Ders 48

```
time_t time(time_t *p);


struct tm {
    int tm_year;  //1900
    int tm_mon;   //0
    int tm_mday;   //
    int tm_wday;  //0 pazar
    int tm_yday;  // 0 1 ocak
    int tm_hour;  //
    int tm_min;  //0 - 59
    int tm_sec;  //0 - 59
    int tm_isdst;

};

< 0
0
```

Tm yapısının içeriği

Statik ömürlü bir nesne adresi döndürüyor.

Örnek localtime fonksiyonunu kullanalım.

```
struct tm* localtime(const time_t*);

static const char* const pmons[] = {
    "Ocak",
    "Subat",
    "Mart",
    "Nisan",
    "Mayis",
    "Haziran",
    "Temmuz",
    "Agustos",
    "Eylul",
    "Ekim",
    "Kasim",
    "Aralik"
};
```

```
nt main(void)

   static const char* const pmons[] = { ... }
   static const char* const pdays[] = {
   "Pazartesi",
   "Sali",
   "Carsamba",
   "Persembe",
   "Cuma",
   "Cumartesi",
   "Pazar",
   };
```

Hastanın ve ayın günlerini göstermek için ddizi tanımladık.

```
struct tm* localtime(const time_t*);
```

```
int main(void)
{
    static const char* const pmons[] = { ... }
    static const char* const pdays[] = { ... }

    time_t sec;
    time(_Time:&sec);
    struct tm* p = localtime(_Time:&sec);

    printf(_Format:"%02d %s %d %s %02d:%02d:%02d\n", p->tm_mday, pmons[p->tm_mon], p->tm_year + 1900, pdays[p->tm_wday],
      p->tm_hour, p->tm_min, p->tm_sec);

}
```

```
Microsoft Visual Studio Debug Console
27 Eylul 2022 Sali 19:46:16

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 25904) exited with code 0.
Press any key to close this window . . .
```

Burada aldığımız mon ve w day değerlerini diziye gönderip ekrana yazdırdık.

//

Örnek bir dosyaya dosyanın oluşturulduğu tarihi yazmak

```
//'function': 'const char *' differs in levels of indirection from 'char *(__cdecl *)(void)'

char* get_log_file_name(void)
{
    static char buffer[BUFFER_SIZE];
    time_t sec;
    time(&sec);
    struct tm* p = localtime(&sec);

    sprintf(buffer, "%d_%02d_%02d_%02d_%02d_%02d.log", p->tm_year + 1900,
        p->tm_mon + 1,
        p->tm_mday,
        p->tm_hour,
        p->tm_min,
        p->tm_sec);

    return buffer;
}


int main(void)
{
    puts(get_log_file_name);
}
```

Burada bir hata var ara bulmaya çalış

Hata nedeni puts argümanı

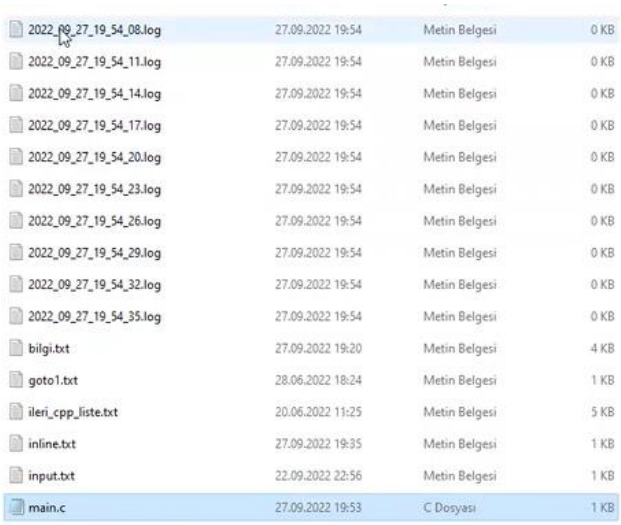
Biz puts ile fonksiyonun geri döndürdüğü değeri kullanmak istiyoruz.

