

## BLGM 343 – DENEY 1 \*

### GİRİŞ

#### *Amaçlar*

1. GNU/Linux İşletim sistemlerinin kullanımı
2. GCC kullanarak program derleme
3. Komut satırı parametreleri

#### *Ön Bilgi*

Görsel arayüz günlük kullanımda birçok kolaylık sağlamasına rağmen, bilgisayar mühendisleri olarak bizlerin, sıkça komut satırını kullanmamız gerekmekte veya komut satırının kullanımı işimizi kolaylaştırıp hızlandırabilmektedir.

GNU/Linux'da bulunan komut satırı programlarının genel olarak ortak özellikleri vardır. Bunlardan bilmemiz gereken en önemli kısım komut satırı parametrelerinin kullanımudur. Komut satırı parametreleri, programın adından sonra boşluk bıraktıktan sonra yazılabilir. Dosya ismi ile birlikte programlama iletilecek parametreler komut satırında yazılabilir. Ancak bunların dışında programın ayarlarını etkileyen anahtarlar mevcuttur. Tek harf anahtarlar “-” karakterinden sonra yazılabilir. Bu anahtarlar ek parametre isteyebilir, ancak istemiyorlarsa, birden fazlası yan yana yazılabilir. Eğer kullanacağınız anahtar birden fazla karakter içeriyorsa, başına “--” koymanız gerekmektedir. Ayrıca “\*.h” şeklinde tüm .h ile biten dosyaları bir programa parametre olarak gönderebilirsiniz. “~” özel karakteri ise sizin kendi dizininizi kullanmanızı sağlar.

Aşağıda sıkça kullanılan temel komut satırı programları listelenmiştir

- **ls:** bulunduğunuz dizindeki dosyaları listeler. Eğer detaylar (dosya boyutu, tarih gibi isteniyorsa “l” anahtarı, dosya boyutlarının kolay okunması için “h” anahtarı kullanılabilir, gizli dosyaları göstermek için “a” kullanılabilir. Ör: ls -la
- **cd:** bu komut şu anki dizini değiştirir. Eğer / ile başlıyorsa dosya sisteminin en başından, yoksa şu an bulunduğunuz dizinden işlem yapmaya başlar. “..” bir üst dizin anlamına gelmektedir.

---

\* BLGM 343 dersi için Güz 2013/2014 döneminde Gürcü Öz ve Cem Kalyoncu tarafından hazırlanmıştır

- `rm`: dosyaları silmek için kullanılabilir. Eğer bir dizin silinecekse “`r`” anahtarı, soru sormadan direkt olarak silmesini isterseniz “`f`” anahtarı kullanılabilir. Ör: `rm -rf birdizin`
- `cp`: dosyaları kopyalamak için kullanılabilir. “`r`” anahtarı bir dizini kopyalar. Hedef dosyaların üzerine sormadan yazması için “`f`” anahtarını kullanabilirsiniz.

Örn: `cp *.cpp ~/lab3`

- `mkdir`: yeni bir dizin oluşturur

Eğer bir komut hakkında ek bilgiye ihtiyacınız olursa, “`help`” anahtarını kullanabilirsiniz. Ancak burada verilen yardım genel olarak İngilizcedir.

## **GCC**

`gcc`, GNU/Linux işletim sistemlerinde kullanılan bir derleyici koleksiyonudur. Temel olarak C kodlarını derlemesine rağmen, bir çok derleyici GCC'ye eklenebilmektedir. Biz Sistem Programlama dersinde, C kodlarıyla program yazacağımız için herhangi bir ek ayara ihtiyaç duymamaktayız. GCC'nin temel kullanımı oldukça basittir. “`o`” anahtarıyla birlikte derlenen programın adını verdikten sonra, kod dosyalarının adlarını yazmamız yeterlidir. Örneğin: `test.c` dosyasını `test` isimli program olarak derlemek istersek

```
gcc -o test test.c
```

yazmamız yeterlidir. Bazı programlar için birden fazla kod dosyası kullanımı gerekli olabilir. Bu durumlarda bu dosyaları aralarına boşluk koyarak derleyebilirsiniz. Ayrıca bir dizindeki tüm C dosyalarını tek bir programa derlemek isterseniz

```
gcc -o program *.c
```

şeklinde bir komut çalıştırabilirsiniz. Bazı durumlarda programınıza ek kütüphane eklemeniz gerekebilir. Bu işlemi “`l`” anahtarıyla yapabilirsiniz. Örneğin:

```
gcc -o program -lpthread program.c
```

Eğer hata olmazsa, GCC mesaj vermeden komut satırına dönecektir. Hatalarınız varsa, bunları düzelterek tekrar programınızı derleyiniz. Derlemiş olduğunuz programları çalıştırmak için, program adınızın önüne “`./`” koymanız yeterlidir. Örneğin:

```
./program
```

## ***Komut satırı parametreleri***

Bizler de GNU/Linux programları gibi komut satırı parametrelerini programımızın akışını

etkilemek için kullanabiliriz. C programlama dilinde bulunan ve programın giriş noktasını oluşturan **main** fonksiyonuna istenmesi durumunda komut satırı parametrelerini alması için ilgili parametreler eklenebilir. Bu parametreler sırasıyla parametre sayısı (**argc**, tam sayı, **int**) ve parametre dizidir (**argv**, sayı dizisi dizisi, char\* []). Komut satırı parametrelerinin ilki her zaman programın adıdır. Bu durumda eğer bir program aşağıdaki şekilde çalıştırılırsa

```
./program a b c
```

Programdaki **argc** değeri 4, **argv** dizisinin içeriği ise {"program", "a", "b", "c"} olur.

Aşağıda örnek bir program mevcuttur. Bu program çalışmak için en az bir parametre istemektedir. Eğer parametre verilmezse, program kullanım bilgisini yazdırmaktadır.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    int i;

    if(argc < 2) {
        printf("Kullanım: %s parametreler\n", argv[0]) ;
        exit (1) ;
    }

    printf("Çalıştırılan program %s \n", argv[0]) ;
    printf("Parametrelerin sayısı%d \n", argc-1) ;

    for(i=1; i<argc; ++i)
        printf("argv[%d]: %s\n",i, argv[i]);

    exit(0) ;
}
```

## Deney

Aşağıda verilen komut satırına göre UNIX de çalıştırılabilen bir C program yazıp sonuçlarını belirleyiniz. Program girilen üç parametreyi topladıktan sonra parametreleri ve sonucu ekrana yazdırıp bitecektir. Programın kullanımını da yazmayı unutmayınız.

İpucu: gelen komut satırı parametrelerini dizgi tipinden sayıya çevirmek için **atoi** kütüphane fonksiyonunu kullanabilirsiniz. atoi fonksiyonu <stdlib.h> kütüphanesinin içerisinde bulunmaktadır.

Örnek kullanım:

```
./topla 3 5 6
14
```