



**FATİH
SULTAN
MEHMET**
VAKIF ÜNİVERSİTESİ

Öğrencinin;

ADI : MUSTAFA
SOYADI : DEVECİ
NO : 1921221023
BÖLÜM : BİLGİSAYAR MÜHENDİSLİĞİ

Projenin;

KONUSU : Linux Shell Yazımı, fork proses çağırısı, exec program çağırısı

Dersin;

ADI : İŞLETİM SİSTEMİ
EĞİTMEN : PROF. DR. ALİ YILMAZ ÇAMURCU ARŞ.GÖR. SAMET KAYA

İçindekiler

1. Proje Hakkında Bilinmesi Gerekenler	3
2. Proje Konusu	3
3. Proje Özeti	3
4. Proje Süresince Yapılanlar	3
5. Proje kazanımları	8
6. Kaynakça	8



**FATİH
SULTAN
MEHMET
VAKIF ÜNİVERSİTESİ**

1. Proje Hakkında Bilinmesi Gerekenler

Shell Nedir?

Linux'un da içinde bulunduğu UNIX sistemlerinde komutları yorumlamak ve yönetmek için kullanılan programa kabuk (shell) denir. Komutların gerektirdiği kaynakları bulur ve kullanacak kaynaklara iletir. Bütün LINUX sistemlerde bulunması zorunlu olan birimdir.

Bash Nedir?

Bash, çoğu LINUX sisteminde varsayılan kabuk olarak ayarlanmıştır. Her kabuğun kendine özgü programlama dili yapısı vardır. Bash kabuğu ise güçlü programlama özellikleriyle karmaşık programların rahatça yazılmasına izin verir. Mantıksal operatörler, döngüler, değişkenler ve modern programlama dillerinde bulunan pek çok özellik bash kabuğunda da vardır ve işleyiş tarzları dandereyde diğerleriyle aynıdır. Bash çeşitli uyumluluk kiplerinde çalıştırılabilir, böylece farklı kabuklar gibi davranabilmektedir.

Fork Nedir?

Fork() komutu çalıştırıldığı anda child ve onu oluşturan process ise parent adını almaktadır. Child oluşturulduğu anda parent'ın hangi değişkeni var ise onun hepsinin kopyasını alır. Böylece child parentın bulunduğu konumdan itibaren çalışır. fork() komutu çalıştırıldığı anda child ve onu oluşturan process ise parent adını almaktadır. Child oluşturulduğu anda parentın hangi değişkeni var ise onun hepsinin kopyasını alır ve bulunduğu konumdan itibaren çalışır.

2. Proje Konusu

C programlama dili kullanılarak temel Linux Shell Yazımı kendine özgü çeşitli komutlar, yazdığımız programdan çıkmadan writef programını çalıştırıp geri myShell programımıza dönmesi için fork proses, wait çağrısı, exec program çağrısı işlemleri.

3. Proje Özeti

Temel bir shell programı yazıldı. Bu shell programının kendine özgü komutları, yanıtları, hata çıktıları tasarlandı. Proje kapsamında; writef, execx, cat komutu, terminali tamamen silen clear komutu ve terminalden çıkılmasını sağlayan bir exit komutu , dosya listesini gösteren ls komutu implemente edilmesi hedeflenmiştir.

4. Proje Süresince Yapılanlar

İlk olarak projemizin en önemli kısmı olan input alma işlemi için gerekli olan algoritma tasarlandı. scanf("%s", input); dizeleri okumak için scanf() fonksiyonu kullanıldı. Boşluklu dizeleri de okuyor ve bir satır karakteri ile sona eriyor.

