

```
/*
 * Copyright (c) 2004 Baris Simsek.
 *
 * Permission is granted to copy, distribute and/or modify this document
 * under the terms of the GNU Free Documentation License, Version 1.2
 * or any later version published by the Free Software Foundation;
 * with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
 * Texts. A copy of the license is included in the section entitled "GNU
 * Free Documentation License".
 */
```

Belge Hakkında

- Bu belge 29 Temmuz 2002'de, Açikkod.ORG Yazılım Destekleme Grubu için yazılmıştır.
- 27 Mayıs 2004'te gözden geçirilerek yeniden yayınlanmıştır.
- 11 Ağustos 2006'da genişletilmiştir. Branch oluşturma, sürüm takibi, birlikte çalışma.

Bu belge GNU GFDL Lisansı ile dağıtılmaktadır.

CVS Nedir?

CVS (Concurrent Versions System), versiyon kontrol sistemidir. Bir projeye ait kaynak kodların tarihçesini tutar.

Programcı istediği tarihteki kaynak kodlara erişip üzerinde çalışma yapabilir. Buna neden ihtiyaç duysun? Programcı yaptığı bir değişikliğin bir güvenlik açığına neden olduğunu on gün sonra fark ettiğini varsayalım. On gündür kod üzerinde oldukça fazla değişiklik yapmış olabilir ve bug içermeyen koda geri dönmesi zor olabilir. CVS ile kaynak kodun istediği bir tarihteki haline ulaşabilir.

Kaynak kodun tüm tarihçesinin saklanması gereğinden fazla disk alanı harcayacakmış gibi düşünülebilir. Ancak CVS bunu oldukça zekice bir yöntemle, yalnızca sürümler arasındaki farkları tutarak halleder.

CVS, kaynak kod tarihçesi tutmasının yanı sıra, farklı fiziksel konumlardan kod geliştirmek isteyenler için kaynak kod deposu sunar. Programcı, diğer proje üyelerinin yazdığı kodları görebilir, onlara ilave yapabilir. Yaptığı bu ilaveleri diğer programcılar takip edebilir. GNU programları genelde dünyanın dört bir yanına yayılmış dağınık konumdaki programcılar tarafından geliştirilmektedir. Bu göz önüne alındığında, CVS'in GNU dünyası için ne kadar önemli bir araç olduğu daha açık ortaya çıkar.

CVS, bütün dosyaları merkezi bir depoda (repository) bulunur. Programcılar kod deposuna doğrudan ulaşmazlar. CVS programı aracılığı ile depo üzerinde işlemler yaparlar.

Dosyaların her sürümü uniq bir baskı numarasına (revision number) sahiptir. Baskı numarası '1.1', '1.1.3' şeklindedir.

Kurulum

CVS, <http://www.cvshome.org> sayfasından cvs'i indirilip kurulabilir. FreeBSD kullananlar portlardan, Linux kullananlar ise kendi paket yöneticilerinden (.deb, .rpm gibi) yararlanabilirler. 'cvs' paketinin açıldığı dizinde aşağıdaki komutlar verilerek hızlı kurulum yapılabilir.

```
# ./configure --prefix /usr/local
# make
# make install
```

CVS sunucu (pserver) 2401. portta çalışır. /etc/services dosyasında aşağıdaki satırların olduğundan emin olun:

```
# cat /etc/services | grep cvs
cvspserver 2401/tcp #CVS network server
cvspserver 2401/udp #CVS network server
```

CVS server inetd veya xinetd ile çalışır. **inetd** kullananlar (FreeBSD, Slackware) /etc/inetd.conf dosyasında şu satır yazılmalıdır:

```
# vi /etc/inetd.conf
cvspserver stream tcp nowait root /usr/local/bin/cvs cvs --allow-root=/var/cvsroot pserver
```

Bunu yaptıktan sonra inetd'yi yeniden başlatmalısınız.

```
# kill -HUP inetd
```

Xinetd kullananlar /etc/xinetd.d/cvspserver dosyasına şunları yazmalı:

```
/etc/xinetd.d/cvspserver
service cvspserver
{
  disable = no
  socket_type = stream
  wait = no
  user = cvs
  server = /usr/local/bin/cvs
  server_args = --allow-root=/var/cvsroot pserver
  log_on_success += DURATION USERID
  log_on_failure += USERID
  nice = 10
}
```

tcpserver kullanmak isteyenler için:

```
/usr/local/bin/tcpserver -c 25 0 cvspserver /usr/bin/cvs --allow-root=/var/cvsroot pserver&
```

