

개발자를 위한 리눅스 유틸리티 활용법

CVS를 이용한 프로젝트 관리

연재의 마지막 시간에는 리눅스의 소스 버전 관리를 위한 툴을 소개한다. 이 툴은 흔히 형상 관리 시스템, 버전 관리 시스템이라고 일컬어진다. 윈도우나 리눅스 시스템 환경에는 여러 가지 형상 관리 시스템이 존재하는데 여기서는 현재 오픈소스로 널리 알려진 CVS에 대해 살펴본다.

4

연재 순서

- 1회 | 2007. 7 | vim 에디터 다시 보기
- 2회 | 2006. 8 | gcc 바로 알기
- 3회 | 2007. 9 | 정적 라이브러리와 공유 라이브러리
- 4회 | 2007.10 | autotool의 이해와 활용 1
- 5회 | 2007.11 | autotool의 이해와 활용 2
- 6회 | 2007.12 | CVS를 이용한 프로젝트 관리

이기영 namola78@gmail.com | 한양대학교 대학원을 졸업하고, (주)한글과컴퓨터 리눅스OS팀에서 리눅스 배포판을 개발하고 있다. 리눅스 테스크탑 연구에 관심이 많아 리눅스 테스크탑이 일반 PC에 자리 잡는 그날을 꿈꾸며 살아가고 있다.

형상 관리는 사실 하드웨어에서 비롯된 개념이다. 하지만 소스가 변경될 때마다 새로운 버전이 생성, 추가, 삭제되고 소프트웨어 개발시 발생하는 모든 작업이 시스템에 영향을 미치게 되므로, 시스템에 영향을 주는 모든 사건을 버전화하고 소스를 버전별로 관리하는 작업이 필요했는데, 이것이 바로 형상 관리(버전 관리란 용어를 사용하기도 한다)라 할 수 있다. 형상 관리는 소스뿐만 아니라 각종 파일에서도 적용할 수 있다.

CVS(Concurrent Versions System)는 오픈소스로 유닉스 및 리눅스 계열에서 가장 오랫동안 많이 사용되어 온 형상 관리 시스템이다. CVS에는 WinCVS와 TortoiseCVS 등의 클라이언트 프로그램이 존재하고 이클립스와 넷빈즈(NetBeans), J빌더(JBuilder), Idea(inteliJ) 등의 개발 툴에 기본으로 통합되어 있다.

CVS는 혼자 하는 개발, 팀을 이루는 개발, 다국적 개발 모두에 유용하며 클라이언트 서버 구조로 잠금 상태를 예약하지 않는 수정병합 방식으로 동일한 소스에 대한 동시 작업을 지원한다. CVS 소프트웨어는 오픈소스 프로젝트 내부에서 개발 중인 여러 프로그램에서 많이 활용되고 있다. 리눅스의 대표적인 데스크톱 매니저인 GNOME, KDE, XFCE 등에 많이 활용되고 있으며 배포판 개발업체인 우분투(Ubuntu), 레드햇(Redhat), 수세(Suse)에서 패키지 버전을 관리하는 데도 널리 쓰이고 있다. 또

한 CVS의 소스 버전 관리 역시 여러 명이 공동 작업을 하는 데 무척 유용한 것으로 평가된다.

CVS 설치 방법

우선 xinetd가 설치되었는지 확인해야 한다. 리눅스(레드햇)에서 CVS 서버를 설치하고 관리하는 방법에 대해 알아보자. 설치된 리눅스에 CVS가 설치되어 있는지 확인하기 위해 시스템 루트 계정으로 로그인한 후 다음 명령어를 입력한다.

```
# cvs -v or # cvs --version
```

CVS가 설치되어 있다면 설치된 CVS에 대한 정보가 나온다. 또는 cvs rpm이 설치되어 있는지 확인할 수 있다. 확인 방법은 다음과 같다.

```
rpm -qa | grep cvs
```

설치되어 있지 않다면 <http://ftp.gnu.org/non-gnu/cvs/binary/stable/x86-linux/RPMS/i386/>에서 패키지를 다운로드할 수 있다. 단 이 주소는 x86 시스템에서 rpm 패키지를 제공한

다. 최신 버전의 rpm 패키지 버전을 다운로드해 다음과 같이 설치한다. 단 debuginfo의 rpm 패키지는 설치할 필요가 없다.

