

네트워크 프로그래밍 구현 part1_개발환경 분석하기

형상관리(Subversion)



한국기술교육대학교
온라인평생교육원



학습 내용

- 서브버전(Subversion) 개요 및 서버 구축
- 서브버전(Subversion) 사용법

학습 목표

- 형상관리 툴인 서브버전(Subversion)을 이용한 버전 관리를 이해하고, 서버를 구축할 수 있다.
- 형상관리 툴인 서브버전(Subversion)을 이용한 프로젝트 진행에 필요한 명령어를 사용할 수 있다.

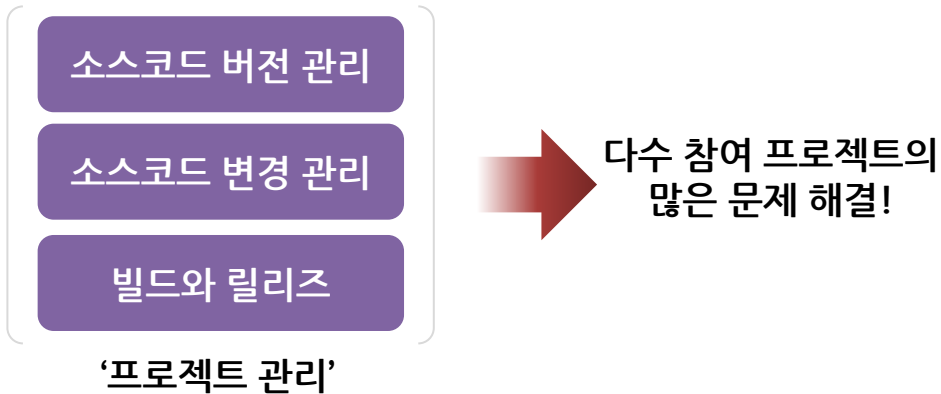


서브버전(Subversion) 개요 및 서버 구축

1. 서브버전(Subversion) 개요

1) 가상머신

- 소스코드 버전 관리에서부터 소스코드 변경 관리, 빌드와 릴리즈 관리를 포함하는 프로젝트 전반의 관리 방법을 체계적으로 정의하는 관리 방법
- 소프트웨어 소스 버전 관리를 포괄적으로 확장시켜 만든 개념
- 종류 : Subversion, Git, CVS, Mercurial, SCCS, RCS



