



Anas Alshourafa

Öğr. No/20360859117

Bilgisayar Mühendisliği Bölümü
3.Sınıf

Sunum tarihi: 25.04.2024

Bash script



İçindekiler

1. Bash script Nedir ?	3
2. Bash script Avantajları	4
3. Bash kabuğuna ve komut satır arayüzüne genel bakış	8
4. Bash komutları	9
5. Bash betikleri nasıl oluşturulur ve çalıştırılır?	10



1. Bash script Nedir?

(Bourne Again Shell)

- Bash script, bash programı tarafından satır satır yürütülen komut dizisini içeren bir dosyadır. Belirli bir dizine gitmek, klasör oluşturmak ve komut satırını kullanarak bir süreç başlatmak gibi bir dizi eylemi gerçekleştirmenize olanak tanır.

2. Bash script Avantajları:

- Bash script, Unix/Linux sistemlerinde sistem yönetimi görevlerini otomatikleştirmek, sistem kaynaklarını yönetmek ve diğer rutin görevleri gerçekleştirmek için kullanılan güçlü ve çok yönlü bir araçtır. Kabuk komut dosyası oluşturmanın bazı avantajları şunlardır:
- **Otomasyon** : tekrarlanan görevleri ve süreçleri otomatikleştirmenize, zaman kazanmak ve manuel yürütme sırasında oluşabilecek hata riskini azaltmanıza olanak tanır.

- **Taşınabilirlik:** Kabuk komut dosyaları, emülatörler veya sanal makineler kullanılarak Unix, Linux, macOS ve hatta Windows dahil olmak üzere çeşitli platformlarda ve işletim sistemlerinde çalıştırılabilir.
- **Esneklik:** Kabuk komut dosyaları son derece özelleştirilebilir ve belirli gereksinimlere uyacak şekilde kolayca değiştirilebilir. Daha güçlü komut dosyaları oluşturmak için diğer programlama dilleri veya yardımcı programlarla da birleştirilebilirler.

- **Erişilebilirlik:** Kabuk komut dosyalarının yazılması kolaydır ve herhangi bir özel araç veya yazılım gerektirmez. Herhangi bir metin düzenleyici kullanılarak düzenlenebilirler ve çoğu işletim sisteminde yerleşik bir kabuk yorumlayıcısı bulunur.
- **Entegrasyon:** Kabuk komut dosyaları, veritabanları, web sunucuları ve bulut hizmetleri gibi diğer araç ve uygulamalarla entegre edilerek daha karmaşık otomasyon ve sistem yönetimi görevlerine olanak tanır.

- **Hata ayıklama:** Kabuk komut dosyalarının hatalarını ayıklamak kolaydır ve çoğu kabukta, sorunların hızlı bir şekilde tanımlanmasına ve düzeltilmesine yardımcı olabilecek yerleşik hata ayıklama ve hata raporlama araçları bulunur.

3. Bash kabuğuna ve komut satır arayüzüne genel bakış

- Kabuk türünü belirlemek için **ps** komutunu kullanır:

```
anaspc@LAPTOP-650GEBN8:~$ ps
  PID TTY          TIME CMD
  1920 pts/0        00:00:00 bash
  1942 pts/0        00:00:00 ps
```


4. Bazı bash komutları :

- Genel olarak komutların sözdizimi:

```
command [OPTIONS] arguments
```

birkaç temel bash komutu :)

date : Geçerli tarihi görüntüler.

pwd (Print Working Directory): Mevcut çalışma dizinini görüntüler.

ls: Geçerli dizinin içeriğini listeler.

echo: Bir metin dizesini veya bir değişkenin değerini terminale yazdırır.

man : bash komutlari için mevcut olan kılavuz sayfaları

5.Bash scripts nasıl oluşturulur ve çalıştırır ?

- Komut dosyası adlandırma kuralları:

Bash betikleri .sh ile biter. Ancak bash betikleri sh uzantısı olmadan da çalışabilir. Örnek : ornekFonksiyon.sh

Shebang'ı eklemek: `#!/bin/bash` (heştek + ünlem + bash yolu)

Shebang, bash # ve bang ! birleşimidir. ardından bash kabuk yolu gelir. Bu betiğin ilk satırıdır. Shebang, kabuğa onu bash kabuğu aracılığıyla çalıştırmasına söyler. Shebang, bash yorumlayıcısına yoldur.

- Aşağıdaki komutu kullanarak bash kabuk yolunuzu (yukarıdakilerden farklı olabilir) bulabilirsiniz:

```
which bash
```

İlk bash betiğimizi oluşturma:

.sh dosyası oluşturmak var. Bunun için (vi , vim , nano) Editor'ü kullanabilir. Bir dosya oluşturmak için bu komutu kullanır:

```
anaspc@LAPTOP-650GEBN8:~$ nano firstScript.sh
```

Örnek:

Satır #1: Shebang (#!/bin/bash) bash kabuk yolunu işaret ediyor.

Satır #2: Echo komutu terminalde geçerli tarih ve saati gösteriyor. Tarihin geri tırnak içindedir.

Satır #4: Kullanıcının geçerli bir yol girmesini istiyoruz.

Satır #5: Okuma komutu girişi okur ve onu the_path değişkeninde saklar.

Satır #8: ls komutu, değişkeni saklanan yolla alır ve geçerli dosya ve klasörleri görüntüler.

```
GNU nano 6.2 firstScript.sh
#!/usr/bin/bash
echo "Today is " `date`

echo -e "\nenter the path to directory"
read the_path

echo -e "\n you path has the following files and folders: "
ls $the_path
```

*Bash script Çalıştırmak:

- Betiği çalıştırılabilir hale getirmek için bu komutu kullanarak kullanıcınıza çalışma haklarını atayın:

```
anaspc@LAPTOP-650GEBN8:~$ chmod u+x firstScript.sh
```

- Bash script çalıştırması yöntemleri:
 - * sh firstScript.sh
 - ** bash firstScript.sh
 - *** ./firstScript.sh



5.Kaynaklar

- FreeCodeCamp:
- <https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/#introduction>
- HOSTINGER REHBERLER:
- <https://www.hostinger.web.tr/rehberler/bash-script-nedir-ve-ornekleri>



Sorularınız!



TEŞEKÜRLER