- 새로 설치한 경우 : rpm -ivh cvs*
- 업데이트할 경우 : rpm -Uvh cvs*
- 설치된 정보를 확인하는 방법 : rpm -qi cvs

x86 시스템이 아닌 다른 아키텍처를 원할 경우에는 <http://ftp.gnu.org/non-gnu/cvs/>에서 다운로드할 수 있다. 설치가 끝난 후 # which cvs를 실행하면 다음과 같이 나타난다.

```
/usr/bin/cvs
```

만일 데비안이나 우분투라면 전용 패키지 관리자를 이용해 설치하면 된다. 패키지 설치가 여의치 않다면 소스를 직접 컴파일해서 설치하면 된다. 단 직접 설치할 때는 소스에 포함되어 있는 README 파일을 읽어 보길 바란다.

저장소 설정

여러 명이 함께 작업할 수 있는 환경을 제공하려면 어떠한 요소가 필요할까? 가장 먼저 떠오르는 것은 함께 작업하는 파일을 보관할 장소다. CVS에서 이러한 역할을 하는 곳을 저장소(repository)라 부른다. 그렇다고 특별한 데이터베이스나 다른 저장 매체를 사용하는 것이 아니라 이는 그저 하나의 디렉토리에 불과하다. 이 디렉토리에는 CVS의 전반적인 설정사항과 각 프로젝트의 파일들(문서, 프로그램 등)은 물론이고, 각 파일의 버전 관리에 필요한 정보, 파일별 작업 기록들이 저장된다. 여기에 있는 파일들은 모두 CVS가 관리하므로 사용자는 이 파일들을 직접 건드릴 필요가 전혀 없다. CVS 설치를 완료했다면 저장소를 설정해 줘야 한다.

저장소로 사용할 디렉토리를 /home/cvs 디렉토리라고 한다면 다음과 같은 과정으로 저장소를 설정할 수 있다. 물론 root 권한으로 로그인한 상태여야 한다. /home/cvs를 프로젝트들을 위한 저장소로 사용한다고 했는데, 저장소로 사용하기 전에 저장소 터를 다지기 위한 사전 작업이 필요하다.

```
# cd /home/ - /home/ 디렉토리로 이동한다.
# mkdir cvs - /home/ 디렉토리 아래에 cvs 디렉토리를 생성한다.
# cvs -d /home/cvs init - /home/cvs 디렉토리를 CVS 저장소로 초기화한다. -d 옵션은 저장소가 될 디렉토리 경로를 지정
```

하기 위한 옵션이고, init는 저장소를 초기화하기 위한 cvs 명령어이다.

이 과정을 거친 후 /home/cvs 디렉토리를 살펴보면 CVS ROOT라는 이름의 디렉토리가 생성되어 있음을 확인할 수 있다. 이것은 정상적으로 초기화된 것으로, 이곳에 CVS의 설정 파일이 존재한다.

사용자 관리

사용자 관리는 다음과 같이 두 가지 방법으로 나눌 수 있다.

- CVS가 설치된 서버 시스템(리눅스)에 클라이언트 계정을 모두 등록해 놓고 사용하는 경우
- CVS 서버에만 접근할 수 있는 계정을 사용하는 경우

두 번째 방법을 사용한다면 CVS 서버에 접근할 수 있게 하기 위해 해당 리눅스 시스템에 일일이 계정을 추가해 두지 않아도 되는 장점이 있다. 오픈 프로젝트의 경우 이 방법을 사용한다. 사내 개발 시스템을 위해 CVS를 도입하는 경우에는 두 번째 방법보다는 첫 번째 방법이 효율적일 것이다. 그 이유는 혼자 CVS 서버에만 접근 가능하게 설정해 두면 리눅스 시스템의 활용도가 크게 떨어지기 때문이다.

참고로 첫 번째 방법으로 관리하는 경우 클라이언트는 ssh나 rsh 프로토콜로 접근이 가능하며, 두 번째 방법으로 관리하는 경우에는 pserver 프로토콜로 접근할 수 있다.