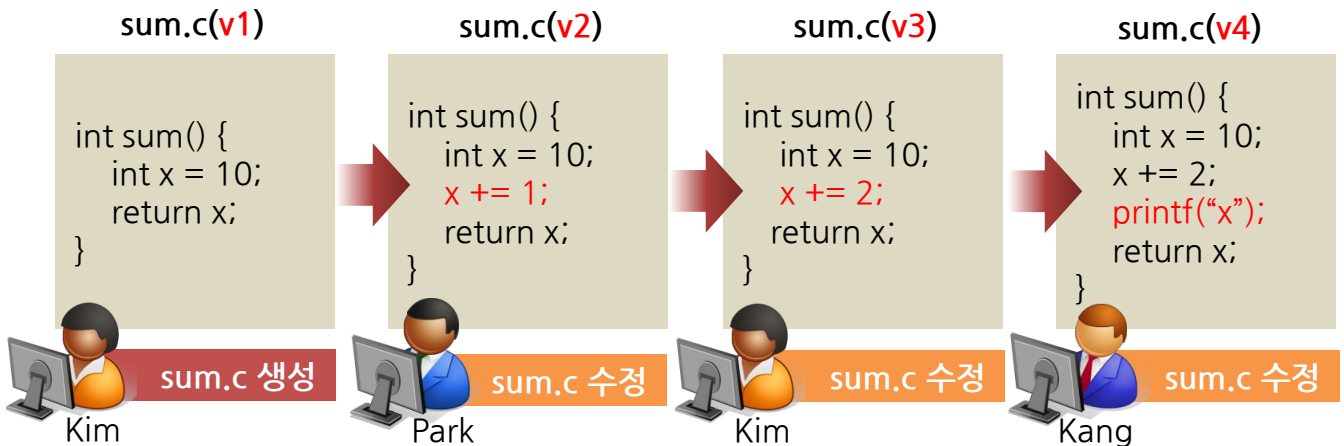


서브버전(Subversion) 개요 및 서버 구축

1. 서브버전(Subversion) 개요

2) 버전 관리 개요

- 프로젝트에서 생성되는 각종 소스, 문서, 실행 바이너리 등과 같은 파일들을 생성부터 소멸까지 체계적인 버전으로 관리하는 것
- 버전 : 버전 관리 툴에서 관리하는 버전을 의미
(파일의 생성, 수정, 소멸에 따른 버전)