Bunun yanında okuma işleminin sona ermesi için getchar(); fonksiyonu kullanıldı. Bu fonksiyonu yazmadığımız vakit okuma işlemi son bulmamakta programımız sonsuz döngüye girmektedir. Gerekli okuma işlemleri yapıldıktan sonra aldığımız bu inputun kullanıcının girdiği inputa projede istenilen

komutlara eşit mi? Sorusunun cevabını almak için strcmp() fonksiyonu kullanıldı böylelikle kullanıcının girdiği inputa göre projede istenilen komutlar ve yapacağı işlemler if else yapısı kurularak sağlandı. Programımız sürekli devam etmesi için while() döngüsüne alındı. Ayrıca kullanıcı yanlış input girerse yardımcı olması programı anlaması açısından help yardımcı komutu yazıldı.

Help -> Hatalı input giriyorsanız kullanıcıya yardımcı olması açısından yardım için help komutu girebilirsiniz yönlendirmesiyle kullanıcıya yardımcı olmak hedeflendi.

```
myshell>>yanlış input
Yanlış bir komut girdiniz, yardım için 'help' komutunu girebilirsiniz
myshell>>asdasd
Yanlış bir komut girdiniz, yardım için 'help' komutunu girebilirsiniz
myshell>>help

Operations:
cat - cat komutu sonrası yazdığınız şeyleri yazdırır.
clear - ekranı temizler
ls - klasör içindeki dosyaları görebilirsiniz.
exit - myShell'den çıkış işlemi.
bash - sistemdeki bash programı çağrılır, bashteki normal işlemleri yapabilirsiniz, myshell için exit yazmanı
z yeterlidir
execx - execx -t kaç kere çalıştırmak istediğiniz (çalıştırmak istediğiniz programı giriniz) örnek kullanım
execx -t 3 help -> help komutu 3 kere çalıştırılır.
writef - writef -f (filename) filename verilir dosya varsa içine ekleme yapılır yoksa oluşturulup içine yazar
. örnek kullanım writef -f deneme -> deneme dosyayı olmadığı için içine oluşturulup sistem tarihi, pid, ppid
değerleri yazıldı.

myshell>>
```

Bash -> Sistem bash'e giriş yapar ona özgü komutlar çalıştırılır burda da fork kullanıldı bu sayede myShell uygulamamız kapanmadan bash işini tamamlayıp çıkacaktır.

```
myshell>>bash

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[bash-3.2$ echo myShell uygulaması çalışıyor ama ben şu an bana özgü olan echo komutunu çalıştırmaktayım.
myShell uygulaması çalışıyor ama ben şu an bana özgü olan echo komutunu çalıştırmaktayım.
[bash-3.2$ exit
exit
myshell>>
```

Ls -> Komutu klasör içindeki dosyaların gösterimini sağlar.

```
myshell>>ls
total 5120
drwxr-xr-x@ 6 mstfdvc staff 192 Oct 25 19:50 iOS çalışmalarım
drwxr-xr-x@ 4 mstfdvc staff 128 Nov 1 15:19 mapApp
drwxr-xr-x@ 18 mstfdvc staff 576 Nov 26 21:32 turkcell
-rw-r--r--@ 1 mstfdvc staff 89380 Nov 28 19:21 1.png
drwxr-xr-x@ 10 mstfdvc staff 320 Nov 29 21:47 deneme
drwxr-xr-x@ 11 mstfdvc staff 352 Nov 30 00:21 shell
-rw-r--r--@ 1 mstfdvc staff 7786 Nov 30 01:25 myShell.c
-rwxr-xr-x 1 mstfdvc staff 34369 Nov 30 01:25 main
-rwx-----@ 1 mstfdvc staff 1986061 Nov 30 01:29 Proje Rapor Taslak.pages
myshell>>
```

Clear -> Ekranın temizlenmesini sağlamaktadır.