Ama fonksiyonun adresini göndermişiz geri dönüş değerini değil. () parantez koymak gerekiyor.

```
int main(void)
{
    puts(get_log_file_name());
}
```

Dosyaları oluşturalım.

```c
#define         BUFFER_SIZE     100

//'function': 'const char *' differs in levels of indirection from 'char *(__cdecl *)(void)'

char* get_log_file_name(void)
{
    static char buffer[BUFFER_SIZE];
    time_t sec;
    time(&sec);
    struct tm* p = localtime(&sec);

    sprintf(buffer, "%d_%02d_%02d_%02d_%02d_%02d.log", p->tm_year + 1900,
        p->tm_mon + 1,
        p->tm_mday,
        p->tm_hour,
        p->tm_min,
        p->tm_sec);

    return buffer;
}


int main(void)
{
    for (int i = 0; i < 10; ++i) {
        FILE* f = fopen(get_log_file_name(), "w");
        printf("%d\n", i + 1);
        fclose(f);
        Sleep(3000);
    }
}
```



10 adet dosya oluşturdum ve bu dosya isimleri 3 saniye aralıklar ile oluşturulduğu tarih.

//

Örnek ilk örneğin üsütüne gidelim.

Local time yerine gmtime fonksiyonu

```
"Mayis",
"Haziran",
"Temmuz",
"Agustos",
"Eylul",
"Ekim",
"Kasim",
"Aralik"
};
static const char* const pdays[] = {
"Pazar",
"Pazartesi",
"Sali",
"Carsamba",
"Persembe",
"Cuma",
"Cumartesi",
};

time_t sec;
time(&sec);
struct tm* p = gmtime(&sec);

printf("%02d %s %d %s %02d:%02d:%02d\n", p->tm_mday, pmons[p->tm_mon], p->tm_year + 1900, pdays[p->tm_wday],
  p->tm_hour, p->tm_min, p->tm_sec);
```

Tek farkı greenwich 0 kordinatının saatini yazdırmak için kullanılıyor.

Utc deniyor buna

Örnek ctime

```
#include <time.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    time_t timer;
    time(&timer);

    char* p = ctime(&timer);
    printf("length = %zu\n", strlen(p));
    printf("(%s)\n", p);
}
```

```
Microsoft Visual Studio Debug Console
length = 25
(Tue Sep 27 20:05:16 2022
)
```

Ctime fonksiyonu 25 karakterlik bir yazı verir bu yazının sonunda null karakteri ve bir \n alt satır karakteri var bu yüzden parantez bir alt satırda yazıldı.

```c
#include <time.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    time_t timer;
    time(&timer);

    timer -= 1'000'000;

    char* p = ctime(&timer);



    printf("length = %zu\n", strlen(p));
    printf("%s\n", p);
}
```

1 milyon saniye öncesini yazdıralım

Örnek astime c tmi birlikte

```c
int main(void)
{
    time_t timer;
    time(&timer);
    struct tm* p = localtime(&timer);

    char* pstr_1 = ctime(&timer);
    char* pstr_2 = asctime(p);

    printf("%p  %p\n", pstr_1, pstr_2);
}
```

Bu iki pointer aynı adresi gösteriyor çünkü aynı değerleri veriyor standartlar bunu destekliyor.

