

# Linux/Unix Sistem Programlama

2023-2024 Bahar Dönemi

7-Hafta

Dr. Öğr. Üyesi Nurullah ÖZTÜRK

# Kabul (Shell) Programlama Nedir?

---

- UNIX veya Linux gibi işletim sistemlerinde kabuk üzerinden otomatik görevleri gerçekleştirmek için kullanılan bir dizi komutun yazılması ve bunların bir dosyada saklanarak çalıştırılması işlemidir.
- Kabuk, kullanıcı ile işletim sistemi arasında bir arayüz görevi görür ve kullanıcının komutlarla işletim sistemine talimatlar vermesine olanak tanır.
- Kabuk programlama, bu komutları bir dosyada bir araya getirerek, belirli işlemleri otomatikleştirmeyi ve basitleştirmeyi amaçlar.

# Kabuk Programlamanın Temel Özellikleri

---

- **Otomatizasyon:** Sık yapılan işlemleri otomatikleştirmek için kabuk betikleri (scriptleri) kullanılır. Bu, dosya işlemleri, sistem bakımı, yedekleme gibi rutin görevlerin hızlı ve hatasız şekilde yapılmasını sağlar.
- **Zaman Tasarrufu:** Bir dizi işlemi manuel olarak yapmak yerine, bir kabuk betiği bu işlemleri tek bir adımda yapabilir.

# Kabuk Programlamanın Temel Özellikleri

---

- **Taşınabilirlik:** Kabuk betikleri, farklı UNIX/Linux sistemlerinde çalışacak şekilde tasarlanabilir, bu da yazılımın farklı ortamlara kolayca taşınmasını sağlar.
- **Esneklik:** Kabuk betikleri, basit metin dosyaları olarak saklandığından, herhangi bir metin editörü ile kolayca oluşturulabilir, değiştirilebilir ve yönetilebilir.

# Kabuk Programlama Dilleri

---

- **Bash (Bourne-Again SHell):** En yaygın kullanılan kabuk programlama dili. Linux sistemlerin çoğunda varsayılan kabuk olarak gelir.
- **SH (Bourne Shell):** Kabuk programlamanın atası olarak kabul edilir. Diğer kabukların çoğu bu kabuğun sunduğu temel özellikler üzerine inşa edilmiştir.
- **Zsh (Z Shell):** Bash'e alternatif olarak geliştirilmiş, birçok gelişmiş özellik sunan bir kabuk.
- **Csh (C Shell) ve Tcsh (TENEX C Shell):** C programlama diline benzer sözdizimi ile dikkat çeker.

# Kabuk Programlama Dilleri

---

- Kabuk betikleri genellikle .sh uzantısı ile kaydedilir ve çalıştırılabilir hale getirilmek için uygun izinlerin verilmesi gerekir.
- Kabuk programlama, sistem yöneticileri, yazılım geliştiricileri tarafından yaygın olarak kullanılır.

# Örnek

---

- Nano ile bir dosya açılır ve uzantısı .sh olarak kaydedilir.
- Shell programlamada ilk başta hangi Shell programı ile çalıştırılacak ise bu bilgi en başta yer alır.
  - `#!/bin/sh`
- Bu satırın altında verilen tüm komutlar Shell tarafından gerçekleştirilir.
- Oluşturulan .sh dosyasını çalıştırmak için
  - `./dosyaadi.sh`

# Örnek-devamı

---

- Erişim engelleme hatası verir. Çünkü çalıştırılabilir yetkisine sahip değildir.
  - `ls -al` komutu ile kontrol edebilirsiniz.
- Çalıştırılabilir yetki vermek için
  - `Chmod +x dosyaadi`



# bash & sh arasında fark

---

- bash (Bourne-Again SHell) ve sh (Bourne Shell) arasındaki farklar, bu kabukların sunduğu özellikler ve kullanım kolaylıkları açısından önem taşır. sh, UNIX işletim sistemlerindeki orijinal kabuk programlama dilidir
- bash, ise sh'nin bir genişletmesi olarak GNU Projesi kapsamında Brian Fox tarafından 1980'lerin sonunda yaratılmıştır. bash, sh'nin tüm özelliklerini içerir ve birçok yeni özellik ekler, dolayısıyla daha gelişmiş ve kullanıcı dostu bir kabuktur

- 
- **Özellikler:** bash, sh'ye kıyasla birçok ilave özelliğe sahiptir. Örneğin; geliştirilmiş tab tamamlama, komut geçmişi, dizi işleme, aritmetik işlemler, ileri düzey desen eşleştirme ve işlem kontrolü gibi özellikler bash'te mevcuttur ama sh'de yoktur veya sınırlıdır.
  - **Uyumluluk:** sh daha çok uyumluluk ve taşınabilirlik için tercih edilirken, bash modern ve gelişmiş özellikleri nedeniyle günlük kullanım ve gelişmiş betik yazımı için tercih edilir.
  - **Sözdizimi:** Her ne kadar bash, sh'nin sözdizimini desteklese de, bash özel sözdizimleri ve genişletilmiş işlevsellikler içerir. sh betikleri genellikle bash üzerinde sorunsuz çalışır, ancak bash özelliklerini kullanan betikler sh üzerinde çalıştırılamayabilir.