

오픈소스 소프트웨어 개요 - 세부 정리

1. 오픈소스 소프트웨어(OSS)란?

- 정의:
 - 소스코드를 누구나 열람, 수정, 재배포할 수 있는 소프트웨어.
 - 오픈소스 라이선스를 준수해야 함.
- 목적:
 - 소프트웨어의 발전과 품질 향상.
 - 개발자 간 협업 및 커뮤니티 활성화.
- 주요 특징:
 - 소스코드 공개, 재사용 가능.
 - 사용자 및 개발자 모두 자유롭게 수정 가능.

2. OSS 의 철학과 역사

- 자유 소프트웨어(Free Software) 운동:
 - 리처드 스톨먼이 GNU 프로젝트와 자유 소프트웨어 재단 설립.
 - "Free"는 무료가 아니라 자유를 의미.
- 오픈소스화의 의미:
 - 자유 소프트웨어 정신을 이어받아 실용적이고 협력적인 방향으로 발전.
 - 라이선스에 따라 수정한 소스코드를 다시 공개해야 하는 경우 존재 (Copyleft).

3. OSS 개발 환경과 협업 방식

- 지원 플랫폼:
 - GitHub, GitLab, Bitbucket 등.
 - 소스코드 저장 및 변경사항 관리.
- 커뮤니티 협업:

- Pull Request 를 통해 코드 리뷰 및 병합.
- 이슈(문제점)와 개선사항 논의.
- 다양한 기여 형태: 문서화, 번역, 버그 리포트 등.

4. OSS 의 장점

- 품질 향상:
 - 많은 사람이 코드 검토 및 개선에 참여.
 - 버그 수정 및 기능 추가가 빠르게 이루어짐.
- 커스터마이징:
 - 특정 기업이나 개인의 필요에 따라 맞춤화 가능.
- 기술 발전 기회:
 - 학생이나 초보 개발자가 코드 작성 및 기여 경험을 통해 성장.

5. OSS 의 단점

- 품질 보장 부족:
 - 코드 품질과 유지보수 수준은 기여자에 따라 달라질 수 있음.
- 보안 위험:
 - 소스코드가 공개된 만큼 취약점이 악용될 가능성 존재.
- 라이선스 준수 필요:
 - 사용 및 배포 시 라이선스 조건을 위반하지 않도록 주의.

6. OSS 의 활용 사례

- 대표적인 OSS 프로젝트:
 - Python, TensorFlow, PyTorch: 빅데이터 및 AI 연구에 널리 사용.
 - Linux: 전 세계적으로 사용되는 오픈소스 운영체제.
 - Git: 분산 버전 관리 시스템.

- Mozilla Firefox, Android, OpenOffice 등.
- 웹 및 데이터베이스:
 - LAMP 스택(Linux, Apache, MySQL, PHP)은 OSS 의 대표적 성공 사례.

7. OSS 라이선스

- 주요 라이선스 종류:
 - GPL(GNU General Public License): 복제와 수정 허용, 수정한 소스코드도 공개해야 함.
 - MIT License: 간단하고 자유로운 조건, 재배포 시 원저작자 명시.
 - Apache License: 수정 및 재배포 가능, 특허 권리 포함.
- 사용 주의:
 - 프로젝트에 적합한 라이선스 선택 중요.
 - 라이선스 위반 시 법적 문제 발생 가능.