

# Sistem Programlama

## Ders 7

Dr. Öğr. Üyesi Mehmet Dinçer Erbaş  
Bolu Abant İzzet Baysal Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü

# Dosya zamanları

- Her dosya için üç zaman bilgisi tutulur:

Bilgi	Tanım	Etkileyen fonksiyon örneği	ls (1) opsiyonu
st_atime	Dosyanın son erişilme zamanı	read	-u
st_mtime	Dosya verisinin son değiştirilme zamanı	write	Varsayılan
st_ctime	i-node durumunun son değişme zamanı	chmod, chown	-c

# Dosya zamanları

- utime fonksiyonu: Dosya erişim ve değiştirilme zamanı utime fonksiyonu ile değiştirilebilir.

```
#include <utime.h>
```

```
int utime(const char *pathname, const struct utimbuf *times);
```

Dönüş: OK ise 0, hata ise -1.

- Eğer zaman değeri NULL verilirse, dosya erişim ve dosya değişim zamanı şu anki zamana eşitlenir.
  - Bu işlem için dosyanın sahibi olmalısınız veya dosya üzerinde yazma hakkınız olmalı.
- Eğer zaman değeri NULL değil ise, timeval timbuf yapısı içerisinde verilen değere göre dosya erişim ve dosya değişim zamanları ayarlanır.
  - Bu işlem için dosyanın sahibi olmalısınız. Dosyaya yazma hakkı yeterli değildir.
- Her iki işlem sonucunda stc\_time değeri şu anki zaman ayarlanır.
- Örnek 19.

# mkdir ve rmdir fonksiyonları

- Klasörler mkdir fonksiyonu ile oluşturulur.

```
#include <sys/stat.h>
```

```
int mkdir(const char *path, mode_t mode);
```

Dönüş: OK ise 0, hata ise -1.

```
#include <unistd.h>
```

```
int rmdir(const char *pathname);
```

Dönüş: OK ise 0, hata ise -1

- Eğer klasöre işaret eden link sayısı 0 olursa (fonksiyon çalıştıktan sonra) ve başka bir işlem klasörü açık tutmuyorsa, klasör silinir.
- Bu işlemin yapılabilmesi için klasörün boş olması gerekir (sadece . ve .. olmalı)

# Klasörleri okumak

- Bu fonksiyonlar kullanılarak bir klasörün içeriğine erişilebilir.

```
#include <dirent.h>
```

```
DIR *opendir(const char *filename);
```

Dönüş: OK ise işaretçi, hata ise NULL

```
struct dirent *readdir(DIR *dp);
```

Dönüş: OK ise işaretçi, hata ve klasör sonunda NULL

```
void rewinddir(DIR *dp);
```

```
int closedir(DIR *dp);
```

Dönüş: OK ise 0, hata ise -1.

```
long telldir(DIR *dp);
```

Dönüş: dp işaretçisi ile alakalı klasördeki bulunan pozisyon.

```
void seekdir(DIR *dp, long loc);
```

# Klasörleri okumak

- dirent yapısının içeriği şu şekildedir:

```
struct dirent {  
    ino_t d_ino;                // i-node numarası  
    char d_name[NAME_MAX + 1]; // NULL ile sonlanan dosya ismi  
}
```

- Örnek1

# chdir, fchdir ve getcwd fonksiyonları

- Çağırılan işlemin mevcut çalışma klasörünü chdir ve fchdir fonksiyonları ile değiştirebiliriz.

```
#include <unistd.h>
```

```
int chdir(const char *pathname);
```

```
int fchdir(int filedes);
```

Dönüş: OK ise 0, hata ise -1.

– Örnek20.

- Çağırılan işlemin mevcut çalışma klasörünü getcwd fonksiyonu ile öğrenebiliriz.

```
#include <unistd.h>
```

```
char *getcwd(char *buf, size_t size);
```

Dönüş: OK ise buf, hata ise NULL.