 호환된다.

마지막으로, 비용과 커스터마이징 측면에서 리눅스는 무료로 사용 가능하며, 사용자가 커널과 시스템을 자유롭게 수정할 수 있다. 윈도우는 유료이며 소스 코드 수정이 불가능하지만, 대신 표준화된 환경과 안정적인 기술 지원을 제공한다.

결론적으로, 리눅스는 개발자와 서버 환경에 적합한 개방적이고 안정적인 운영체제이며, 윈도우는 일반 사용자와 기업 환경에 적합한 직관적이고 상용화된 운영체제라고 정리할 수 있다.

1.3 cmd 명령어와 단축키

1) 파일 및 디렉토리 관련

pwd : 현재 작업 디렉토리 경로 출력

ls : 현재 디렉토리의 파일/폴더 목록 표시

ls -l : 상세 정보 출력 (권한, 소유자, 크기, 수정 시간 등)

ls -a : 숨김 파일까지 표시

cd [경로] : 디렉토리 이동

cd .. : 상위 디렉토리 이동

cd ~ : 홈 디렉토리 이동

mkdir [폴더명] : 새 디렉토리 생성

rmdir [폴더명] : 빈 디렉토리 삭제

rm -r [폴더명] : 폴더 및 내부 파일 삭제

2) 파일 관리

touch [파일명] : 새 파일 생성

cat [파일명] : 파일 내용 출력

less [파일명] : 긴 파일을 페이지 단위로 출력

head -n 10 [파일명] : 파일 앞부분 10줄 출력

tail -n 10 [파일명] : 파일 끝부분 10줄 출력

cp [원본] [대상] : 파일/폴더 복사

mv [원본] [대상] : 파일/폴더 이동 또는 이름 변경

rm [파일명] : 파일 삭제

3) 권한 및 소유권

chmod 755 [파일명] : 파일/폴더 권한 변경

chown user:user [파일명] : 소유자 변경

4) 검색 및 필터링

find /경로 -name [파일명] : 특정 파일 찾기

grep "문자열" [파일명] : 파일에서 특정 문자열 검색
grep -r "문자열" [폴더] : 폴더 전체에서 문자열 검색

5) 시스템 관리

df -h : 디스크 사용량 확인
du -sh [폴더명] : 특정 폴더 용량 확인
free -h : 메모리 사용량 확인
top : 실시간 프로세스 확인
htop : top의 향상된 버전 (별도 설치 필요)
ps aux : 실행 중인 프로세스 확인
kill [PID] : 특정 프로세스 종료
shutdown -h now : 시스템 즉시 종료
reboot : 시스템 재부팅

6) 네트워크

ping [도메인] : 네트워크 연결 확인
ifconfig : 네트워크 인터페이스 정보 확인
ip addr : 네트워크 IP 정보 확인
netstat -tuln : 열려 있는 포트 확인
curl [URL] : 웹 요청 테스트

7) 압축 및 묶기

tar -cvf [이름.tar] [폴더/파일] : tar 아카이브 생성
tar -xvf [이름.tar] : tar 압축 풀기
tar -czvf [이름.tar.gz] [폴더] : gzip 압축
unzip [파일.zip] : zip 파일 압축 풀기

리눅스 단축키 정리 (터미널 단축키)

1) 프로세스 및 실행 제어

Ctrl + C : 현재 실행 중인 명령어 강제 종료
Ctrl + Z : 현재 프로세스 일시 중지 (백그라운드로 이동)
Ctrl + D : 로그아웃 또는 EOF(End Of File) 입력
Ctrl + S : 터미널 출력 중지 (freeze)
Ctrl + Q : 터미널 출력 재개 (unfreeze)
Ctrl + W : 프로세스 강제 종료(SIGQUIT)

2) 화면 및 출력 관리

Ctrl + L : 화면 지우기 (clear와 동일)
reset : 터미널 환경 초기화

3) 커맨드 라인 이동

Ctrl + A : 커서 맨 앞으로 이동

Ctrl + E : 커서 맨 뒤로 이동

Alt + B : 단어 단위 왼쪽 이동

Alt + F : 단어 단위 오른쪽 이동

Ctrl + XX : 줄 맨 앞과 현재 위치 사이 전환

4) 텍스트 편집

Ctrl + U : 커서 앞의 모든 글자 삭제

Ctrl + K : 커서 뒤의 모든 글자 삭제

Ctrl + W : 커서 앞 단어 삭제

Alt + D : 커서 뒤 단어 삭제

Ctrl + H : 커서 앞 한 글자 삭제 (Backspace와 동일)