서브버전(Subversion) 개요 및 서버 구축

1. 서브버전(Subversion) 개요

3) 버전 관리의 장점

잘못 수정한 내용을 이전 버전으로 되돌릴 수 있음

개발자 개인이 따로 백업하지 않아도 됨

변경 내역 추적 및 확인 가능

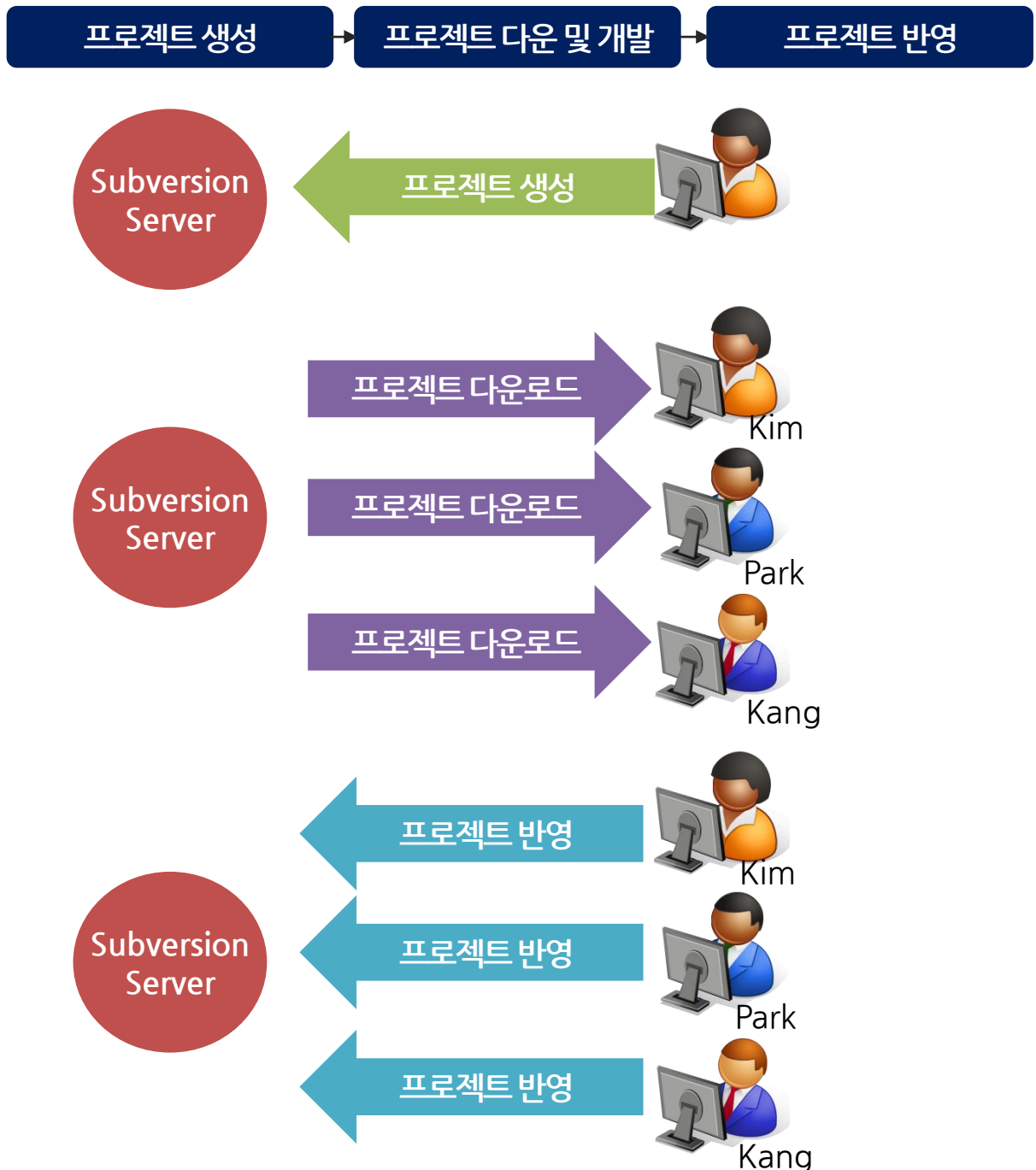
파일의 동기화와 덮어쓰기 문제가 해결됨

릴리즈 관리가 체계적

서브버전(Subversion) 개요 및 서버 구축

1. 서브버전(Subversion) 개요

4) 서브버전(Subversion)을 이용한 프로젝트 진행





서브버전(Subversion) 개요 및 서버 구축

2. 서브버전(Subversion) 서버 구축

Subversion 서버 설치

프로젝트 저장소 생성

Subversion 서버 구축

- **설치**

- # yum install subversion

- **설치 확인**

- # svn --version

- **프로젝트 저장소**

- 프로젝트 파일이 저장될 장소
 - 프로젝트 하나에 저장소 하나를 지정

- **프로젝트 저장소로 사용할 디렉토리 생성**

- # mkdir /home/svnroot

- **프로젝트 저장소 생성**

- # svnadmin create --fs-type fsfs /home/svnroot/myproject



서브버전(Subversion) 개요 및 서버 구축

2. 서브버전(Subversion) 서버 구축



■ ssh를 이용한 원격 접속

- ① Subversion 사용을 위한 그룹 생성
- ② Subversion 사용을 위한 사용자 생성 및 암호 지정
- ③ 프로젝트 저장소 권한 설정
- ④ 프로젝트 저장소 설정 확인

```

[root@korea ~]# groupadd svn
[root@korea ~]# useradd kang -g svn
[root@korea ~]# passwd kang
kang 사용자의 비밀번호 변경 중
새 암호:
잘못된 암호: 암호는 사전 검사에 실패했습니다 - 사전에 있는 단어를 기반으로 합니다
새 암호 재입력:
passwd: 모든 인증 토큰이 성공적으로 업데이트되었습니다.
[root@korea ~]# useradd shin -g svn
[root@korea ~]# passwd shin
shin 사용자의 비밀번호 변경 중
새 암호:
잘못된 암호: 암호는 사전 검사에 실패했습니다 - 사전에 있는 단어를 기반으로 합니다
새 암호 재입력:
passwd: 모든 인증 토큰이 성공적으로 업데이트되었습니다.
[root@korea ~]# chown root:svn /home/svnroot/myproject -R
[root@korea ~]# chmod g+rw /home/svnroot/myproject -R
  
```

```

[root@korea ~]# svn info svn+ssh://localhost/home/svnroot/myproject
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:UbyAEQUXCrsaAQqFJ1KsqCcp03xnT9zE8Lq3EsMhN8.
Are you sure you want to continue connecting (yes/no)? yes
root@localhost's password:
경로: myproject
URL: svn+ssh://localhost/home/svnroot/myproject
Relative URL: ^/
저장소 루트: svn+ssh://localhost/home/svnroot/myproject
저장소 UUID: 0064466d-14c4-4861-af52-3a6de6bc2502
리비전: 0
노드 종류: 디렉토리
마지막 수정 리비전: 0
마지막 수정 일자: 2015-09-25 15:13:56 +0900 (2015-09-25, 금)
  