리눅스 시스템에 클라이언트 계정을 만들고 그 계정을 통해 CVS 서버에 접근하도록 관리하는 방법

우선 리눅스 시스템에 클라이언트로 접속하게 될 사용자 계정을 추가해 줘야 한다. 차후의 설명을 위해 user01, user02 등 두 개의 사용자 계정을 추가해 보도록 한다.

- 리눅스 사용자 계정 추가


```
# useradd user01 - root 권한으로 로그인한 후에 실행해야 하며 user01이라는 이름의 계정을 추가하기 위한 것
# passwd user01 - 방금 추가한 사용자 계정 user01의 패스워드를 설정하기 위한 명령.
user02라는 이름의 계정도 이와 같은 방법으로 추가한다.
```

- cvs 접근 그룹 추가

cvs 서버에 접근할 수 있는 사용자를 계정별로 관리하기보다

는 그룹으로 관리하는 편이 훨씬 더 편하다. 그럼 cvs라는 이름으로 그룹을 추가하도록 하자. 그룹을 추가하는 명령어는 groupadd이다.

```
# groupadd cvs
```

cvs라는 그룹을 추가하게 되면 /etc/group 파일에 다음과 같은 내용이 추가된다.

```
cvs:x:501
```

이번에는 앞서 추가한 사용자 계정 user01과 user02를 cvs 그룹에 추가한다. 특정 그룹에 사용자 계정을 추가하는 명령어는 usermod이다.

```
# usermod -G cvs user01
```

```
# usermod -G cvs user02
```

user01과 user02 사용자를 cvs 그룹에 추가시키면 /etc/group 파일의 내용이 다음과 같이 변경된다.

```
cvs:x:503:user01,user02
```

● 권한 설정

cvs에 접근하기 위한 사용자 계정과 그 사용자 계정들을 하나로 묶어두기 위한 사용자 그룹 만들기가 완료되었다. 이제 사용자 각각이 아니라 사용자 그룹에게 CVS를 사용할 수 있는 권한을 설정해 줄 차례이다. 다음 명령을 써서 저장소 디렉토리 (/home/cvs/)와 관련해 cvs 그룹에게 권한을 열어 준다. cvs 그룹에 포함되어 있는 사용자 계정은 cvs 그룹이 가지는 권한을 그대로 물려받게 될 것이다.

```
# chgrp -R cvs /home/cvs
```

```
# chmod 770 /home/cvs /home/cvs/CVSROOT
```

CVS 서버에만 접근할 수 있는 계정 관리

CVS 접근 사용자 관리의 또 다른 방법으로 CVS 서버에만 접근 가능하도록 CVS 전용 계정을 관리하는 방법을 살펴본다.

● 서비스 설정

CVS 전용 계정으로 관리하는 경우 클라이언트는 CVS 서버에

pserver 프로토콜을 사용해 접근한다. 이 프로토콜을 사용할 수 있으려면 몇 가지 준비사항이 필요하다. 우선 pserver 프로토콜이 사용하는 포트가 사용 가능하게 설정되어 있는지를 확인해야 한다.

/etc/services 파일에는 서비스와 포트 번호, 프로토콜에 대한 내용이 기록되어 있다. 이 중에서 2401 포트에 대한 설정을 찾아야 한다. vi 에디터나 cat 명령어를 사용해 services 파일의 내용을 일일이 확인할 수도 있겠지만 다음과 같은 간단한 명령어로 2401 포트에 대한 설정을 확인할 수 있다.

```
# cat /etc/services | grep 2401
```

이 명령어의 결과로 내용이 나타난다면 pserver 프로토콜을 사용할 준비가 되어 있는 것이다(cvs로도 확인 가능).

```
# cat /etc/services | grep cvs
```

체대로 cvs 서비스가 이뤄지고 있는지 알아보기 위해 포트 스캐닝 도구인 nmap를 이용해 확인할 수 있다.

```
# nmap 192.168.0.4
```

```
Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on localhost.localdomain (127.0.0.1):(The 1527
ports scanned but not shown below are in state: closed)
```

```
Port      State Service
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
111/tcp   open  sunrpc
2401/tcp   open  cvspserver
3306/tcp   open  mysql
```

2401번 포트로 cvspserver가 서비스되고 있음을 알 수 있다.