Printf ile yazıyı yazdırdım formatı aynı.

```
Microsoft Visual Studio Debug Console
00FED698  00FED698
Tue Sep 27 20:09:13 2022
```

////////////////////////////////////

////////////////////////////////////

Örnek setlocale

```c
int main(void)
{
    double dval;

    printf("bir gercek sayi girin: ");
    scanf("%lf", &dval);

    printf("dval = %f\n", dval);
    char* ploc = setlocale(LC_ALL, "turkish");
    if (!ploc) {
        printf("locale degisikligi yapilamadi");
        return 1;
    }
    printf("locale : %s\n", ploc);
}
```

```
Microsoft Visual Studio Debug Console
bir gercek sayi girin: 45,123
dval = 45.000000
locale : Turkish_Turkey.1254

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 21424) exited with code 0.
Press any key to close this window
```

Burada virgülü tanımıyor çünkü . ondalık ayracı

```c
int main(void)
{
    double dval;

    char* ploc = setlocale(LC_ALL, "turkish");
    if (!ploc) {
        printf("locale degisikligi yapilamadi");
        return 1;
    }
    printf("locale : %s\n", ploc);
    printf("bir gercek sayi girin: ");
    scanf("%lf", &dval);

    printf("dval = %f\n", dval);
}
```

```
Microsoft Visual Studio Debug Console
locale : Turkish_Turkey.1254
bir gercek sayi girin: 34,912
dval = 34,000000
```

Suanda noktayı tanımadı artık Türkçeye göre yazıyor.

Önemli bir özellik local çok etkili

Burada lc_all yazdığım için dilin bütün özellikleri değişti ama bunu belli özellikler için yapabilirdim sadece yazılar sadece harfler gibi ilk parametre bu işe yarıyor.

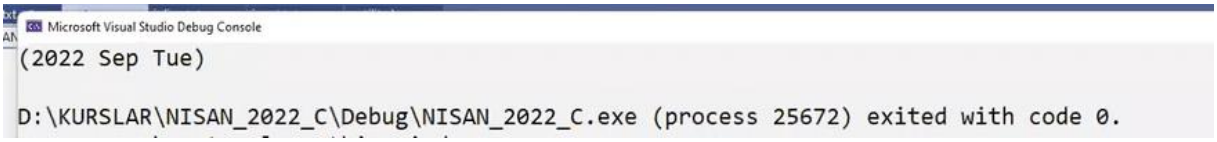
//

Örnek

```
//size_t strftime(char *pbuf, size_t n, const char *pfm, const struct tm*)
```

Strftime fonksiyonu bildirimi

```
#define         SIZE        100

int main(void)
{
    char str[SIZE];

    time_t timer;
    time(&timer);
    strftime(str, SIZE, "%y %b %a", localtime(&timer));

    printf("(%s)\n", str);
}
```



Şimdi Türkçe alalım

Locale değiştirilim

```c
int main(void)
{
    char str[SIZE];

    char *p = setlocale(_Category:LC_TIME, _Locale:"turkish");
    if (!p) {
        printf(_Format:"locale degistirilemedi\n");
        return 1;
    }

    time_t timer;
    time(_Time:&timer);
    strftime(_Buffer:str, _SizeInBytes:SIZE, _Format:"%Y %b %A", _Tm:localtime(_Time:&timer));

    printf(_Format:"(%s)\n", str);
}
```

Locale Türkçe oldu bu yüzden yürkçe yazdı ama sadece time değiştiği için ı harfini yazdırmadı.

```
clude <stdio.h>
Microsoft Visual Studio Debug Console
(2022 Eyl Sal²)

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 536) exited with code 0.
```

Örnek mk time

```c
#define         SIZE        100


int main(void)
{
    struct tm x;

    x.tm_year = 1987 - 1900;
    x.tm_mon = 5 - 1;
    x.tm_mday = 21;
    x.tm_hour = 0;
    x.tm_min = 0;
    x.tm_sec = 1;

    static const char* const pdays[] = {
    "Pazar",
    "Pazartesi",
    "Sali",
    "Carsamba",
    "Persembe",
    "Cuma",
    "Cumartesi",
    };

    time_t t = mktime(&x);
    ///

    puts(pdays[x.tm_wday]);
```

Timer değişkeni locale göndermeden kendimiz yazdık.