Burada /var/cvsroot olarak verilen yol *CVSROOT*'u yani kod deposunu gösterir. Kodlar ve CVS yönetim bilgileri bu dizindedir. *CVSROOT* değişkenine bu değeri atamayı unutmayın.

```
# export CVSROOT=/var/cvsroot (bash için)
% set CVSROOT=/var/cvsroot (csh için)
```

/usr/local/bin/cvs ise cvs'in çalışma yoludur. ('which cvs' komutunu vererek öğrenebilirsiniz)

CVS sunucunun cvs kullanıcısı ile çalışacağı varsayılmıştır. Bu kullanıcıyı açmanız gerekir.

```
# groupadd cvs
# useradd -g cvs -d /var/cvsroot -s /bin/false -c "CVS" cvs
```

Bu ayarları bitirtikten sonra CVSROOT dizinini oluşturmak gerekir.

```
# cvs init
```

Bu komutu çalıştırmadan önce CVSROOT değerini export veya set ettiğimize emin olun. Aksi halde şu şekilde kullanmalısınız.

```
# cvs -d /var/cvsroot init
```

Bu komut /var/cvsroot dizinini açacaktır. Ayrıca bu dizin altında yönetim bilgilerini sakladığı CVSROOT dizinini de açar. Bu dizinin haklarını ayarlamayı unutmayın.

```
# chown -R cvs.cvs /var/cvsroot
```

Bundan sonra tüm projeler bu dizin altında bir dizin olarak gözükecektir. Bu dizinlerin kullanıcı haklarını da aynı şekilde ayarlamayı unutmayın.

CVS kullanıcılarını, \$CVSROOT/CVSROOT/passwd dosyasında tanımlayın.

```
# cat /var/cvsroot/CVSROOT/passwd
```

```
simsek:EvQlNsSbLl08I:cvs
user1:WfE9wnFM6gsoQ:cvs
user2:FfJdnVt99Ro/:cvs
```

Parolalar şifrelenmiş şekilde tutulur. Bu şifreleri üretmek için şu scripti kullanabilirsiniz:

```
# vi /usr/local/bin/sifrele
```

```
#!/usr/bin/perl
```

```
 srand (time());
 my $randletter = "(int (rand (26)) + (int (rand (1) + .5) % 2 ? 65 : 97))";
 my $salt = sprintf ("%c%c", eval $randletter, eval $randletter);
 my $plaintext = shift;
 my $crypttext = crypt ($plaintext, $salt);

 print "${crypttext}\n";
```

Bundan sonra "./sifrele sifreniz" komutu ile kodlanmış şifreyi alın ve yukarıda belirtilen biçimde passwd dosyasına yazın. CVSROOT/passwd dosyasındaki satırların en sonundaki cvs, bu kullanıcının map edileceği kullanıcıdır ki bu da CVS sunucunun çalıştığı kullanıcıdır. CVS kullanıcılarımız gerçek olmadığı için gerçek bir kullanıcıya bu şekilde map edilmesi gerekir. Şifre için /etc/shadow'daki kodları da kullanabilirsiniz. Oradan kendi şifrenizi kopyalayıp paste edebilirsiniz.

Proje Başlatmak

cvs sunucuyu istemci olarak kullanmaya başlamadan önce CVSROOT değeri atanmalıdır.

```
# CVSROOT=':pserver:simsek@cvs.enderunix.org:/var/cvsroot '
# export CVSROOT
```

Şimdi bir projeye başlayabiliriz. Sunucu bağlantısını test etmek için aşağıdaki komut verilir:

```
# cvs login
CVS password:
```

Parolanızı girdiğinizde hata vermiyorsa her şey yolunda demektir. Hata veriyorsa /var/cvsroot dizininin kullanıcı haklarının cvs'e ait olup olmadığını, /etc/inetd.conf'da cvs'in yolunun doğru verildiğine emin olun.