Ctrl + D : 커서 위치의 글자 삭제

Ctrl + Y : 마지막으로 삭제한 텍스트 붙여넣기(yank)

Ctrl + T : 커서 앞뒤 문자 교환

5) 명령어 히스토리

↑ / ↓ (화살표 키) : 이전/다음 명령어 탐색

Ctrl + P : 이전 명령어 실행

Ctrl + N : 다음 명령어 실행

Ctrl + R : 히스토리에서 문자열 검색 (reverse search)

Ctrl + G : 히스토리 검색 모드 종료

6) 복사 & 붙여넣기

Ctrl + Shift + C : 터미널에서 텍스트 복사

Ctrl + Shift + V : 터미널에 텍스트 붙여넣기

Shift + Insert : 붙여넣기 (대체 단축키)

마우스 드래그 : 텍스트 선택 → 자동 복사 (X11 환경 기본 기능)

마우스 가운데 버튼 클릭 : 선택한 텍스트 붙여넣기

7) 자동 완성 & 편의 기능

Tab : 명령어/파일 이름 자동 완성

Tab Tab (두 번) : 가능한 자동완성 후보 전체 표시

!! : 마지막 명령어 다시 실행

!n : 히스토리에서 특정 번호의 명령어 실행

!문자열 : 가장 최근에 실행한 해당 문자열로 시작하는 명령어 실행

8) 터미널 창 조작

Ctrl + Shift + T : 새 탭 열기 (GNOME Terminal 등)

Ctrl + Shift + W : 현재 탭 닫기

Ctrl + PageUp/PageDown : 탭 간 전환

Shift + PageUp/PageDown : 터미널 출력 스크롤

Ctrl + Shift + N : 새 터미널 창 열기

2. 우분투

2.1 우분투 란?

- 개요

우분투(Ubuntu) 는 리눅스(Linux) 계열의 오픈소스 운영체제(OS) 중 하나로, 영국의 캐노니컬(Canonical Ltd.) 이 2004년부터 개발·배포하고 있다.

사용 편의성을 강조하여 리눅스 초보자부터 전문가까지 폭넓게 활용할 수 있으며, 서버와 데스크톱 환경 모두에서 널리 사용된다.

- 주요 특징

1) 오픈소스 & 무료

누구나 자유롭게 다운로드, 설치, 사용, 배포 가능

기업이나 개인 모두 별도 비용 없이 활용 가능

2) 사용자 친화적인 인터페이스

그래픽 환경(GUI, 예: GNOME, KDE 등)을 제공하여 Windows 사용자도 쉽게 적응 가능

3) 보안성과 안정성

바이러스와 악성코드에 강함

꾸준한 보안 패치와 업데이트 제공

4) 패키지 관리

APT(Advanced Package Tool) 기반의 소프트웨어 관리 시스템 사용

apt-get, apt install 명령어로 손쉽게 프로그램 설치·삭제 가능

5) 커뮤니티 지원

전 세계적으로 활발한 사용자 커뮤니티 존재

문제 해결 자료와 튜토리얼이 풍부함

- 사용 분야

데스크톱 OS : 일반 PC 운영체제 대체 (웹 서핑, 문서 작업, 개발 등)

서버 OS : 웹 서버(Apache, Nginx), 데이터베이스 서버(MySQL, PostgreSQL) 등에서 활용

클라우드/가상화 : AWS, Azure, Google Cloud 등 주요 클라우드 플랫폼에서 지원

IoT/임베디드 : Raspberry Pi, 로봇 운영체제(ROS) 등에서 사용

- 버전 특징

우분투는 6개월마다 정기 업데이트, 2년마다 LTS(Long Term Support, 장기 지원 버전) 발표

LTS 버전은 보통 5년간 보안 업데이트 지원 → 기업·서버 환경에서 안정적으로 사용

- 이름의 의미

"Ubuntu"는 남아프리카 공용어 중 하나인 줄루어(Zulu), 코사어(Xhosa)에서 유래한 단어로
“타인에 대한 인간성”, “우리는 존재하기에 내가 존재한다”라는 뜻을 가진 철학적 개념

3. git

3.1 git이란?

- 개요

Git은 분산 버전 관리 시스템(Distributed Version Control System, DVCS) 으로, 소스 코드나 프로젝트의 변경 이력을 체계적으로 관리할 수 있게 해주는 도구이다.

2005년, 리누스 토르발스(Linus Torvalds)가 리눅스 커널 개발을 위해 만든 시스템으로, 현재는 전 세계적으로 가장 널리 사용되는 버전 관리 도구이다.