Kalanları kendi oluşturuyor.

```c
#define         SIZE        100


int main(void)
{
    time_t timer;
    time(&timer);

    struct tm* p = localtime(&timer);

    p->tm_hour -= 2345;

    timer = mktime(p);
```

Yapının belli elemanlarını set edip geri kalanı mktime fonksiyonu oluşturması içinde kullanabiliriz yada direk olarak time_t türüne saniye türüne çevirebiliriz bize kalmış.

//

Clock fonksiyonu

Örnek clock

```c
#include <time.h>
#include <stdio.h>
#include "nutility.h"

#define         SIZE        100000


int main(void)
{
    int a[SIZE];

    randomize();
    set_array_random(a, SIZE);
    sort_array(a, SIZE);
    clock_t x = clock();

    printf("%ld\n", x);
}
```

Çıktısı 84

Main başlayıp clock çağırılana kadar gecen süreyi hesaplıyor ama clock tick dediğimiz bir birim veriyor.

Bunu saniyeye çevirmek için bir makro kullanıyoruz.

```
#define         SIZE        100000


int main(void)
{
    int a[SIZE];

    randomize();
    set_array_random(a, SIZE);
    sort_array(a, SIZE);
    clock_t x = clock();

    printf("%f\n", (double)x / CLOCKS_PER_SEC);
}
```

FREKANS MUHABBETİ DİYE DÜŞÜNÜYORUM.

Time içindeki hazır bir makroya böldük.

İdiomatik clock kullanımı

```
int main(void)
{
    int a[SIZE];

    randomize();
    clock_t start = clock();
    set_array_random(a, SIZE);
    sort_array(a, SIZE);
    clock_t end = clock();
    printf("%f saniye\n", (double)(end - start) / CLOCKS_PER_SEC);
}
```

İki tick birbirinden çıkartıp frekansa böldük

```
Microsoft Visual Studio Debug Console
0.065000 saniye
```

//////////////////////////

Örnek buble sort ile q sort karşılaştırılmasını süre göstererek yapalım

```c
int main(void)
{
    size_t n;

    prinft("kac elemanli dizi: ");
    scanf("%zu", &n);

    int* pd = (int*)malloc(n * sizeof(int));
    if (!pd) {
        printf("bellek yetersiz\n");
        return 1;
    }

    for (size_t i = 0; i < n; ++i) {
        pd[i] = rand();
    }

    clock_t start = clock();
    bsort(pd, n);
    clock_t end = clock();

    printf("siralama bitti..  %f saniye\n", (double)(end - start) / CLOCKS_PER_SEC);
    _getch();
    print_array(pd, n);
    free(pd);
    ///
```

Tam 22 saniye sürdü

Eğer qsort çağırsaydık ne olurdu

```c
clock_t start = clock();
sort_array(pd, size:n);
clock_t end = clock();
```

Qsort ile yapınca 0.021 saniye sürdü nerdeyse 1000 kat daha kısa.

//

Oop tarzında kendi kutuphamizi yazalım.

Tarihi set eden fonksiyonlarımız.

```
//setter - mutator - set_functions

Date* set_date(Date*p, int day, int mon, int year);
Date* set_today(Date*p);
Date* set_date_from_str(Date* p, const char* pstr);
Date* set_date_time_t(Date* p, time_t sec);
Date* set_date_random(Date* p);
Date* set_year(Date* p, int year);
Date* set_month(Date* p, int mon);
Date* set_month_day(Date* p, int mday);
```

```
int  get_y
```

```
//input-output functions
void print_date(const Date* p);
```

Set test fonksiyonları.

```
int main(void)
{
    Date d1;

    set_date(&d1, 3, 5, 1992);
    print_date(&d1);

    Date d2;
    set_today(&d2);
    print_date(&d2);

    set_date_from_str(&d2, "24-02-2012");
    print_date(&d2);

    set_year(&d1, 2017);
    set_month(&d2, 1);
    set_month_day(&d1, 25);
    print_date(&d1);
    print_date(&d2);
}
```

