



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
İŞLETİM SİSTEMLERİ
PROJE RAPORU

B211210572 -> Yazan Alsolyman

B211210584 -> Arif Muhammed

B211210575 -> Ahmed Mohamed

G211210564 -> Muhammed Elzabi

G211210036 -> Abdullah Bünyamin Adsever

Proje Tanımı:

Bu proje, bir komut satırı yorumlayıcısı (kabuk) uygulamasını geliştirmeyi hedefler. Proje kapsamında kullanıcı komutlarını alabilen, komutları çalıştıran, giriş/çıkış yönlendirmesi yapan, boru (pipe) mekanizmasını destekleyen ve arka planda işlem çalıştırabilen bir sistem tasarlanmıştır.

1) Prompt

Prompt, kullanıcıdan sürekli olarak komut almayı sağlayan bir komut istemidir. Her komut tamamlandıktan veya bir arka plan komutu başlatıldıktan sonra kullanıcıya tekrar gösterilir.

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> |
```

2) Built-in Komut (quit)

Eğer quit girilmişse, programdaki ana döngü sonlandırılır. Arka planda çalışan süreçler varsa, tüm süreçlerin tamamlanması beklenir.

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> quit
yazan@Yazan:~/Proje$ |
```

3) Tekli komutlar

Kullanıcıdan alınan tek bir komutu çalıştırmak.

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> ls -l
Readme.txt
bin
include
lib
makefile
src
test.txt

> |
```

4) Giriş yönlendirme

Bir komutun girdisinin standart giriş (klavye) yerine bir dosyadan alınmasını sağlar.

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> cat < test.txt
Sakarya Üniversitesi
> cat < file.txt
file.txt giriş dosyası bulunamadı

> |
```

5) Çıkış yönlendirme

Bir komutun çıktısını standart çıktı (ekran) yerine bir dosyaya yazdırmak için kullanılır.

Örnekte file.txt dosyasının içeriği okunur ve file2.txt dosyasına yönlendirilir. Eğer file2 yoksa oluşturulur, varsa içeriği silinip üzerine yazılır.

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> cat file.txt > file2.txt

> cat < file2.txt
Hello from linux

> |
```

Başka örnekte increment.c bir dosya oluşturup bu kodu yazdık.

- İlk komutta 12, test.txt dosyasına yazılır (**çıkış yönlendirme**).
- İkinci komutta test.txt dosyasındaki değer programın girdisi olarak alınır (**giriş yönlendirme**) ve artırılmış hali ekrana yazdırılır.

```
#include <stdio.h>
int main() {
    int i;
    scanf("%d", &i);
    i++;
    printf("%d\n", i);
    return 0;
}
```

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> echo 12 > test.txt

> ./bin/increment < test.txt
13

> |
```

6) Arkaplan çalışma

Bir komutun terminalde çalışırken ana sürecin engellenmeden çalıştırılmasıdır. Kullanıcı başka komutlar yazabilirken, arka plandaki işlem devam eder. Bu özellikle uzun süreli veya zaman alan komutları çalıştırmak için kullanılır.

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> sleep 5 &

> cat test.txt
Hello World!
> [984] retval: 0
|
```

7) Boru (pipe)

Bir komutun çıktısını alıp başka bir komutun girdisi olarak yönlendiren bir mekanizmadır.

Soldaki fotoğrafta test.txt dosya içerisinde 4 satır yazdık ve her satırda "pipe" kelimesini yazdık.

Sağdaki fotoğrafta > cat test.txt | grep pipe | wc -l komutuyla:

- test.txt dosyasını okur ve içeriğini borunun girişine gönderir.
- grep ile gelen içeriği tarar ve "pipe" kelimesini içeren satırları filtreler.
- "pipe" kelimesini içeren satırların sayısını sayar.

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> cat test.txt
That's a pipe line
Another pipe line here
Yet another pipe line
pipes are so useful
> |
```

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> cat test.txt | grep pipe | wc -l
4

> |
```

noktalı virgül (;) örneği

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> echo 12 | ./bin/increment
13
[406] retval: 0

> |
```

```
yazan@Yazan:~/Proje$ make
gcc -g ./src/main.c -o ./bin/program
gcc -g ./src/increment.c -o ./bin/increment
./bin/program

> echo 12; sleep 2; echo 13
12
13

> |
```

makefile dosyası

```
all: derle calistir

derle:
    gcc -g -I./include ./src/main.c ./src/handler.c -o ./bin/program
    gcc -g -I./include ./src/increment.c -o ./bin/increment

calistir:
    ./bin/program
```

Github linki : <https://github.com/4yznx/Isletim-Sistemleri-Projesi>