

GRUP NO: 37

Metin Hakan Yılmaz / B221210075

Elşat Mirdesoğlu / B221210579

Osman Yıldız / B221210021

Abdulsamed Kurubal / B221210009

Ferdi Kaynar / G201210311

Github linki: <https://github.com/heikan1/37>

## KABUĞUN FONKSİYONLARI

Geliştirdiğimiz kabuk uygulamasında bizlerden istenilen boru (“|”) özelliği hariç tüm özellikleri, dokümanlarda kullanmamız istenilen linux sistem çağrılarını kullanarak ubuntu distrosunda gerçekleştirdik. Programımız “quit” komudunu girdi alana kadar veyahut status değişkeni 1’den farklı bir değer alana kadar sonsuza kadar prompt kabul eden bir döngü içerisinde çalışmaktadır. Önce kullanıcıdan bir satır girdisi alınır, bu satır noktalı virgül (“;”) karakterine göre ayrılarak komutlara bölünür ve bir dizi içerisinde tutulurlar. Sonrasında her bir komut boşluk karakteri dikkate alınarak argümanlarına bölünür ve ilgili system çağrıları kullanılarak execute edilir. Execute aşamasında komutlar arkaplanda çalışanlar ve çalışmayanlar olarak ikiye ayrılır (& işaretinin varlığına göre), eğer ki arkaplanda çalışan bir komut ise ebeveyn proste çocuk prosesin çalışmasının bitmesi beklenilmez, kabuk döngüsü olduğu gibi çalışmaya devam eder, yeni komutları kabul eder. Çocuk prosesin bittiği ise gönderdiği sinyal ile anlaşılır.

Kabuk uygulamamızda gömülü (built-in) komut olarak, “quit” “help” ve “cd” komutları bulunur. Uygulama her bir komutun, gömülü komut mu yoksa linux varsayılan sistem komutu mu olduğunu kontrol ediyor, eğer girilen komut gömülü bir komut ise ilişkili fonksiyonu çalıştırıyor, değilse varsayılan komut execute ediliyor. Girdi-Çıktı yönlendirmesi sırasında eğer ki girdi dosyası bulunmuyorsa, “böyle bir dosya bulunmuyor.” hatası kabuğa yazdırılır, eğer ki çıktı dosyası bulunmuyor ise o dosya oluşturulur ve istenilen yönlendirme gerçekleştirilir.

## GELİŞTİRME SÜRECİ

Kabuk uygulaması geliştirilirken öncelikle bir kabuk uygulamasının tam olarak ne olduğunu anlamak üzerine örnek uygulamalar incelendi ve dokümanda bizlere sunulan sistem çağrıları hakkında araştırmalar yapıldı. Nasıl çalıştıkları ve nasıl argümanlar bekledikleri incelendikten sonra sistemin her bir fonksiyonu bu çerçevede tersten tasarlanarak inşa edildi.

## GÖREV PAYLAŞIMI VE GRUP HALİNDE ÇALIŞMA

Gruptaki herkesin program geliştirmek adına temel bilgilere sahip olduğundan emin olunduktan sonra, programın temel akışı ve hangi fonksiyonlardan ibaret olacağı, bu fonksiyonların girdilerinin ve çıktılarının neler olacakları tüm grup üyeleri ile beraber kararlaştırıldı. Sonrasında Her bir fonksiyon, herkese olabildiğince eşit yük düşecek biçimde paylaştırıldı. Herkes kendi kısmını tamamladıktan sonra ise uygulama bir araya getirildi.

## TEST SÜRECİ

Uygulamayı, istenilen kriterleri karşıladığından emin olmak için, dokümanda verilen örnek komutları kullanarak test ettik.