Eğer daha önce başladığımız bir projeyi CVS sunucuya yerleştirmek istiyorsak bu projeyi **'import'** etmemiz gerekir. Proje /home/simsek/test altında yer alsın.

```
# cd /home/simsek/test
# cvs import -m "CVS e tasiniyoruz" test enderunix ilksurum
N test/test.c
N test/parser.c
N test/global.h
No conflicts created by this import
```

Eğer CVSROOT değişkeni setlenmemişse -d parametresi ile CVSROOT'un belirtilmesi gerekir. Burada test proje adı ve CVSROOT içindeki dizini, baris dağıtıcı, ikincisurum ise sürüm bilgisidir. Kodlarımız \$CVSROOT altında proje ismi ile aynı dizin içerisine yerleştirilir.

Geliştirici Komutları

Buraya kadar anlatılanlar sistem yöneticisinin yapması gerekenleri içermektedir. Yukarıda sistem yöneticisi test isimli bir proje başlattı. Artık programcılar bu proje üzerinde çalışabilir.

Her geliştirici CVSROOT değişkenini setlemeli. (yapmazsa her defasında -d ile vermek zorundadır.) CVSROOT değişkeninin otomatik setlenmesi için .bashrc veya .cshrc dosyasına yazabilirsiniz.

```
/home/simsek/.bashrc
CVSROOT=:pserver:simsek@acikkod.org:/var/cvsroot
export CVSROOT
```

Artık sisteme login olabilirsiniz.

```
cvs login
CVS password: parolanız
```

CVS sunucuya login olduk. .cvspass isminde bir dosya oluştuğuna dikkat edin. Eğer girişte bir hata oluşmuşsa ekrana hata mesajı verilir. Muhtemel hatalar: 1. Yanlış parola, 2. Erişim hakları, izin hakları

Kullanıcı giriş yaptıktan sonra basitçe aşağıdaki işlemleri yapabilir.

Kodları Sunucudan Çekmek

```
$ cvs checkout test
cvs checkout: Updating test
U test/global.h
U test/parser.c
U test/test.c
```

Bununla test modülüne (CVS terminolojisinde projeler modül olarak isimlendirilir) ait kodları CVS deposundan kendi yerel bilgisayarınıza indirmiş olursunuz. Kodlarla birlikte CVS isimli bir dizin de indirilir. Bu, her bir kaynak kod için ilave bilgiler tutan CVS'e özel bir dizindir. Geliştiricinin bu dizinle herhangi bir işi olmayacaktır.

Kodları Sunucuya Yükleme

```
cv$ commit
```

Bu komut dizindeki tüm dosyaları cvs'e göndermenizi sağlar. Göndermeden önce \$CVSEEDITOR veya o yoksa \$EDITOR çevre değişkenlerinde tanımla dosya düzenleyiciyi açarak açıklama girmenizi ister. Eğer dosya düzenleyiciyi açmak istemiyorsanız -m parametresi ile komut satırından açıklamayı girin.

```
cv$ -m "memory leak sorunu cozuldu" commit
```

Eğer yalnızca bir dosyayı sunucuya göndermek istiyorsanız dosya adını da vermeniz gerekmektedir.

```
cv$ commit parse.c
```

Kodların Senkronizasyonunu Yapmak

Diyeliki test.c yi çektiniz. Üzerinde bazı değişiklikler yaptınız. Diğer takım arkadaşımız da aynı dosya üzerinde değişiklikler yaptı ve üstelik sizden önce sunucuya gönderdi. Siz eğer yeni sürümü sunucuya göndermeye çalışırsanız CVS, Orijinal kopyanın değiştiği konusunda sizi uyaracak ve işleminizi yapmayacaktır.

```
$ cv$ commit
```

```
cv$ commit: Up-to-date check failed for `test.c'  
cv$ [commit aborted]: correct above errors first!
```

Bu nedenle kendi çalışma dizininiz ile depodaki sürümü senkron etmeniz gerekmektedir.

```
$ cv$ update
```

```
cv$ update: Updating .  
U global.h  
RCS file: /cv$up/cv$/test/test.c,v  
retrieving revision 1.2  
retrieving revision 1.3  
Merging differences between 1.2 and 1.3 into test.c  
rc$merge: warning: conflicts during merge  
cv$ update: conflicts found in test.c  
C test.c
```

Yukarıdaki çıktıyı inceleyelim:

```
U global.h
```

Bu, global.h dosyasının 'Update' olduğunu bildiriyor. Bizim dizindeki global.h'da biz değişiklik yapmamışızama takım arkadaşımız global.h 'i değiştirip sunucuya göndermiş. Bu durum 'Update' durumudur ve bizim çalıştığımız dizindeki sürüm, güncel sürüm ile değiştirilir.

```
C test.c
```