Şimdi get fonksiyonlarını yazalım

```
//getter - accessor
int  get_year(const Date* p);
int  get_month(const Date* p);
int  get_month_day(const Date* p);
int  get_weekday(const Date* p); //0 pazar, 1 pazartesi 2 salı....
int  get_year_day(const Date* p); //
```

Get test fonksiyonlarımız.

```
#include "date.h"
#include "nutility.h"


int main(void)
{
    randomize();

    for (int i = 0; i < 10; ++i) {
        Date date;
        set_date_random(&date);
        print_date(&date);
        printf("yil          : %d\n", get_year(&date));
        printf("ay           : %d\n", get_month(&date));
        printf("ayin gunu    : %d\n", get_month_day(&date));
        printf("haftanin gunu    : %d\n", get_week_day(&date));
        (void)getchar();

    }

}
```

Formatlı scan ve 2 tarih arasındaki farkı oluşturduk.

```
//formatted input-output functions
void print_date(const Date* p);
void scan_date(Date* p);

// utility functions
int date_diff(const Date* p1, const Date* p2);
```

2 tarih arasındaki farkı bulan test kodu

```c
int main(void)
{
    Date birth_date;

    printf("doğrum tarihinizi girin: ");
    scan_date(&birth_date);
    print_date(&birth_date);
    Date todays_date;
    set_today(&todays_date);
    print_date(& todays_date);

    int n = date_diff(&todays_date, &birth_date);

    printf("Bugun su fani hayatinizin %d gunu\n", n);
```

Karşılaştırma fonksiyonu

```c
// utility functions
int date_diff(const Date* p1, const Date* p2);
int cmp_date(const Date*, const Date*);
```

Tarihleri sıralayan test fonksiyonu

```c
void set_date_array_random(Date* p, size_t size)
{
    while(size--) {
        set_date_random(p++);
    }
}

void print_date_array(const Date* p, size_t size)
{
    while (size--) {
        print_date(p++);
    }
}


int main(void)
{
    size_t n;

    printf("kac tarih: ");
    scanf("%zu", &n);

    Date* pd = (Date*)malloc(n * sizeof(Date));
    if (pd == NULL) {
        fprintf(stderr, "bellek yetersiz\n");
        return 1;
    }


    set_date_array_random(pd, n);
    printf("siralama basladi\n");
    clock_t start = clock();
    sort_date_array(pd, n);
    clock_t end = clock();
    printf("siralama bitti %f saniye\n", (double)(end - start) / CLOCKS_PER_SEC);
    _getch();
    print_date_array(pd, n);

    free(pd);
```

N gün kadar sonraki tarihi hesaplayan kod bildirimi

```c
int cmp_date(const Date*, const Date*);
Date* ndays_date(const Date* p, Date* pdest, int n);
```

Suanki günden kaç gün sonrasını istiyorsak o kadar sonraki tarihi hesaplayan test kodu.

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


int main(void)
{
    Date today;

    set_today(p:&today);

    printf(_Format:"bugun:  ");
    print_date(p:&today);
    Date future_date;

    for (int i = 1; i <= 1'000'000'000; i *= 10) {
        printf(_Format:"%d fun sonraki tarih\n", i);
        ndays_date(p:&today, pdest:&future_date, n:i);
        print_date(p:&future_date);
    }

