1 time.h

Tarih ve zaman ile ilgili işleri yapmak istediğimizde, standart C kütüphanesinden `time.h` başlık fonksiyonu çağırmalıyız. time.h başlık fonksiyonu, tarih ve zamanla ilgili işleri yapan fonksiyonları içerir (bkz. time.h fonksiyonları). Bu fonksiyonlardan bazılarını ele alacağız.

1.1 UNIX Zamanı

Belirli bir dönemin başladığı zamana *epoch time* denilir. UNIX işletim sistemlerinde zaman başlangıcı 1 Ocak 1970 günü saat 00:00 'dan başlar. O zamandan başlayarak şimdiki ana kadar geçen saniye sayısına *epoch zamanı* ya da *UNIX epoch zamanı* denilir. C dilinde

```
time_t
```

UNIX epoch zamanı'nı gösteren veri tipidir. Bu zaman başlangıcı yalnız UNIX işletim sistemlerinde değil, başka bilgisayar işletim sistemlerinde de çoğunlukla kullanılır.

Şu andaki *epoch zamanı*'nı bulmak için Program 1.1 çalıştırılabilir.

Program 1.1.

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

int main ()
{
    time_t saniye;
    saniye = time (NULL);
    printf ("Ocak 1, 1970 tarihinden beri geçen saniye sayısı %ld : \
     n", saniye);
```

```
    printf ("Ocak 1, 1970 tarihinden beri geçen gun sayısı    %ld : \
     n", saniye/3600);

    return 0;
}
```

```
/**
Ocak 1, 1970 tarihinden beri geçen saniye sayısı : 1424542538
Ocak 1, 1970 tarihinden beri geçen gün    sayısı : 395706
*/
```

Bu program her koşturulduğunda, burada yazılı olan zaman yerine, programın koşturulduğu zaman çıkacaktır.

Epoch zamanı, *saniye* cinsinden olduğu için, onun bildirdiği zamanı algılamamız çok zordur. O nedenle, epoch zamanını insanların anlayacağı tarih ve zaman biçimindeki bir string'e dönüştürmek gerekir. C dilinde bu işi yapan `ctime()` fonksiyonudur. `ctime()` fonksiyonu saniye cinsinden olan epoch zamanını *haftanın_hangi_günü, ay_adı, ayın_kaçıncı günü, saat:dakika:saniye, yıl* olarak yazar. Progam 1.2 ctime() fonksiyonunun kullanılışını gösteriyor.

Program 1.2.

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

int main(void)
{
  time_t simdiki_zaman;
  simdiki_zaman = time(NULL);
  printf(ctime(&simdiki_zaman));

  return 0;
}
```

```
/**
  Sat Feb 21 19:51:46 2015
*/
```

Bu program her koşturulduğunda, o andaki tarih ve zaman çıkacaktır.

`ctime()` fonksiyonunun çıktısı

```
Hhh Ayy gg hh:mm:ss yyyy
```

biçemindedir (format). Bu biçemdeki kısaltmaların açılımları Liste 1.3'deki gibidir.

Liste 1.1.

```
Hhh = Haftanın hangi günü. Üç harflik kısaltma, ilk harf büyük.
Ayy = Hangi ay; üç harflik kısaltma, ilk harf büyük.
gg = Ayın kaçıncı günü; iki haneli tamsayı.
hh:mm:ss = saat:dakika:saniye; ikişer haneli tamsayılar.
yyyy = yıl; dört haneli tamsayı.
```

1.2 difftime() fonksiyonu

difftime() fonksiyonu, bilgisayarda bir işin başladığı an ile bittiği an arasında geçen zamanı saniye cinsinden verir.

Liste 1.2.