● pserver 환경 파일 작성

pserver 프로토콜을 사용하기 위해서는 환경 파일을 작성해야 한다. 슈퍼 데몬인 xinetd를 사용하는 경우 /etc/xinetd.d 디렉토리에 cvspserver 파일을 만들고 다음 내용을 입력해 줘야 한다. cvspserver 파일을 생성하고 내용을 편집하기 위해 vi 에디터를 사용하는데 그 명령어는 다음과 같다.

```
# vi /etc/xinetd.d/cvspserver
```

이 명령어를 실행시킨 후 나타나는 vi 에디터에 <리스트 1>의 내용을 입력한다.

<리스트 1> vi 에디터 입력

```
# default: on
# description: The cvspserver serves CVS Password Server
sessions; it uses \
# unencrypted username/password pairs for authentication.
service cvspserver
{
  disable = no
  flags = REUSE
  socket_type = stream
  wait = no
  user = root
  server = /usr/bin/cvs
  server_args = --allow-root=/home/cvs pserver
  log_on_failure += USERID
}
```

<리스트 1>의 내용을 /etc/xinetd.conf에 추가해도 무방하다. 만약 inetd를 사용하는 종전의 리눅스라면 다음과 같이 설정해야 한다.

```
# echo "2401 stream tcp nowait root /usr/bin/cvs cvs -f \ --allow-root=/cvsroot pserver" >> /etc/inetd.conf
```

--allow-root는 cvs 저장소가 /home/cvs임을 명시하기 위해 사용된다. 만약 새로운 namola_world라는 프로젝트를 만들었다면 이 프로젝트는 /home/cvs/namola_world 디렉토리에 저장된다. /usr/bin/cvs는 which cvs라는 명령어를 써서 알아낸 cvs 설치 디렉토리이며 /home/cvs 디렉토리는 cvs 저장소 디렉토리이다. cvspserver 파일의 편집을 완료했다면 다음 명령어를 사용해 xinetd 서비스를 재시작한다.

```
# /etc/init.d/xinetd restart 또는 # service xinetd restart
```

부팅시 자동으로 시작 설정을 하려면 ntsysv 명령어를 이용해 cvspserver에 체크한다. cvs는 cvspserver를 이용한 서비스 외에도 rsh, ssh를 이용한 서비스도 할 수 있다.

● cvs server 전용 사용자 계정 추가

이제 마지막 단계에 돌입했다. cvs 전용의 사용자 계정을 추가하는 것은 cvs 저장소 디렉토리 아래에 있는 CVSROOT 디렉토

리의 passwd 파일을 편집하는 것과 동일한 개념이다. passwd 파일은 기본적으로 생성되는 파일이 아니므로 vi 에디터 등을 통해 직접 생성해 줘야 한다.

```
# vi /home/cvs/CVSROOT/passwd
```

passwd 파일에 저장되는 내용은 cvs에 접근 가능한 계정에 대한 정보로 그 형식과 예는 다음과 같다.

```
CVS account :encrypt password:real account
CVS 계정:암호화된패스워드:리눅스시스템계정
```

```
예) cvsuser01:ldlmR2K0mqw:cvs / cvsuser02:FL9PJOK9zs
    GVY:cvs
```

가운데의 암호화된 패스워드 데이터는 다음과 같이 htpasswd 명령어를 사용해 구할 수 있다.

```
# htpasswd -n cvsuser01 <암호화할 패스워드>
```

이 명령어의 결과는 다음과 같다. 암호화할 패스워드로 사용자 계정 이름과 동일하게 cvsuser01을 입력한 경우이다. cvs용 계정을 더 쉽게 추가하려면 /usr/local/bin 디렉토리에 cvspasswdgen 파일을 생성하고 <리스트 2>와 같은 내용의 스크립트를 기록한다.

```
cvsuser01:ldlmR2K0mqw:cvs
```

<리스트 2> cvspasswdgen 파일의 스크립트

```
#!/usr/bin/perl
#
# Simple script to take a username and password and
# return a line suitable for pasting into the CVS
# password file
#
($u, $p)=@ARGV;
@d=(A..Z,a..z);
$s=$d[rand(52)].$d[rand(52)];
print $u.":".crypt($p, $s).":cvs\n"
```

이어서 cvspasswdgen 파일에 대해 다음과 같이 실행 권한을 부여한다.

```
# chmod +x /usr/local/bin/cvspasswdgen
```