}


//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
```

.h kütüphanesinin son hali

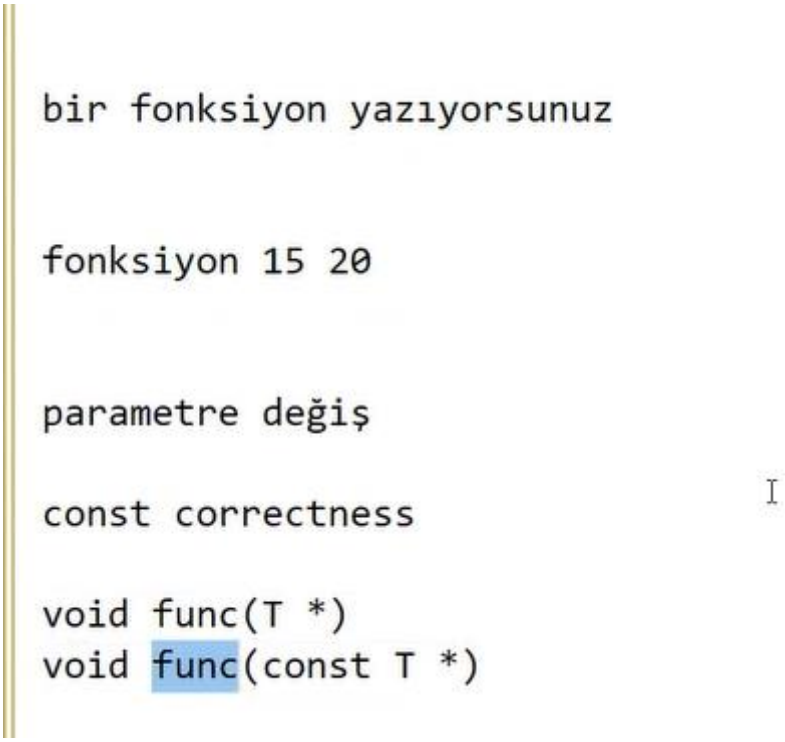
```h
#ifndef DATE_H
#define DATE_H


#include <time.h>

typedef struct {
    void* vptr;
}Date;


//setter - mutator - set_functions

Date* set_date(Date*p, int day, int mon, int year);
Date* set_today(Date*p);
Date* set_date_from_str(Date* p, const char* pstr);
Date* set_date_time_t(Date* p, time_t sec);
Date* set_date_random(Date* p);
Date* set_year(Date* p, int year);
Date* set_month(Date* p, int mon);
Date* set_month_day(Date* p, int mday);

//getter - accessor
int  get_year(const Date* p);
int  get_month(const Date* p);
int  get_month_day(const Date* p);
int  get_weekday(const Date* p); //0 pazar, 1 pazartesi 2 salı....
int  get_year_day(const Date* p); //


//formatted input-output functions
void print_date(const Date* p);
void scan_date(Date* p);

// utility functions
int date_diff(const Date* p1, const Date* p2);
int cmp_date(const Date*, const Date*);
Date* ndays_date(const Date* p, Date* pdest, int n);

#endif
```

```h
typedef struct {
    int md;
    int mm;
    int my;
}Date;
```

Yapımızın elemanlarını oluşturduk .h dosyasında

Ortak kodu mutlaka bir yere toplayın.

Kod tekrarı yapmayınız kodun kalitesi çok önemlidir lütfen yapma.



fonksiyon 15 20 satırı geçmemeli

parametre değişkeni sayısı çok fazla olmamalı.

Const doğruluğu olmalı.

Bunları neden anlattım set fonksiyonlarında ortak kod olma olasılığı çok yüksek.

Kod tekrarını engellemek için en başta kodun geçerli olup olmadığını bir fonksiyonla sorgulayacağız

Çünkü her seferinde bu bilgiye ihtiyaç duyacağız.

```c
#include "date.h"


#define  PRIVATE            static
#define  PUBLIC
#define  YEARBASE           1900

#define  isleap(y)          ((y) % 4 == 0 && ((y) % 100 != 0 || (y) % 400 == 0))
#define  mdays(m, y)        (daytabs[isleap(y)][m])

PRIVATE const int daytabs[][13] = {
    {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
    {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
};



PRIVATE int is_valid_date(int d, int m, int y);



//private functions

PRIVATE int is_valid_date(int d, int m, int y)
{
    return y >= YEARBASE &&
        m >= 0 && m <= 12 &&
        d > 0 && d <= mdays(m, y);
}
```