Öncesi:

```
drwxr-xr-x@ 18 mstfdvc staff      576 Nov 26 21:32 turkcell
-rw-r--r--@  1 mstfdvc staff    89380 Nov 28 19:21 1.png
drwxr-xr-x@ 10 mstfdvc staff      320 Nov 29 21:47 deneme
drwxr-xr-x@ 11 mstfdvc staff      352 Nov 30 00:21 shell
-rw-r--r--@  1 mstfdvc staff     7786 Nov 30 01:25 myShell.c
-rwxr-xr-x  1 mstfdvc staff    34369 Nov 30 01:25 main
-rwx-----@  1 mstfdvc staff   1986061 Nov 30 01:29 Proje Rapor Taslak.pages
myshell>>asdasd
Yanlış bir komut girdiniz, yardım için 'help' komutunu girebilirsiniz
myshell>>help

Operations:
cat - cat komutu sonrası yazdığınız şeyleri yazdırır.
clear - ekranı temizler
ls - klasör içindeki dosyaları görebilirsiniz.
exit - myShell'den çıkış işlemi.
bash - sistemdeki bash programı çağrılır, bashteki normal işlemleri yapabilirsiniz, myshell için exit yazmanı
z yeterlidir
execx - execx -t kaç kere çalıştırmak istediğiniz (çalıştırmak istediğiniz programı giriniz) örnek kullanım
execx -t 3 help -> help komutu 3 kere çalıştırılır.
writef - writef -f (filename) filename verilir dosya varsa içine ekleme yapılır yoksa oluşturulup içine yazar
. örnek kullanım writef -f deneme -> deneme dosyayı olmadığı için içine oluşturulup sistem tarihi, pid, ppid
değerleri yazıldı.

myshell>>
```

Sonrası:

```
myshell>>
```

Exit -> Programdan çıkış işlemi yapılmakta.

```
myshell>>exit

Time: Wed Nov 30 14:41:00 2022

myShell'den çıkış yapıldı.
(base) mstfdvc@mustafas-MacBook-Air deneme %
```

Cat -> Normal bash kabuğunun echosu gibi çalışmakta index index okuma için yazdığım subStringFinder() fonksiyonu sayesinde projede istenildiği gibi:

input ->> cat xxx Output ->> cat:xxx olarak yazması sağlandı

```
myshell>>cat tam istenildiği gibi çalışmaktadır
cat:tam istenildiği gibi çalışmaktadır
myshell>>
```

Bu komutların yanında sırayla kullanıcının girdiği inputu index index okuması için subStringFinder() fonksiyonu yazıldı. Bu fonksiyon `execx -t 3` (çalıştırılacak program) inputunun ve `writeln -f (fileName)` inputunun alınması açısından index index baktığı için bizlere kolaylık sağlamakta.

Execx komutu

Kullanımı : `execx -t` (kaç kere çalıştırmak istediğiniz) (çalıştırmak istediğiniz komut)

Execx yapımında index index inputu kontrol etmesi için subStringFinder() fonksiyonu kullanıldı. Bunun yanında for ile kaç kare çalıştıracağımızı bulmak için kullanıcının yazdığı karakter dizini integer değere çeviren atoi kullanıldı buna numberOfTime ismi verildi for yapısının kaç kere çalışması gerektiği belirlendi ve kullanıldı. For içine de hangi program çalıştırmak istiyorsa newInput isimli değişkene index index okumamızı sağlayan subStringFinder() fonksiyonu sayesinde atandı. Ama bu işlemler execx programı içinde yapıldı projede istenilen execx programı çalıştırılacak fakat myShell programımız sonlanmayacaktı işte burada fork() devreye girdi. İçine exec ile diğer programı okuma işlemi yapıldı.

```
myshell>>execx -t 2 help

Operations:
cat - cat komutu sonrası yazdığınız şeyleri yazdırır.
clear - ekranı temizler
ls - klasör içindeki dosyaları görebilirsiniz.
exit - myShell'den çıkış işlemi.
bash - sistemdeki bash programı çağrılır, bashteki normal işlemleri yapabilirsiniz, myshell için exit yazmanı
z yeterlidir
execx - execx -t kaç kere çalıştırmak istediğiniz) (çalıştırmak istediğiniz programı giriniz) örnek kullanım
execx -t 3 help -> help komutu 3 kere çalıştırılır.
writef - writef -f (filename) filename verilir dosya varsa içine ekleme yapılır yoksa oluşturulup içine yazar
. örnek kullanım writef -f deneme -> deneme dosyayı olmadığı için içine oluşturulup sistem tarihi, pid, ppid
değerleri yazıldı.

Operations:
cat - cat komutu sonrası yazdığınız şeyleri yazdırır.
clear - ekranı temizler
ls - klasör içindeki dosyaları görebilirsiniz.
exit - myShell'den çıkış işlemi.
bash - sistemdeki bash programı çağrılır, bashteki normal işlemleri yapabilirsiniz, myshell için exit yazmanı
z yeterlidir
execx - execx -t kaç kere çalıştırmak istediğiniz) (çalıştırmak istediğiniz programı giriniz) örnek kullanım
execx -t 3 help -> help komutu 3 kere çalıştırılır.
writef - writef -f (filename) filename verilir dosya varsa içine ekleme yapılır yoksa oluşturulup içine yazar
. örnek kullanım writef -f deneme -> deneme dosyayı olmadığı için içine oluşturulup sistem tarihi, pid, ppid
değerleri yazıldı.

myshell>>
```

