



Öğrencinin;

ADI: **Burak Enes**
SOYADI: **Demir**
NO: **1721221223**
BÖLÜM: **BİLGİSAYAR MÜHENDİSLİĞİ**

Projenin;

KONUSU: **SHELL PROGRAMLAMA**

Dersin;

ADI: **İŞLETİM SİSTEMLERİ**
EĞİTMEN: **ALİ YILMAZ ÇAMURCU**
SAMET KAYA

İçindekiler

1- Özet	Hata! Yer işareti tanımlanmamış.
2- Proje Konusu.....	3
3- Proje Süresince Yapılanlar	4
4- Ek Açıklamalar	4
5- Kaynakça.....	5



1- Özet

Linux altında hızlı ve pratik programlama yapmanın en kısa yolu shell(kabuk) programlamadır diyebiliriz. İşletim sisteminin doğal komutlarını çalıştırma ve ek herhangi bir yorumlayıcı istememesi shell script(betik) dilini diğerlerinden ayıran en önemli faktördür. Yazdığınız bir kodu tüm Linux/UNIX sistemlerde değişiklik yapmadan kullanabilirsiniz. Linux altında shell(kabuk) programlama birçok farklı shell ile yapılabilir. Projemizde ise bununla ilgili örnekleri görüntüleyeceksiniz.

2- Proje Konusu

İşletim sistemini oluşturan kavramlar dediğimizde temel olarak çekirdek (kernel), kabuk (shell) ve komutlardan bahsedebiliriz. İşletim sistemini bu 3 kavram faaliyette tutar.

Kabuk, girilen komutları inceleyip çalıştırmak üzere gerekli kaynakları bulup çekirdeğe ileten bir yazılımdır. Komutların gerektirdiği kaynakları bulur ve kullanacak kaynaklara iletir.

Unix için açık kaynak olmasının getirdiği avantaj ile geliştirilmiş birçok kabuk programı vardır.

Çekirdek, kullanıcı tarafından girilen komutu, kabuk tarafından yorumlanıp kendisine iletildikten sonra işleme alır. Kabuk ile çekirdek birlikte çalışır.

Bu projemizde ise işletim sistemimizi oluşturan bu temel kavramlardan yola çıkarak birbiri ile iletişim halinde olan 3 farklı C dosyasının terminal üzerinde ki çalışmasını göreceğiz.

Proje içerisinde Fork, Exec ve Wait sistem çağrılarının yanı sıra sistem çağrılarını kullanacağız. Sistem çağrıları OS tarafından servisler için sağlanan arayüzdür. Doğrudan donanıma erişenler gibi düşük seviye görevler (tasks) assembly dilinde instructionlar içerdiği halde genellikle C, C++ dilleriyle yazılmışlardır. System call'lar sayesinde yazılımcı doğrudan donanıma müdahale etmez. Donanım üzerinde gerçekleştireceği işlemi system call kullanarak gerçekleştirir. Bu sayede olası sistem hatalarından kaçınılmış olur.

3- Proje Süresince Yapılanlar

Proje süresince başlangıç olarak proje analiz raporu oluşturuldu. Bu rapor geliştirici tarafından yapılması gerekenler ve hedefleri açıkça belirten bir rapordur. Bu sayede geliştirici geliştirme aşamasında ki adımları rahatlıkla takip edebilir.

Proje analiz raporuna göre yapılan programda ilk olarak proje özetimizde de bahsettiğimiz Fork, Exec ve Wait system çağrılarının ne olduğunu ve hangi durumlarda kullanıldığı araştırıldı. Örnek çalışmalar incelendi. Kaynak taraması yapıldı. Projemizde ihtiyacımız olan system call çağrıları göz önünde bulundurularak bir dosyanın boyutunu nasıl system call kullanarak okuyabiliriz gibi konularda araştırmalar yapıldı.

Yapılan bu araştırmalar neticesinde projenin uygulama safhasına geçildi. Öğretim görevlisi tarafından verilen proje raporunda eğitmenin istediği hususlar belirlendi. Bu hususlar göz önünde bulundurularak geliştirme başlandı.

Projemiz shell programlamasına yönelik olduğu için mevcut işletim sistemimiz üzerinde bir sanal makine oluşturuldu ve bu sanal makineye Linux Mint işletim sistemi kuruldu. Bu işletim sisteminde gerekli olan ana C dosyalarımız oluşturuldu ve istenilen parametrelere göre geliştirilmeye devam edildi.

Geliştirilme aşamasında C programlama dilinin fonksiyonları incelendi ve ihtiyacımıza yönelik fonksiyonlar kullanıldı. Geliştirilme sonlandığında projenin testleri yapıldı ve eğitmen tarafından verilmiş olan proje raporunda ki belirtilen hususlar kontrol edildi.

4- Ek Açıklamalar

Projemizde geliştirme amacıyla kullandığımız programlama dili C ' dir. C dili shell programlama konusunda uzman kişiler tarafından kullanılan diller arasında %80' lik bir orana tekabül etmektedir.

C programlama dili orta seviye bir dildir. Yani düşük seviye diller makine diline yakın olan çok kod yazma gerektiren diller, yüksek seviye diller ise az kod yazma işlemi gerektiren ama arka planda çok kod çalıştıran dillerdir. C programlama dili bu iki seviye arasındadır.

Windows, Linux ve Unix için C programlama dili kullanılarak program yazılabilir. Ayrıca C programlama dili Assembly dili ile geliştirilmiştir.

Bu projede bize sağlamış olduđu kaynaklardan ve öz verili bir şekilde öğrenebilmemiz için gayret gösteren değerli Öğretim Görevlilerimiz Ali Yılmaz Çamurcu ve Samet Kaya ' ya teşekkür ederiz.

5- Kaynakça

- 1-)<https://www.youtube.com/watch?v=aw9wHbFTnAQ>
- 2-)<https://stackoverflow.com/questions/238603/how-can-i-get-a-files-size-in-c>
- 3-)<https://www.geeksforgeeks.org/basics-file-handling-c/>
- 4-)[https://wiki.ubuntu-tr.net/index.php?title=Temel Linux komutlar%C4%B1](https://wiki.ubuntu-tr.net/index.php?title=Temel_Linux_komutlar%C4%B1)
- 5-)<https://stackoverflow.com/questions/6936096/how-to-close-an-application-itself-and-a-terminal>
- 6-)http://www.omurfurkanulu.com/blog/konu/egitim/c/c_fork.html
- 7-)https://www.tutorialspoint.com/c_standard_library/c_function_strcmp.htm
- 8-)<https://www.programiz.com/c-programming/c-pointer-functions>

TC.
FATİH
SULTAN
MEHMET
VAKIF ÜNİVERSİTESİ
2010



**T.C.
FATİH
SULTAN
MEHMET**
VAKIF ÜNİVERSİTESİ
2010