이제 다음과 같은 명령어로 CVS 전용 사용자 계정을 추가할 수 있다. cvspasswdgen 명령어의 첫 번째 파라미터는 사용자 계정 이름이고, 두 번째 파라미터는 해당 계정의 패스워드이다.

```
# cd /home/cvs/CVSROOT
# cvspasswdgen cvsuser02 cvsuser02 >> passwd
```

● 사용자 그룹 추가

마지막으로 cvs라는 사용자 그룹을 만들고 cvs 저장소에 대한 권한을 설정해야 한다. 가장 중요한 건 공동으로 작업할 프로젝트 파일들이 저장될 CVS 저장 디렉토리(저장소)를 설정하는 일이다. 일단 앞의 /etc/xinetd.d/cvspserver를 설정하면서 cvs 데몬이 뜨게 될 경우 --allow-root 옵션을 이용해 /home/cvs를 홈 디렉토리(프로젝트가 저장될 디렉토리)로 지정했다. 이제 /home/cvs를 실제 프로젝트 사용자들이 사용할 수 있도록 권한 설정을 해줘야 한다. cvs의 권한 설정을 위해 cvs란 그룹을 만들게 하고 /home/cvs 디렉토리에 cvs 그룹에 대해 읽기/쓰기/실행 권한을 부여하도록 한다.

```
# groupadd cvs
# chgrp -R cvs /home/cvs
# chmod ug+rwx /home/cvs /home/cvs/CVSROOT
```

cvs 명령어

그럼 지금부터는 cvs 명령어에 대해 살펴본다.

● 저장소에 프로젝트 만들기

저장소가 초기화되고 사용 권한까지 줬으면 이제 CVS를 사용하기 위해 먼저 프로젝트를 초기화시켜야 한다. 그럼 1th_project라는 이름으로 프로젝트를 만들어 보자. 저장소에 새 프로젝트를 만들고 해당 파일들을 CVS에 저장하기 위해서는 import 명령을 사용한다.

```
cvs -d /home/cvs import -m "message" Project_name vender_tag
release_tag
```

-d는 저장소의 위치를 알려 주는 옵션이며 "message"에는 프로젝트에 대한 주석을 넣으면 된다. 'project_name'에는 말 그대로 프로젝트 이름을 넣는다. 여기에 적는 '프로젝트 이름'으로 /home/cvs에 디렉토리가 생성되고 그 안에 소스들이 위치하게 된다. 이어서 vender_tag에는 프로젝트 관리자 ID 정도를 넣어

주고 release_tag는 처음 시작이니 start로 하기로 한다. 마지막의 vender_tag와 release_tag는 별 의미가 없으므로 신경 쓰지 않아도 된다. 그럼 소스가 /root/project에 있다고 가정하고 소스가 있는 디렉토리로 이동한 다음에 해당 소스들을 CVS에 넣어 보기로 한다.

```
#cd project# cvs -d /home/cvs import -m "CVS를 통한 첫 번째 프로젝트" 1th_project kwon_pd start N 1th_project/test.shN
1th_project/INSTALLN 1th_project/READMENo conflicts created by
this import
```

제일 앞의 N은 새로운 파일이 추가되었음을 나타내며, 맨 끝의 No conflicts created by this import 메시지는 저장소에 아무 파일도 없어서 아무런 충돌(conflict)이 일어나지 않았다는 의미다. import 명령으로 프로젝트를 초기화하고 나면 현재 디렉토리에는 CVS와 관련된 정보가 아무것도 없다. 대신 저장소에 모든 내용이 있으므로 project 디렉토리 자체를 지워도 상관없다. 이상으로 프로젝트 초기화는 모두 끝난다.