`execx -t 2 help` -> komutu ile 2 kere help komutu çalıştı

[illegible]

Writef komutu

Kullanımı : writef -f (fileName)

Eğer filename isimli dosya varsa sistem tarihini, pid değerini ve ppid değerini içine ekler , eğer böyle bir isimde dosya yoksa dosya oluşturulup sistem tarihini, pid değerini ve ppid değerini içine yazar.

Writef yapımında da execx'de olduğu gibi index index inputu kontrol etmesi için subStringFinder() fonksiyonu kullanıldı. Çünkü burada önemli olan filename'i almamızdı. Filename'ı aldık fakat projede istenilen writef isimli bir program yazmamız ve yine execx de olduğu gibi myShell'den çıkmadan writef çalıştırılıp sonra myShell uygulamamıza devam etmesiydi bu yüzden burda da fork() kullanıldı. Okuma işlemi yapması için execx kullanımıyla projemiz sonlandı.

Aşağıdaki görselde görüldüğü üzere öncelikle ls komutu ile deneme adlı dosya var mı diye kontrol edildi dosya yok. Olmadığı için writef -f deneme komutu ile dosya oluşturulup içine projede istenilen sistem tarihi, pid ve ppid değerleri yazdırıldı.

```
myshell>>ls
total 11608
drwxr-xr-x@ 6 mstfdvc staff 192 Oct 25 19:50 iOS çalışmalarım
drwxr-xr-x@ 4 mstfdvc staff 128 Nov 1 15:19 mapApp
drwxr-xr-x@ 18 mstfdvc staff 576 Nov 26 21:32 turkcell
-rw-r--r--@ 1 mstfdvc staff 89380 Nov 28 19:21 1.png
drwxr-xr-x@ 10 mstfdvc staff 320 Nov 29 21:47 deneme
drwxr-xr-x@ 11 mstfdvc staff 352 Nov 30 00:21 shell
-rw-r--r--@ 1 mstfdvc staff 7786 Nov 30 01:25 myShell.c
-rwxr-xr-x 1 mstfdvc staff 34369 Nov 30 01:25 main
-rwx-----@ 1 mstfdvc staff 5367960 Nov 30 01:36 Proje Rapor Taslak.pages
myshell>>writef -f deneme
myshell>>ls
total 11616
drwxr-xr-x@ 6 mstfdvc staff 192 Oct 25 19:50 iOS çalışmalarım
drwxr-xr-x@ 4 mstfdvc staff 128 Nov 1 15:19 mapApp
drwxr-xr-x@ 18 mstfdvc staff 576 Nov 26 21:32 turkcell
-rw-r--r--@ 1 mstfdvc staff 89380 Nov 28 19:21 1.png
drwxr-xr-x@ 10 mstfdvc staff 320 Nov 29 21:47 deneme
drwxr-xr-x@ 11 mstfdvc staff 352 Nov 30 00:21 shell
-rw-r--r--@ 1 mstfdvc staff 7786 Nov 30 01:25 myShell.c
-rwxr-xr-x 1 mstfdvc staff 34369 Nov 30 01:25 main
-rwx-----@ 1 mstfdvc staff 5367960 Nov 30 01:36 Proje Rapor Taslak.pages
-rw-r--r-- 1 mstfdvc staff 54 Nov 30 01:37 deneme
myshell>>
```

```
deneme
Tarih: Wed Nov 30 01:25:33 2022
pid: 1380
ppid: 1216
```