Conflict yani çakışma olmuştur. Bu durumda CVS iki sürümü merge edip ne yapması gerektiği konusunu geliştiriciye bırakmıştır. Bu kararı kendisinin vermesi mümkün değil.

```

if (limit == MAXUSERS)
    return 1;
<<<<<<< test.c

if (limit < MAXUSERS)
    get_user_profile();

=====

if (limit < MINUSERS)
    return 2;
>>>>>>> 1.3

```

Burada <<<<<<< ile >>>>>>> arasında kalan kısımda iki versiyonun kodları ===== ayırıcı ile ayrılmış şekilde geliştiricinin dikkatine sunulmuştur.

Eğer M test.c gibi Bir şey olsaydı bu 'Modified' demek olacaktı. Yani siz orijinal sürüm üzerinde değişiklik yapmışsınız ancak diğer geliştiricilerin bundan henüz haberi yok. İlk commit işleminden sonra bu uyarı kalkacaktır. Çünkü CVS deposu artık bu değişiklikten haberdar olmuş olacaktır.

\$ cvs update

```

cvs update: Updating .
M lex.c
$ cvs commit lex.c
Checking in lex.c;
/home/simsek/parser/src/lex.c,v <-- lex.c
new revision: 1.5; previous revision: 1.4

```

Yeni Dosya Ekleme

```

$ vi yeni.c
$ cvs add yeni.c
$ cvs commit

```

Mevcut Dosyayı Silme

```

$ rm silinecek.c
$ cvs delete silinecek.c
$ cvs commit

```

Çalışma Dizinini Silme

Program yazma işlemini bitirdikten sonra çalışma dizininizi silmeye kara vermişseniz basitçe:

```

$ cd ..
$ rm -rf test

```

Ancak bunun yerine release kullanmanız daha sağlıklıdır. Zira release yaptığınız tüm değişiklikleri kontrol edip sunucuya gönderilmemiş değişiklik varsa sizi uyarır. Böylece çalışmanızı kaybediğinize emin olursunuz.

```
cd ..  
cvs release -d test
```

CVS'e Bildirilmemiş Kodlar

update, import ve release komutları verildiğinde CVS, dizinde olan ancak CVS'e eklenmemiş dosyalar için uyarı yapar.

```
$ cvs update
```

```
? test
```

```
? parser.o
```

```
cvs update: Updating .
```

'test' derleme sonucu oluşmuş bir binary, 'parser.o' ise object kod. Bu tür dosyalar çalışma esnasında oluşur ve bunları siz CVS'e eklemesiniz. Ancak sürekli bu şekilde CVS sizi uyacaktır. Bundan kurtulmak için:

1. `CVSROOT/CVSROOT/cvsignore` (yani `/var/cvsroot/CVSROOT/cvsignore`) dosyasına CVS'in dikkate almamasını istediğiniz dosya isimlerini veya patternleri yazabilirsiniz. Bu liste bütün kullanıcılar için geçerli olacaktır.
2. Her kullanıcı kendi ev dizininde `.cvsignore` dosyasında bu dosyaları veya patternleri belirtebilir. Bu yalnızca kendisi için geçerli olacaktır.
3. `CVSIGNORE` çevre değişkeninde dikkate alınmayacak dosya isimleri veya patternler belirtilebilir.

Artık 'cvs update' yaptığınızda uyarı almayacaksınız.

Sürümler Arası Farkları İzlemek

test.c dosyasında değişiklik yapıp yapmadığınıza emin değilsiniz. Kod deposundaki sürümle elinizdeki (çalıştığınız dizindeki) sürüm arasında fark var mı kontrol etmeniz gerekecektir.

```
$ cvs diff test.c
```

```
Index: test.c
```

```
=====
```

```
RCS file: /cvsup/cvs/test/test.c,v
```

```
retrieving revision 1.1.1.1
```

```
diff -r1.1.1.1 test.c
```

```
0a1,2
```

```
> if (limit == MAXUSERS)
```

```
>     return 1;
```

```
$ cvs commit
```

```
Checking in test.c;  
/cvsup/cvs/test/test.c,v <-- test.c  
new revision: 1.2; previous revision: 1.1  
done
```

```
$ cvs diff test.c
```

```
$
```