● 프로젝트 디렉토리에 체크아웃

CVS 서버에서 디렉토리 체크아웃을 할 경우는 다음과 같이 실행한다.

```
cvs -d /home/cvs checkout "프로젝트명" 또는 cvs -d /home/cvs
co "프로젝트명"
```

여기서 /home/cvs는 저장소 경로를 의미한다. cvs checkout(또는 cvs co)은 저장소의 일부를 지역 작업 공간으로 가져온다. 저장소의 모듈 client와 server의 내용을 체크아웃해서 현재 디렉토리의 하위 디렉토리로 저장한다.

```
cvs co client server
```

client/templates와 하위 디렉토리의 파일들만 체크아웃한다.

```
cvs co client/templates
```

CVS는 기본적으로 기본 브랜치(중심 개발축)의 가장 최신 버전을 체크아웃한다. -r이나 -D 옵션을 쓰면 이것을 바꿀 수 있다. -r 옵션은 특정한 개정판을 체크아웃하려고 할 때 사용한다.

```
cvscvscv co -r REL_1_34 client
```

개정판은 추상적인 버전 번호나(1.34처럼) 꼬리표로 지정한다. CVS는 파일마다 각각 개정 기록을 유지하기 때문에, 추상적인 버전 번호를 가지고 전체 프로젝트를 체크아웃하는 것은 의미가 없다. 대신 꼬리표를 흔히 쓰게 된다. CVS는 여러분이 만드는 꼬리표 이외에도 HEAD(저장소에서 가장 최신 버전을 나타냄, 대부분의 명령에서 HEAD는 기본 값으로 이용), BASE(마지막으로 현재 디렉토리에 체크아웃된 개정판을 가리킴) 두 개의 꼬리표를 자동으로 제공한다. -D 옵션은 <리스트 3>과 같이 날짜를 가지고 체크아웃한다.

```

<리스트 3> 명세 예
ISO8601 2007-09-04
20070904
2007-09-04 20:12
2007-09-04T20:12
2007-09-04 20:12Z
2007-09-05 20:12:00-0500

E-Mail format Mon Sun 9 17:12:56 CDT 2007
Sun, Jun 9 17:12:56 2007
Jun 9 17:12:56 2007
June 9, 2007

상대적 날짜 1 day ago
27 minutes ago
last monday
yesterday
third week ago

```

특정 개정판을 지정해서 체크아웃할 경우 '스티키(sticky)' 지정된 디렉토리가 생긴다. 이 말은 이제부터 체크아웃한 디렉토리에서 작업한 모든 내용은 오직 그 개정판에만 적용된다는 의미다. 체크아웃한 개정판이 특정한 브랜치인 경우에만 이 말이 의미를 가진다. 스티키한 꼬리표가 달린 파일을 대상으로 `cvscvscv status` 명령을 내리면 해당 꼬리표를 볼 수 있다.

스티키 꼬리표는 파일의 여러 복사본을 동시에 작업할 수 있도록 하기 위해 CVS가 사용하는 메커니즘이다. 파일에 꼬리표를 다는 방법으로 CVS는 작업하는 파일이 과거 특정 시점에서 온 파일이거나, 저장소에 있는 프로젝트 트리 구조의 특정 브랜치에서 온 파일이라는 사실을 기억한다. 이렇게 함으로써 CVS가 다음 체크인할 때 그 파일의 중심 개발축 브랜치의 HEAD 버전을 덮어쓰는 등의 잘못을 하지 않게 된다. 체크아웃하는 디렉토리 이름을 바꾸고 싶을 경우에는 다음과 같이 하면 된다.

```
cvscvscv co -r REL_1_34 -d rel1.34 client
```

이와 같이 하면 client 모듈의 REL_1_34 꼬리표가 포함하는 파일들을 rel1.34 디렉토리에 체크아웃한다. -q 옵션은 CVS에게 조용히 동작하라고 지시한다. 그 뒤의 -d 옵션은 저장소를 지정하는 것이고, checkout 뒤의 -d 옵션은 체크아웃될 디렉토리를 지정한다.

```
cvscvscv -q -d /usr/repository checkout -d temp
```

● 수정사항 비교

작업 내용을 `commit`으로 반영하기 이전에, 다른 개발자들의 변경 내용을 볼 수 있다. 매우 복잡한 변경을 반영할 때 다른 개발자들의 작업으로 인한 충돌을 막기 위해서다. 여러분이 작성한 파일과 디렉토리를 CVS 서버의 것들과 비교하려면 `cvscvscv diff` 명령을 쓴다. 그러면 어느 코드의 어떤 라인이 수정되었는지를 확인할 수 있다. 이러한 버전별 코드 변경사항을 히스토리로 남긴다면 향후 문제가 생겼을 때 이전 버전의 코드를 쉽게 얻을 수 있다.

