42 Shell00 Çözümleri

ex00:

1) touch z ile dosya oluştur.
2) vim z ile aç.
3) i tuşuna basarak yazma moduna geç.
4) Z yaz.
5) ESC, : wq! ile kaydedip çık.

ex01:

1) -rw-rw-r-- iznini -r--r-xr-x'e dönüştür.
2) 3'er 3'er bölelim: r-- r-x r-x.
3) Dosya karakterleri:
   - -: Normal dosya
   - d: Dizin
   - l: Sembolik bağlantı
4) İzinler: r=4, w=2, x=1.
5) İzinleri sayısal değerlere çevir:
   - r--: 4
   - r-x: 5
   - r-x: 5
6) chmod 455 komutunu kullanarak izni değiştir.
7) testShell00 dosyasının boyutu 40 byte.
8) Tarih değişikliği: touch -t 06012342 komutuyla.
9) tar -cf testShell00.tar testShell00 ile dosyayı arşivle.

ex02:

1) mkdir test0 test2 ile klasörler oluştur.

2) touch test1 test3 test4 ile dosyalar oluştur.

3) ln -f test3 test5 **ile link bağlantısı** oluştur.

4) ln -s test0 test6 **ile sembolik bağlantı** oluştur.

5) Dosya izinlerini değiştir.

6) Byte değerlerini ayarla **(1 byte olan dosyalara bir karakter yazıp silerek boyutu ayarla)**

7) touch -t komutuyla tarihi değiştir. touch -h -t ile sembolik bağlantının tarihini değiştir.

** Önemli : stat komutu ile tarih değişliğine bakabilirsin...

Link bağlantısı çok kısaca şu aslında mesela bir oyun yükledik masaüstünde duran oyun kısayolu bir link bağlantısı aslında oraya tıklayınca oyunun kurulu olduğu dizine çıkıp oradaki ana çalıştırma dosyasına erişiyor buna link bağlantısı diyoruz.

Sembolik bağlantı, bir dosya veya klasörün aslında bulunduğu yerin 'adresini' gösteren bir yol gibidir. Link bağlantısı direkt oradaki dosyaya erişirken sembolik bağlantı orayı gösterir bir nevi.

ex03

SSH, bilgisayarlar arasında güvenli veri iletimi ve uzak sistemlere bağlanmak için kullanılan şifreli bir protokoldür. Özellikle uzak sunucu yönetimi için yaygındır.

Diyelim ki senin evine gitmek istiyorsun, fakat kapıyı sadece sen açabilirsin.

Public Key (Genel Anahtar)

- Bu, senin evinin kapı numaran gibi düşünülebilir.
- Herkese açıklanabilir; yani, kapının numarası herkes tarafından bilinir.
- Bu numara, herkesin kapıya ulaşmasını sağlar, fakat kapıyı açmak için sadece senin anahtarına ihtiyaç vardır.

Private Key (Özel Anahtar)

- Private key, senin anahtarın gibi. Bu anahtar, sadece sende bulunur ve evinin kapısını açmak için gereklidir.
- Eğer bu anahtarı kaybedersen ya da başkalarına verirsen, başkaları evine girebilir ve güvenliğin tehlikeye girer.
- Bu anahtar, sadece senin ve evinin kapısının uyumlu olduğu bir anahtardır.

Şimdi, SSH Key ile bağlanma örneğine dönersek:

- Public key'i evinin kapısına asıyorsun (sunucuya yükleniyor).
- Senin private key'inle kapıyı açabiliyorsun, çünkü sadece senin özel anahtarınla evin kapısı açılır (bağlantı sağlanır).
- Başkası bu kapı numarasını (public key) bilse bile, private key olmadan kapıyı açamaz.

Kısaca, public key kapıyı bulmak için gerekli olan numara, private key ise kapıyı açan anahtar gibi düşünülebilir. Bu ikisi birbirini tamamlar ve güvenli bir bağlantı sağlar.

SSH key nasıl yaratılır ?

ssh-keygen komutuyla SSH anahtar çifti oluştururuz.

cd ~/.ssh **ile anahtarların bulunduğu dizine gideriz.**

id_rsa.pub public key, id_rsa ise private key'tir.

Bizden istenen ise id_rsa_pub dosyası oluşturup içerisine public key’imizi atmamız isteniyor.

ex04:

1)Burada bizden klasör içindeki dosyaları sıralamamız isteniyor bunu net görmemiz için ayrı zamanlarda 3 klasör 3 dosya yaratıyoruz.