Sürüm Günlüğünü Görüntüleme

Diğer geliştiricilerin neler yaptıklarını merak ediyorsanız 'cvs log' ile sürümlere ait günlük kayıtlarına ulaşabilirsiniz.

```
devel~/test$ cvs log test.c
```

```
RCS file: /cvsup/cvs/test/test.c,v
```

```
Working file: test.c
```

```
head: 1.5
```

```
branch:
```

```
locks: strict
```

```
access list:
```

```
symbolic names:
```

```
    eu: 1.1.1.1
```

```
    errr: 1.1.1
```

```
keyword substitution: kv
```

```
total revisions: 6;      selected revisions: 6
```

```
description:
```

```
-----  
revision 1.5
```

```
date: 2006/08/11 22:45:38; author: simsek; state: Exp; lines: +3 -0
```

```
limit kontrolu eklendi  
-----
```

```
revision 1.4
```

```
date: 2006/08/11 22:44:44; author: afsin; state: Exp; lines: +0 -2
```

```
memory leak giderildi  
  
.  
.  
.  
=====
```

Dallanma

Dallanma (branch) farklı bir geliştirme yolu açmak veya paralel geliştirme için yapılır. Bu durumda kod için paralel bir geliştirme hikayesi oluşacaktır. Dallanmaya neden ihtiyaç duyulur? Düşünün ki geliştirici yeni bir sürüm üzerinde çalışırken çok eski sürümdeki bir hata kendisine raporlandı. Mevcut sürümde bu düzeltildi, ancak eski sürümde de düzeltilip müşteriye bu patch'in gönderilmesi gerekmektedir. İşte böyle durumlar için yeni sürümün kod ağacını etkilemeyecek ancak eski sürümü ana kod ağacından ayırıp ayrıca takip edebilecek bir mekanizma olarak dallanma yapılır.

Bir kaynak kodun üzerinde çalışılan ana koda 'trunk' denir. 'cvs checkout' veya 'cvs commit' komutları ana trunk'dan kodları alır veya ana trunk'a kodları gönderir.

Ana trunk üzerindeki qsheff modülünden qsheff_II isimli yeni bir dal oluşturmak için:

```
$ cvs rtag -b qsheff_II qsheff
```

Ana trunk üzerinde etiketlenmiş bir modülden dal oluşturmak için:

```
$ cvs rtag -r qsheff_I_last -b qsheff_II qsheff
```

'tag' komutunun aynı zamanda bir sürümü etiketlemek için kullanılması sizi şaşırtmasın. Kullanımları farklıdır.

!!! Etiketlemek yapmak için '-r etiket_ismi' şeklinde ayrıca parametre ilave edilmesi de gerekir.

'rtag' kod deposu üzerindeki son sürüm üzerinde işlem yapar. Eğer çalıştığınız kopya üzerinde dallanma yapacaksanız 'rtag' yerine **'tag'** kullanmalısınız.

!!! 'tag' ve 'rtag' depo üzerinde 'commit' gerektirmeksizin hemen etki yapar.

Dallanmadan sonra sizin üzerinde çalıştığınız kod eski daldan devam eder. Yeni oluşturulan dala geçmek için:

```
$ cvs update -r branch_ismi
```

Bundan sonra yapacağınız commit işlemleri yeni dal üzerinde baskı (revision) olarak kaydedilecektirler.

Bir kod dalındaki son baskıyı çekmek için:

```
$ cvs checkout -r branch_adi modul_adi
```

Etiketleme

Etiket, kodun bir anlık resmini alıp (snapshot) isimlendirmedir. Bundan sonra kodun bu haldeki durumunu çağırarak için bu etiket kullanılabilir.

```
$ cvs tag tag_adi
```

```
$ cvs tag SNAPSHOT_06_12
```

Daha sonra kodun etiketlenmiş bu anını çağırmak için:

```
$ cvs checkout -r tag_adi modul_adi
```

```
$ cvs checkout -r SNAPSHOT_06_12 qsheff
```

!!! Etiketler kodun belirlenmiş sabit bir anını gösterirken branş isimleri o branşa ait son sürümü gösteren değişken bir referanstır. Bu nedenle etiketin gösterdiği kodda değişiklik yapılamazken branş üzerinde değişiklik yapılabilir.

SONSÖZ

CVS'in pek çok özelliği mevcuttur. Bir sonraki güncellemede bunlar eklenecek. Şimdilik bu kadar.

Barış Şimşek
EnderUNIX Yazılım Geliştirme Takımı
<http://www.enderunix.org/simsek/>