- 파일 버전별 차이점 확인

```
#cvscvscv diff -r 1.5 -r 1.6 main.cpp
```

- 저장소에 있는 최신 소스와 현재 작업 디렉토리의 파일을 비교할 때

```
#cvscvscv diff -r HEAD main.cpp
#cvscvscv diff -r BASE main.cpp
```

● 저장소 갱신

받아온 소스를 분석해 본 결과, 버그가 존재한다면 문제가 되는 부분을 수정하고 이 내용을 저장소로 보내려면 `commit`을 수행해야 한다. `commit`하기 전에는 반드시 `update`해야 한다. 내가 아닌 다른 누군가가 또 다시 소스를 수정할 수도 있기 때문이다. `update`를 수행해 충돌(conflict)된 경우에는 바르게 고치고 다시 `commit`을 시도한다. `commit`하는 방법은 다음과 같다.

```
# cvscvscv ci filename
혹은
# cvscvscv commit filename
```

이와 같이 실행하면 갑자기 에디터 화면이 뜬다. 바로 고친 내용을 적으라는 것이다. 적당히 자신이 수정한 내용을 적은 후 저장하고 에디터를 빠져나오면 실제로 `commit` 작업이 수행된다.

메시지를 잘 살펴보면 수정한 소스 파일의 버전이 어떻게 변하는지를 알 수 있다.

cvs는 기본적으로 CVSEEDITOR 환경변수가 있는지 확인해, 있다면 이 변수에 지정된 에디터를 띄우고 없다면 EDITOR 환경변수를 사용한다. EDITOR 환경변수마저 없다면 vi를 사용한다. 에디터를 사용하지 않고 cvs에서 -m 옵션을 써서 change 내용을 적을 수도 있다.

```
# cvs ci -m "main.c modify" filename
또는
# cvs commit -m "main.c modify" filename
```

● 갱신 확인

현재 작업 파일에 대한 상태를 보기 위해서는 <리스트 4>와 같이 status 명령을 사용할 수 있다.

```
<리스트 4> status 명령 사용
$ cvs status README
=====
File: README Status: Up-to-date
Working revision: 1.6 Mon Nov 5 02:14:40 2007
Repository revision: 1.6 /home/CVS/test/README,v
Sticky Tag: (none)
Sticky Date: (none)
Sticky Options: (none)
```

-v 옵션을 줄 경우 파일에 대한 tag를 보여준다.

```
<리스트 5> -v 옵션 추가
# cvs status -v README
=====
File: README Status: Up-to-date
Working revision: 1.6 Mon Nov 5 02:14:40 2007
Repository revision: 1.6 /home/CVS/test/README,v
Sticky Tag: (none)
Sticky Date: (none)
Sticky Options: (none)
Existing Tags:
                                release1-1 (revision: 1.1)
```

● 변경기록

CVS는 작업 내용을 저장소에 저장할 때마다 메시지를 적도록 하고 있다. 이 내용은 저장소에 함께 저장되며 log 명령으로 언제든지 꺼내 볼 수 있다. <리스트 6>은 hello.c를 예로 든 것이다.

매번 commit을 할 때마다 날짜와 시각, 저장한 사람, 메시지 등이 기록되는 것을 알 수 있다. 이렇게 log 명령을 쓰면 누가 언

<리스트 6> 변경기록의 예

```
$ cvs log hello.c
RCS file: /home/CVS/myprj/hello.c,v
Working file: Hello.c
head: 1.5
branch:
locks: strict
access list:
symbolic names:
                                start: 1.1.1.1
                                webdiox: 1.1.1keyword

substitution: kv
total revisions: 6; selected revisions: 6
description:
-----
revision 1.5
date: 2007/11/12 04:04:23; author: namola; state: Exp;
lines: +1 -0
인사말 변경
-----
revision 1.4
date: 2007/11/12 04:03:50; author: namola; state: Exp;
lines: +0 -1
환영합니다 추가
-----
revision 1.3
date: 2007/11/12 04:01:40; author: namola; state: Exp;
lines: +2 -1
주석 추가
-----
revision 1.2
date: 2007/11/12 03:51:36; author: namola; state: Exp;
lines: +1 -0
인사말 추가
-----
revision 1.1
date: 2007/11/12 03:50:48; author: namola; state: Exp;
branches: 1.1.1;
Initial revision
-----
revision 1.1.1.1
date: 2007/11/12 03:50:48; author: namola; state: Exp;
lines: +0 -0
프로젝트 시작
=====
```

제 어떤 작업을 했는지를 한눈에 알 수 있으므로 그간의 작업 내용을 쉽게 파악할 수 있다. 따라서 commit을 할 때 작업 내용을 잘 반영할 수 있는 메시지를 알기 쉽게 적는 것이 중요하다.

