

# 오픈소스SW의 이해

## 2주차

# 1.CVCS용 소프트웨어 – Team Foundation Version Control(TFVC)

- 1.정의

-소프트웨어 개발 프로젝트의 소스 코드 및 관련 자산을 버전 관리하는 도구, 모든소스 코드 변경내용을 중앙서버에서 관리하고 다른 개발자와 협업을 지원한다.

- 2.특징

- 모든 변경 내용에 대한 버전 제어 및 변경 이력 추적을 제공하여 프로젝트의 투명성을 유지
- 엄격한 액세스 제어 및 관리로 프로젝트 보안 유지

- 3.장단점

장점

- 엄격한 권한 관리를 통해 데이터 무결성을 유지
- 중앙 서버를 통해 프로젝트를 관리하여 일관성을 유지

단점

- git과 비교하면 브랜치 , 병합 작업이 좀 더 복잡하다.
- 주로 Window환경에 최적화 되어있다.

# CVCS용 소프트웨어 –

## 2.Concurrent Versions System(CVS)

- 1.정의

-소프트웨어 개발 프로젝트의 소스 코드 및 관련 자산을 버전 관리하는 도구, 모든소스 코드 변경내용을 중앙서버에서 관리하고 다른 개발자와 협업을 지원한다.

- 2.특징

-여러 사용자가 동시에 프로젝트를 수정하고 중앙서버에 커밋이 가능하다.

-주로 텍스트 파일의 버전관리에 사용하고 바이너리 파일은 제한적이다.

- 3.장단점

### 장점

- 오픈소스로 제공되어 라이선스 비용이 없고 커스터마이징이 가능하다

-복잡한 프로젝트에서 브랜치 및 병합이 가능하다.

### 단점

-git과 비교하면 브랜치, 병합 작업이 좀 더 복잡하다.

-주로 Window환경에 최적화 되어있다.

# CVCS용 소프트웨어 -

## 3.Perforce

- 1.정의

-협업을 위한 중앙 집중식 버전 관리 시스템으로 대규모 프로젝트에서 소스코드 및 관련 자산을 효과적으로 관리하는 시스템이다.

- 2.특징

- CVS와 달리 엄격한 보안으로 데이터 무결성을 유지한다

- 특정 파일을 관리하고 병합할 수 있다.

- 3.장단점

- 장점

- 성능이 좋아서 대규모 프로젝트에 효과적이다.

- 중앙 서버를 통해 프로젝트를 관리하고 보안이 좋아 데이터의 안정성을 유지한다.

- 단점

- 사용소프트웨어로 라이선스 비용이 발생한다.

- 초기에 사용자가 학습해야 할 것이 많아 진입장벽이 높은 편이다.

# DVCS용 소프트웨어 –

## 1.Git

- 1.정의

-로컬 저장소에서 변경 내용을 추적하고 원격 저장소로 Push하여 다른 개발자와 협업하고 버전을 관리하다. 즉, 소스코드와 파일을 버전 관리하는 협업 도구이다.

- 2.특징

- 모든 개발자가 로컬저장소에서 작업하고 원격으로 변경 내용을 푸시하는 분산버전 관리 시스템이다

- 타 시스템에 비해 빠른 속도와 효율적인 성능을 제공한다 -> 대용량 프로젝트에서 효과적

- 3.장단점

- 장점

- 모든 개발자들이 오프라인 작업이 가능하다

- 빠르고 효율적인 성능으로 대규모 프로젝트에서 우수한 성능을 보장한다.

- 단점

- 이미 Push된 내용을 수정하거나 삭제하기에는 어려울 수 있다.

- 초기에 사용자가 학습해야 개념이 다양해 일부 고급기능을 활용하려면 추가학습이 필요하다.

# DVCS용 소프트웨어 -

## 2.Fossil

- 1.정의

-협업을 위한 도구로서 로컬에서 작업 후 중앙 서버로 푸시하여 협업하고 웹 기반 사용자 인터페이스와 이슈 추적 기능을 통합하여 프로젝트 관리를 위한 기능을 제공한다.

- 2.특징

- 모든 개발자가 로컬저장소에서 작업하고 원격으로 변경 내용을 중앙서버로 동기화 한다
  - 이슈 및 버그 추적을 지원해 프로젝트 관리 및 문제 해결에 도움을 준다.

- 3.장단점

- 장점

- 모든 개발자들이 오프라인 작업이 가능하다
    - 빠르고 효율적인 성능으로 대규모 프로젝트에서 우수한 성능을 보장한다.

- 단점

- 이미 Push된 내용을 수정하거나 삭제하기에는 어려울 수 있다.

- 초기에 사용자가 학습해야 개념이 다양해 일부 고급기능을 활용하려면 추가학습이

필요하다.

# DVCS용 소프트웨어 –

## 3.Mercurial

- 1.정의

-분산 버전 관리 시스템으로 git과 비슷한 기능을 제공하며 사용자 친화적인 명령어와 간단한 워크플로우로 개발자에게 편리한 버전 관리 환경을 제공한다.

- 2.특징

- 로컬 저장소에서 작업하여 중앙서버로 푸시하여 협업한다

- 오픈소스로 제공되어, 라이선스 비용이 없다.

- 3.장단점

- 장점

- 사용자 친화적인 명령어로 git보다 학습 곡선이 낮다.

- 빠른 속도와 우수한 성능으로 대규모 프로젝트에서도 효과적으로 작동한다.

- 단점

- git보다 사용자가 적어 정보가 부족하다.

- git확장성 측면에서 제한적인 부분이 있다.