```

```

[root@korea ~]#
  
```




서브버전(Subversion) 개요 및 서버 구축

2. 서브버전(Subversion) 서버 구축



‘프로젝트 저장소 접속 방식’



1) Subversion 서버 구축

- http를 이용한 원격 접속

- ① 아파치 웹 서버와 Subversion 모듈 설치
- ② 프로젝트 저장소 권한 설정
- ③ 아파치 웹 서버 설정
- ④ Subversion 사용자 계정 암호 생성 및 확인
- ⑤ 방화벽 설정
- ⑥ SELinux 비활성화
- ⑦ 아파치 웹 서버 재시작 및 Subversion 프로젝트 저장소 정보 확인



서브버전(Subversion) 사용법

1. 서브버전(Subversion) 환경 설정

1) 로그 작성 편집기 설정

- 환경변수 : SVN_EDITOR
- 설정파일 : \$HOME/.bash_profile

```
[root@korea ~]# vi .bash_profile
[root@korea ~]# cat .bash_profile | grep SVN_EDITOR
export SVN_EDITOR=vi
[root@korea ~]# source .bash_profile
[root@korea ~]# echo $SVN_EDITOR
vi
[root@korea ~]#
```

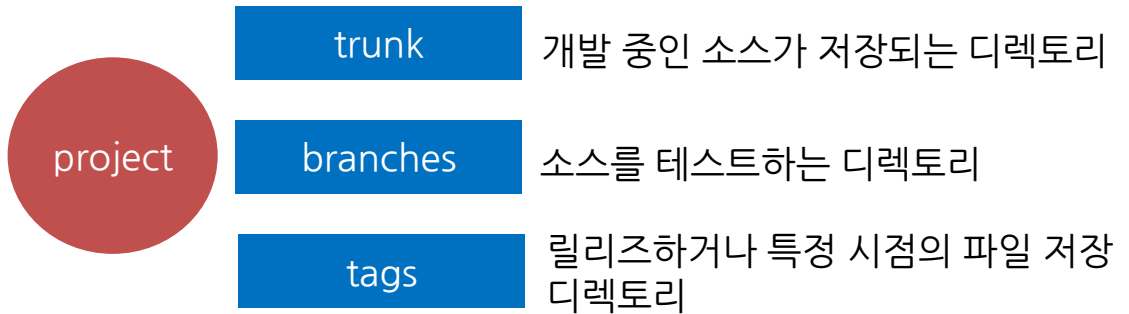


서브버전(Subversion) 사용법

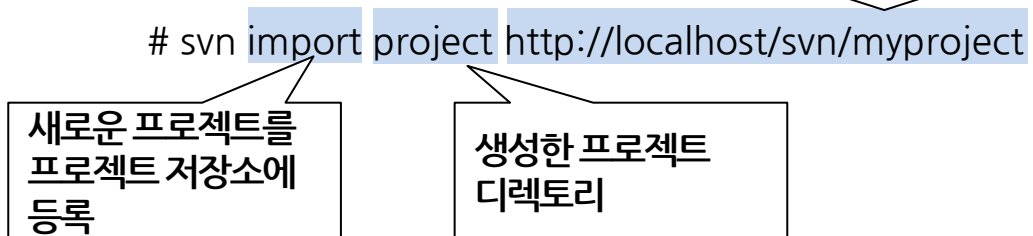
1. 서브버전(Subversion) 환경 설정

2) 프로젝트 생성

- 디렉토리 구조



- 프로젝트를 서버에 등록



- 로그 기록



서브버전(Subversion) 사용법

2. 서브버전(Subversion) 명령어

1) 프로젝트 가져오기

- 서버의 프로젝트 저장소에서 프로젝트 가져오기

\$ svn checkout http://localhost/svn/myproject

```
[root@korea svnroot]# su - kang
[kang@korea ~]$ ls -l
합계 0
[kang@korea ~]$ svn checkout http://localhost/svn/myproject
인증 영역 (realm): <http://localhost:80> MyProject
'kang'의 암호: *****

A    myproject/trunk
A    myproject/branches
A    myproject/tags
A    myproject/trunk/test.c
체크아웃된 리비전 1.
[kang@korea ~]$ ls -l
합계 4
drwxr-xr-x 6 kang svn 4096  9월 30 14:53 myproject
[kang@korea ~]$ ls -l myproject/
합계 12
drwxr-xr-x 2 kang svn 4096  9월 30 14:53 branches
drwxr-xr-x 2 kang svn 4096  9월 30 14:53 tags
drwxr-xr-x 2 kang svn 4096  9월 30 14:53 trunk
[kang@korea ~]$
```



서브버전(Subversion) 사용법

2. 서브버전(Subversion) 명령어

2) 프로젝트 수정 후 반영

- 프로젝트를 수정 후에 프로젝트 저장소에 반영하기

\$ svn commit

```
[kang@korea ~]$ cd myproject/  
[kang@korea myproject]$ cd trunk/  