● 프로젝트 갱신

받아온 프로젝트를 이리저리 뜯어고친 후 합치려는 순간이나, 체크아웃한 지 오래된 상태에서 누군가가 수정한 부분을 갱신하고자 할 때 다음의 명령을 수행한다.

```
# cvs update myprogram
```

또는 myprogram이라는 프로젝트 안에서 다음과 같이 수행할 수도 있다.

```
# cvs update
```

update와 함께 -P 옵션을 쓰면 add로 추가된 CVS 디렉토리를 받아온다.

```
# cvs update -P
```

● 파일/디렉토리 추가하기

새로운 파일을 로컬 작업 디렉토리에서 생성하고 수정한 후 CVS 버전 관리 저장소에 add하기 위해서는 다음과 같은 명령을 이용한다.

```
cvs add 파일명
```

이어서 'cvs commit 파일명' 명령을 수행해야 한다. 만약 이

파일이 먼저 Add된 것이 아니라면 CVS 시스템은 인식하지 못할 것이다. 만약 여러분이 CVS 쓰기 권한이 있다면, 프로젝트의 소스 트리에 'cvs add' 명령을 써서 하위 디렉토리들을 추가할 수 있다. 기존에 존재하는 파일들을 새로운 하위 디렉토리에 옮기려면 먼저 파일들을 새 위치에 복사하거나 이동시켜야 한다. 아울러 'cvs add' 명령을 사용해서 새로운 위치로 이동 및 복사된 파일들을 CVS 저장소에 추가해야 한다. 기존의 위치에 있는 파일들을 제거하기 위해서는 먼저 기존의 경로에 파일들이 물리적으로 제거되었음을 확인해야 한다. 그런 다음 'cvs remove' 명령과 'cvs commit' 명령으로 CVS 버전 관리 저장소에서 파일들을 삭제하도록 해야 한다. ④

이 글의 기고 형식은 백창우 님의 저서 『유닉스 리눅스 프로그래밍 필수 유틸리티』(한빛미디어)의 내용을 참고했음을 밝힙니다.

참고 자료

1. 『GCC완전정복』, 에이콘
2. <http://sourceware.org/autobook>
3. <http://www.levenez.com/unix>
4. 『Linux Programming 언리쉬드』, 정보문화사
5. 『유닉스 리눅스 프로그래밍 필수 유틸리티』, 한빛미디어

<월간>마소는 늘 개발자의 곁에 서 있습니다

A	Q	B	Y	Z	M	C	U	O	P
D	F	R	C	W	N	H	J	K	V
Z	L	K	S	I	D	Q	N	V	X
C	A	X	M	A	S	O	U	P	T
B	Y	V					G	H	A
E	T	U	O	P	R	W	P	E	S
T	Q	P	U	W	N	C	Z	J	B
I	A	G	S	N	B	L	K	A	C

1983년 11월 창간 역사를 갖고 있는 <월간> 마소, 그 때 태어난 아이가 어느덧 대학생이 되어 마소를 찾는 '예비개발자'가 되어 있습니다. 이미 개발자의 곁에 들어서 있는 것입니다. 프로젝트를 수행하다, 패키지를 사용하다 막힐 때가 많습니다. 우린 이럴 때 어떻게 할까요? 곁에 있는 친구나 동료에게 물어보고 그래도 해결책이 나오지 않으면, IT 단행본을 뒤적여 보기도 합니다. 그래도 해결책이 나오지 않을 때 <월간> 마소가 여러분의 든든한 벗이 될 것입니다. IT 개발자가 힘들어 할 때마다 <월간> 마소는 늘 개발자의 곁을 떠나지 않겠습니다.

IT테크비즈니스 정보지

micro
Software

서울시 강남구 대치동 942-6 5F ● TEL : 02-540-3020 ● FAX : 02-540-3090
www.imaso.co.kr