- 주요 특징

1) 분산형 구조

중앙 서버뿐 아니라, 각 개발자의 로컬 저장소에도 프로젝트 전체 이력이 저장됨
네트워크 연결이 없어도 버전 관리 및 작업 가능

2) 효율적인 버전 관리

코드 변경 내용을 "스냅샷(snapshot)" 단위로 저장
특정 시점으로 손쉽게 되돌아가거나 비교 가능

3) 협업 기능 강화

여러 개발자가 동시에 같은 프로젝트에서 작업 가능
브랜치(branch) 기능을 통해 독립적으로 개발 후 병합 가능

4) 빠른 속도와 안정성

대규모 프로젝트에서도 속도가 빠르고 안정적
리눅스 커널처럼 수십만 개 파일도 관리 가능

5) 보안성

SHA-1 해시 알고리즘을 사용하여 데이터 무결성 보장

- Git 기본 개념

저장소(Repository, Repo) : 프로젝트의 파일과 변경 이력이 저장되는 공간

커밋(Commit) : 코드 변경 사항을 저장하는 단위 (스냅샷)

브랜치(Branch) : 독립적인 작업 공간, 기능 추가나 실험에 사용

머지(Merge) : 다른 브랜치의 변경 사항을 합치는 작업

클론(Clone) : 원격 저장소를 로컬로 복사

푸시(Push) : 로컬에서 작업한 내용을 원격 저장소로 전송

풀(Pull) : 원격 저장소의 최신 내용을 로컬로 가져옴

- 사용 예시

개인 프로젝트 관리 : 버전별 코드 관리 및 되돌리기

팀 협업 개발 : 여러 명이 동시에 하나의 소프트웨어를 개발

오픈소스 기여 : GitHub, GitLab, Bitbucket 등 플랫폼을 통한 협업

3.2 git명령어

1) Git 기본 설정

git config --global user.name "이름" : 사용자 이름 설정

git config --global user.email "이메일" : 사용자 이메일 설정

git config --list : 현재 Git 설정 확인

2) 저장소 관리

git init : 현재 디렉토리를 Git 저장소로 초기화

git clone [URL] : 원격 저장소 복제

git remote -v : 원격 저장소 확인

git remote add origin [URL] : 원격 저장소 추가

git remote remove origin : 원격 저장소 제거

3) 변경 사항 관리

git status : 현재 변경 상태 확인

git add [파일명] : 특정 파일 스테이징

git add . : 모든 변경 파일 스테이징

git restore --staged [파일명] : 스테이징 취소

git restore [파일명] : 수정된 파일 원래대로 복구

4) 커밋

git commit -m "메시지" : 스테이징된 변경 커밋

git commit -am "메시지" : 수정된 파일 바로 추가 & 커밋 (새 파일은 제외)

git log : 커밋 기록 확인

git log --oneline : 간단히 커밋 기록 확인

git diff : 변경된 코드 비교

5) 브랜치

git branch : 브랜치 목록 확인

git branch [브랜치명] : 새 브랜치 생성

git checkout [브랜치명] : 브랜치 이동

git checkout -b [브랜치명] : 브랜치 생성 후 이동

git merge [브랜치명] : 현재 브랜치에 다른 브랜치 병합

git branch -d [브랜치명] : 브랜치 삭제

6) 원격 저장소와 동기화

git push origin [브랜치명] : 원격 저장소에 업로드

git push -u origin [브랜치명] : 기본 업스트림 설정 후 푸시

git pull : 원격 저장소 변경 사항 가져오기 + 병합

git fetch : 원격 변경 사항 가져오기 (병합 X)

7) 되돌리기 & 리셋

git checkout [커밋ID] : 특정 커밋으로 체크아웃

git revert [커밋ID] : 특정 커밋 되돌리기 (새 커밋 생성)

git reset --soft [커밋ID] : HEAD만 이동 (스테이징 유지)

git reset --mixed [커밋ID] : 스테이징 취소 (수정 내용 유지)

git reset --hard [커밋ID] : 완전히 해당 커밋으로 되돌림 (변경 사항 삭제)

8) 태그

git tag : 태그 목록 확인

git tag [태그명] : 현재 커밋에 태그 생성

git push origin [태그명] : 태그 원격 저장소에 푸시

git push origin --tags : 모든 태그 푸시

9) 기타 유용한 명령어

git stash : 변경 사항 임시 저장

git stash pop : 임시 저장한 변경 사항 복구

git blame [파일명] : 파일의 각 줄이 언제, 누가 수정했는지 확인

git show [커밋ID] : 특정 커밋 상세 내용 확인