



**FATİH
SULTAN
MEHMET**
VAKIF ÜNİVERSİTESİ

Öğrencinin;

ADI: TALHA MAHMUT

SOYADI: DEREYURT

NO: 1921221040

BÖLÜM: BİLGİSAYAR MÜHENDİSLİĞİ

Projenin;

KONUSU: LİNX SHELL YAZIMI, FORK PROSES ÇAĞRISI, EXEC PROGRAM
ÇAĞRISI

Dersin;

ADI: OPERATING SYSTEMS

EĞİTMEN: ALİ YILMAZ ÇAMURCU

SAMET KAYA & TUĞÇE GÖKSU

İçindekiler

1- Proje Konusu.....	3
2- Proje Süresince Yapılanlar	3
3- Ek Açıklamalar	3
4- Ekran Çıktıları ve Başarı Ölçütleri	4
5- Kaynakça.....	4



**FATİH
SULTAN
MEHMET**
VAKIF ÜNİVERSİTESİ

1- Proje Konusu

Bu projede kendi linux shell programımızı yazmamız gerekiyor. Bu yazdığımız programımıza özel kendi komutlar olacaktır. Bu komutların da yine kendine özel çıktıları, yanıtları ve hata kodları olacaktır. Bizden istenen sistem programlama kapsamında bir C program yazmamızdır.

2- Proje Süresince Yapılanlar

Proje için ilk öncelikle Github üzerinden private bir repository oluşturdum. Burada projemde ilerledikçe gelişmeleri commit etcektim. Daha sonra gerçekleştirmemiz istenen komutların metotlarını yazdım. Bundan sonra menü oluşturdum burda aldığım girdiyi token mantığıyla bölüp yeni bir array'e atadım ve menü kontrollerimi buna göre gerçekleştirdim. Oluşturduğum array'in boyutunu bulmak için de boşluklara göre girdiyi bölerek integer bir değeri arttırdım. Sonrasında yazdığım bu metotların doğru çalışıp çalışmadığını test ettim. Daha sonra writef ve execx programları başka dosyalar olduğundan haberleşmek için pipe kullandım. Shell'den aldığım file name'ini writef programına yollayıp burda gelen dosyayı fopen metoduyla "append" modunda açarak dosya varsa üstüne yazmasını yoksa da oluşturmasını sağladım. Sonra gerekli parametreleri dosyaya yazdırma işlemini gerçekleştirdim. Execx için ise girdiden gelen sayı ve komutu pipe'a shell'den yazdım ve exec.c üzerinden okudum. Burada execx'den sonra gelen writef komutunu da execx.c dosyası üzerinden pipe'a yazıp döngü içerisinde kaç kere çalıştırmam gerekiyorsa o kadar çalıştırdım. Writef program da döngünün içinde çalıştığı için gerekli sayıda sıkıntısız bir şekilde çalıştı. Diğer komutları çalıştırırken main process'i bekletiyordum fakat execx için shell'de wait fonksiyonunu kullanmamamız gerekiyordu ve bu sayede uzun süren dosyaya yazım işlemleri devam ederken ben kendi yazdığım shell'im'i Rahat bir şekilde kullanabiliyorum.

3- Ek Açıklamalar

Ben c dosyalarının birbiriyle haberleşmesini pipe kullanarak gerçekleştirdim. Çok daha kolay bir şekilde iki-üç satır kod ile bunu sağladım. Ayrıca proje sayesinde fork mantığını da çok daha iyi şekilde anladım. Programda kullanılan komutların aşağıda kısa açıklamalarını görebilirsiniz:

ls : Bu komut bulunduğu dizinin içindeki dosyaları ekrana yazdırır.

bash : Bu komut kendi yazdığımız shell programından çıkarak linuxun kendi bash programına geçer.

clear : Terminal üzerindeki tüm geçmiş yazıları ekrandan siler.

exit : Terminali kapatarak myshell'den çıkış yapar.

cat : Girilen değeri istenilen şekilde ekrana yazar.

writef -f: Verilen dosyanın içine bizden istenilen değerleri yazar.

Execx -t: Girilen sayı kadar içine verilen programo çalıştırır.

4- Ekran Çıktıları ve Başarı Ölçütleri

```
myshell>>ls
execx  execx.c  makefile  myshell.c  myshell.o  program  writef  writef.c
myshell>>bash
talha@talha-VirtualBox:~/Masaüstü/opSisProject/TalhaMahmut_Dereyurt_1921221040_opsis_proj1/opSisProject/TalhaMahmut_Dereyurt_1921221040_opsis_proj1$ exit
exit
myshell>>cat my project
cat: my project
myshell>>writef -f os.txt
myshell>>execx -t 10 writef -f os.txt
myshell>>execx -t 5 writef -f abcd.txt
myshell>>abab
invalid value
myshell>>xdxd
invalid value
```

The screenshot shows a file manager window with a sidebar on the left containing icons for 'n', 'dizli', 'lgeler', 'lirilenler', 'izik', 'simler', and 'leolar'. The main area displays a grid of files: 'abcd.txt', 'execx', 'execx.c', 'makefile', 'myshell.c', 'myshell.o', 'os.txt', 'program', 'writef', and 'writef.c'. Below the file manager, two text editors are open. The first editor, titled 'os.txt', shows a log of system times and process IDs (pid, ppid) for 11 iterations. The second editor, titled 'abcd.txt', shows a similar log for 5 iterations.

Iteration	System Time	pid	ppid
1	Wed Nov 30 20:22:06 2022	5212	4943
2	Wed Nov 30 20:22:24 2022	5257	5256
3	Wed Nov 30 20:22:24 2022	5258	5256
4	Wed Nov 30 20:22:24 2022	5260	5256
5	Wed Nov 30 20:22:24 2022	5261	5256
6	Wed Nov 30 20:22:24 2022	5262	5256
7	Wed Nov 30 20:22:24 2022	5263	5256
8	Wed Nov 30 20:22:24 2022	5264	5256
9	Wed Nov 30 20:22:24 2022	5265	5256
10	Wed Nov 30 20:22:24 2022	5266	5256
11	Wed Nov 30 20:22:24 2022	5267	5256

Iteration	System Time	pid	ppid
1	Wed Nov 30 20:22:37 2022	5301	5300
2	Wed Nov 30 20:22:37 2022	5302	5300
3	Wed Nov 30 20:22:37 2022	5303	5300
4	Wed Nov 30 20:22:37 2022	5304	5300
5	Wed Nov 30 20:22:37 2022	5305	5300

5- Kaynakça

https://www.tutorialspoint.com/inter_process_communication/inter_process_communication_pipes.htm

<https://www.geeksforgeeks.org/fork-system-call/>