2) Terminale ls -tmp yazıyoruz

-t:
Dosya ve dizinleri düzenleme tarihine göre sıralar.
En son düzenlenen dosyalar en üstte olur.

-m:

Listeleme biçimini virgül ve boşluklarla ayırarak yatay bir satırda gösterir.
Bu, uzun listeler yerine daha kompakt bir görünüm sağlar.

-p:

Dizinlerin sonuna bir / ekler.
Bu, dizinleri dosyalardan kolayca ayırt etmeye yardımcı olur.

3) test5/, test4/, test3/, test2, test1, test0,

- t en son açılanı birinci sıraya aldı
- m virgül attı
- p klasörlerin sonuna dizin işareti attı

ex05:

Git Commit: Git'te commit, ilerlemeyi kaydetmek ve projeye geri dönmek için kullanılır. Commit'ler, projenin geçmişine dair kayıtlar oluşturur.

Sondan 5 commit'in hash kimliklerini görmek için;

git log --format='%H' -n5 komutuyla;

git log: Commit kayıtlarını listeler.

- --format='%H': Sadece commit hash'lerini gösterir.
- -n5: En güncel 5 commit'i gösterir.

ex06:

.gitignore dosyası, gereksiz dosyaların git tarafından izlenmesini engellemek için kullanılır. GitHub'a yüklerken, bu dosya istenmeyen dosyaları gizler.

Adımlar;

1) .DS_Store ve mywork.C dosyalarını oluşturun.
2) .gitignore dosyasını oluşturun.
3) .gitignore dosyasına bu iki dosyayı yazın.
4) git_ignore.sh dosyasını oluşturun.
5) İçine şu komutu yazın;

git ls-files --others --ignored –exclude-standard

git ls-files: Depodaki dosyaları listeler.

- --others: Git tarafından **henüz takip edilmeyen** dosyaları gösterir.
- --ignored: **.gitignore** dosyasında belirtilen ve **göz ardı edilen** dosyaları gösterir.
- --exclude-standard: Git'in **varsayılan göz ardı kurallarına** uyan dosyaları hariç tutar.

ex07:

diff komutu, iki dosya arasındaki farkları görmek ve kıyaslama yapmak için kullanılır.

Adımlar:

1) a ve b adında iki dosya oluşturun.
2) a dosyasına bir metin ekleyin.
3) b dosyasındaki metni küçük değişikliklerle düzenleyin.
4) diff a b > sw.diff komutuyla iki dosya arasındaki farkları sw.diff dosyasına yazdırın. (Git deposu başlatmanız gerekir)
5) patch b < sw.diff komutuyla, sw.diff dosyasındaki farkları b dosyasına uygulayın.

ex08:

Verilen komutlarla, belirli dosya adlarına sahip dosyaları bulup sileriz:

1) touch clean: clean adında bir dosya oluşturur.
2) **find . -type f \(-name '#*#' -o -name '*~' \) -print -delete**: Geçerli dizin ve alt dizinlerde, adı #*# veya *~ ile biten dosyaları bulur ve siler.
3) touch '#example#' 'file~': Bu tür dosyaları oluşturur.
4) bash clean: clean dosyasını çalıştırarak işlemi başlatır.

Açıklamalar;

1) find . : Geçerli dizinde ve alt dizinlerde arama yapar.
2) -type f: Sadece dosyaları arar.
3) \(-name '#*#' -o -name '*~' \) Adı #*# veya *~ ile biten dosyaları bulur.
4) -print: Bulunan dosyaların adlarını yazdırır.
5) -delete: Bulunan dosyaları siler.

Notlar;

1) * joker karakteriyle herhangi bir dosya adı temsil edilir.
2) -o "veya" anlamına gelir.
3) Dosya isimlerini tırnak içine almak, özel karakterlerin kabuk tarafından yanlış yorumlanmasını engeller.

ex09 :

touch file_A file_B ile dosyalar oluşturulur.

1) file_A içerisine text yazılır.
2) file_B içerisine 42. satırına kadar boşluk bırakıp 42 yazılır.
3) ft_magic dosyası yaratılır.
4) İçerisine 41 string 42 42_karakteri_bulundu yazılır.
5) file -m ft_magic file_A file_B komutu çalıştırılır.

Magic, syntaxı şudur;

<bayt_sırası> <veri_türü> <değer> <etiket>

- 41 - string - 42 - 42_karakteri_bulundu