[kang@korea trunk]$ vi test.c  
[kang@korea trunk]$ svn commit  
인증 영역(realm): <http://localhost:80> MyProject  
'kang'의 암호:*****
```

```
전송 중          test.c  
파일 데이터 전송 중 .  
커밋된 리비전 2.  
[kang@korea trunk]$
```

```
#include <stdio.h>  
int main() {  
    printf("hello\n");  
    return x;  
}
```



```
#include <stdio.h>  
int main() {  
    printf("hello\n");  
    printf("World\n");  
    return x;  
}
```



서브버전(Subversion) 사용법

2. 서브버전(Subversion) 명령어

3) 프로젝트 최신 내용 가져오기

- 프로젝트 저장소의 최신 프로젝트 가져오기

\$ svn update

```
[root@korea svnroot]# su - shin
[shin@korea ~]$ svn checkout http://localhost/svn/myproject
인증 영역(realm): <http://localhost:80> MyProject
'shin'의 암호: *****
```

```
체크아웃된 리비전 2.
[shin@korea ~]$ cd myproject/trunk/
[shin@korea trunk]$ vi test.c
[shin@korea trunk]$ svn commit
인증 영역(realm): <http://localhost:80> MyProject
'shin'의 암호: *****
```

```
전송 중          test.c
파일 데이터 전송 중 .
커밋된 리비전 3.
[shin@korea trunk]$
```

```
#include <stdio.h>
int main() {
    printf("hello\n");
    printf("World\n");
    return x;
}
```



```
#include <stdio.h>
int main() {
    printf("HELLO\n");
    printf("World\n");
    return x;
}
```



서브버전(Subversion) 사용법

2. 서브버전(Subversion) 명령어

4) 프로젝트 브랜치하기

- 프로젝트의 테스트 목적으로 브랜치하기

\$ svn copy trunk branches/test-1

```
[kang@korea myproject]$ svn copy trunk branches/test-1
A      branches/test-1
[kang@korea myproject]$ svn commit
인증 영역(realm): <http://localhost: 80> MyProject
'kang'의 암호: *****

추가      branches/test-1

커밋된 리비전 4.
[kang@korea myproject]$

[root@korea svnroot]# su - shin
[shin@korea ~]$ svn checkout http://localhost/svn/myproject/branches/test-1
인증 영역(realm): <http://localhost: 80> MyProject
'shin'의 암호: *****

A      test-1/test.c
체크아웃된 리비전 4.
[shin@korea ~]$
```