Proje de fork() exec() yeni bir işlem (process) oluşturmaya yarayan fork() ve exec() fonksiyonları kullanıldı. Ne için kullanıldı? Fork işlemi çalıştığı anda bulunduğu programın bir kopyasını oluşturuyor ve çalıştığı andan itibaren iki ayrı dal olarak aynı kodu çalıştırıyor. Küçük nüans farklılıkları yanında aynı programı iki defa çağırırsan da olabilir diye düşünülebilir. Fakat fork işleminin asıl amacı çalışan ana programın yanında farklı programları ve program parçalarını da çalıştırabiliyor olmasıdır. Projede istenilen de zaten myShell çıkmadan gerekli olan writef ve execx programlarının çalıştırılıp geri myShell'e dönmesiydi.

Exec komutu ile başka bir programı programımızın içerisinden çalıştırabiliriz. Exec işlemi sonrasında çalışan prosesin pid değeri ve control bloğu değişmez, proses yaşamını başka bir programı çalıştırmak için devam ettirir.

pid ve ppid deęerleri nedir?

Bir process programın bir rneęini kořar, process id iin writef iine getpid() yazdırıldı bunun yanısıra her process bir parent process'e sahiptir, ebeveyn process ocuk process retir. parent process id iin getppid() ile writef iine yazıldı.

Wait komutu neden kullanıldı?

Wait komutu child proses işini bitirene kadar ana prosesin bitmemesini saęlar. Bařka bir deyiřle "wait" komutu ile ana proses ocuklarının işini bitirmesini bekler. "Wait" komutu geri ocuk prosesin pid deęeri dner.

zetle:

Unix sistemlerde her proses "fork" system aęrısıyla oluřturulur. Fork onu aęıran prosesin bir kopyasını oluřturur. aęrılan process forktan yeni bir pid ile dner bu ocuk(child) proses olarak adlandırılır. yeni oluřturulan process ebeveynini(parent) kopyalar. ocuk process fork aęrı fonksiyonundan geri "0" deęerini dndrr.

5. Proje kazanımları

Bu proje bana c bilgilerimin yeniden tazelenmesine bunun yanısıra asıl olarak fork() process aęrısıyla ikinci programın alıřtırılıp ilk programın kapanmadan arkada bekleyeceęini daha sonra işini tamamladıktan sonra ilk programa geri dneceęini kazandırmıřtır. Exec() fonksiyonu sayesinde ikinci programın nasıl aęrılacaęı ve farklı kullanımları bulunan exec'in duruma gre hangi parametreler alacaęını kazandırmıřtır. Bunların yanında c'de kullanılan atoi(), malloc() gibi fonksiyonların kullanımı bu proje sayesinde bizlere kazandırmıř oldu.

6. Kaynaka

https://medium.com/@kahil_kubilay/linux-dnyası-kabuk-nedir-959706cc658f

<https://clearingtutorial.blogspot.com/2014/05/c-de-fork-ve-exec-kavram.html>

<https://linuxhint.com/read-lines-stdin-c-programming/>

<https://stackoverflow.com/questions/2144459/using-scanf-to-accept-user-input>