서브버전(Subversion) 사용법

2. 서브버전(Subversion) 명령어

5) 프로젝트 태깅하기

- 프로젝트의 릴리즈 또는 특정시점 표시를 위한 태깅하기

\$ svn copy trunk tag/release-1.0

```
[root@korea svnroot]# su - kang
[kang@korea ~]$ cd myproject/
[kang@korea myproject]$ svn copy trunk tags/ release-1.0
A          tags/ release-1.0
[kang@korea myproject]$ svn commit
인증 영역(realm): <http://localhost:80> MyProject
'kang'의 암호:*****

추가          tags/ release-1.0

커밋된 리비전 5.
[kang@korea myproject]$
```




서브버전(Subversion) 사용법

2. 서브버전(Subversion) 명령어

[기타 유용한 명령어]

리비전별 로그 확인

```
$ svn log
```

변경된 내용 비교

```
$ svn diff
```

라인 단위의 작성자 확인

```
$ svn blame test.c
```

작업 디렉토리 상태 확인

```
$ svn status
```

파일이나 디렉토리를 버전 관리 대상에 추가

```
$ svn add test1.c
```

파일이나 디렉토리를 버전 관리 대상에서 제거

```
$ svn delete test1.c
```

파일이나 디렉토리 이름 변경

```
$ svn rename test1.c test2.c
```

파일이나 디렉토리 잠금/해제

```
$ svn lock test.c / $ svn unlock test.c
```

파일 이전 상태로 되돌리기

```
$ svn revert test.c
```

파일 합치기

```
$ svn merge
```

핵심요약

1. 서브버전(Subversion)개요 및 서버 구축

1) 형상관리의 개요

- 소스코드 버전 관리에서부터 소스코드 변경 관리, 빌드와 릴리즈 관리를 포함하는 프로젝트 전반의 관리방법을 체계적으로 정의하는 관리 방법
- 소프트웨어 소스 버전 관리를 포괄적으로 확장시켜 만든 개념
- 종류 : Subversion, Git, CVS, Mercurial, SCCS, RCS

2) 버전 관리 개요

- 정의 : 프로젝트에서 생성되는 각종 소스, 문서, 실행 바이너리 등과 같은 파일들을 생성부터 소멸까지 체계적인 버전으로 관리하는 것
- 버전 : 버전 관리 툴에서 관리하는 버전을 의미
(파일의 생성, 수정, 소멸에 따른 버전)

3) 버전 관리의 장점

- 잘못 수정한 내용을 이전 버전으로 되돌릴 수 있음
- 개발자 개인이 따로 백업하지 않아도 됨
- 변경내역 추적 및 확인 가능
- 파일의 동기화와 덮어쓰기 문제가 해결됨
- 릴리즈 관리가 체계적임

핵심요약

1. 서브버전(Subversion)개요 및 서버 구축

4) Subversion을 이용한 프로젝트 진행

- 프로젝트 생성 - 프로젝트 다운 및 개발 - 프로젝트 반영

5) 프로젝트 생성 - 프로젝트 다운 및 개발 - 프로젝트 반영

6) 프로젝트 저장소 접속 방식

- 서버에서 직접 접속하는 방식
- ssh를 이용한 원격 접속 방식
- http를 이용한 원격 접속 방식
- svnserve를 이용한 원격 접속 방식

핵심요약

2. 서브버전(Subversion) 사용법

1) 로그 작성 편집기 설정

- 환경변수 : SVN_EDITOR
- 설정파일 : \$HOME/.bash_profile

2) 프로젝트를 서버에 등록

- # svn import project http://localhost/svn/myproject

3) 서버의 프로젝트 저장소에서 프로젝트 가져오기

- \$ svn checkout http://localhost/svn/myproject

4) 프로젝트를 수정 후에 프로젝트 저장소에 반영하기

- \$ svn commit

5) 프로젝트 저장소의 최신 프로젝트 가져오기

- \$ svn update

6) 프로젝트의 테스트 목적으로 브랜치하기

- \$ svn copy trunk branches/test-1

7) 프로젝트의 릴리즈 또는 특정시점 표시를 위한 태깅하기

- \$ svn copy trunk tag/release-1.0