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

int main(void)
{
  time_t basla,son;
  volatile long unsigned counter;

  basla = time(NULL);

  for(counter = 0; counter < 500000000; counter++)
    ; /* Hiç bir şey yapma */

  son = time(NULL);
  printf("Loop için gecen zaman : %f saniyedir. \n", difftime(son,
    basla));
  return 0;
}
```

```
/**
    Loop icin gecen zaman : 2.000000 saniyedir.
```

1.3 Zamanı Ölçmek

Fizik ve astronomi'nin önemli problemlerinden birisi zamanı doğru ölçmektir. Tarih boyunca, ayın dünya çevresindeki dönmesi ile dünyanın güneş etrafındaki dönmesini temel alan takvimler zaman ölçen araçlar olarak kullanılmıştır. Bunlara G*üneş Zamanı* diyoruz. Ancak gezegenlerin bir tam dönüşleri sabit olmadığından, onların hareketlerine göre tanımlanan zaman ölçümleri bilimin istediği duyarlığa sahip değildir.

Ayrıca, farklı kültürlerde tarih ve zaman yazma biçemleri farklıdır.

GMT

GMT (Greenwich Mean Time), başlangıç meridyeni üzerinde ortalama güneş zamanıdır.

Bir yerin yerel zamanı (ortalama güneş zamanı) yerküre üzerindeki zaman dilimine (boylam, meridien) göre ölçülür. Dünyada ulaşım, haberleşme ve ticaret yaygınlaştıkça, yerküre üzerinde ortak bir zaman tanımlama zorunluğu doğmuştur. Sosyal yaşamı en çok etkileyen gece-gündüz zamanına dayalı bir sistem geliştirilmiştir. Bu sistemde, Londra'nın güneydoğusunda yer alan Greenwich kasabasındaki gözlemevinden (rasathane) geçen boylam başlangıç kabul edilir (0^o). Bu boylamın öğle zamanı günün saat başlangıcı kabul edilir (00.00:00). Adına GMT (Greenwich Mean Time) deniyor. Greenwich'ten batıya doğru yerküre üzerindeki boylamlar 24 eşit parçaya bölünmüş, ardışık iki boylam arası 1 saat olarak kabul edilmiştir. İki boylam arası 15^o dir (15 derece). 12 saat tabanlı zaman ölçümlerinde Greenwich'e göre doğu yönündeki saatler GMT zamanına 1 den 12 ye kadar, batı yönünde ise -1 den -12 ye kadar saat eklenerek bulunur. Örneğin, Türkiye Greenwich meridyenine göre 2 meridyen doğudadır. Dolayısyla GMT zamanına göre 2 saat ileridedir. Türkiyedeki zamanı bulmak için GMT zamanından 2 saat çıkarılır.

Doğal olarak, bir ülkenin sınırları birden çok zaman dilimi içinde kalıyor. O durumda, her ülke kendi zaman dilimini seçiyor.

1925'te 697 sayılı *Günün Yirmi Dört Saate Taksimine Dair Kanun* ile birlikte, saat diliminde de uluslararası bir standart haline gelmiş olan GMT sistemi kabul edilmiştir.

UTC

Dünyanın kendi çevresindeki ve güneş çevresindeki tam dönüşleri için geçen zamanlar sabit değildir. Dolayısıyla, ortama güneş zamanı (GMT) bilimin istediği duyarlığa sahip değildir. Daha doğru bir zaman ölçüsü olarak, *International Telecommunications Union*, atom saatine dayalı olarak hesaplanan UTC (Eşgüdümlü evrensel Zaman, Coordinated Universal Time) zaman ölçüsünü tanımladı. UTC 1963 yılında kullanılmaya başlandı. Güneş zamanına göre hesaplanan GMT zamanından farklıdır.

Liste 1.3.

```
UTC       :  20:00:00
İstanbul  :  22:00
Londra    :  20:00
```

```
Paris     :  21:00
Mekke     :  23:00
Beiijing  :  04:00
Tokyo     :  05:00
NewYork   :  15:00
```

1.4 gmtime() fonksiyonu

gmtime() fonksiyonu, UTC zamanının başka bir zaman dilimine dönüştürür.

Liste 1.4.

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

#define PST (-8)  // PST Pasific Standart Time = GMT -8
#define CET (1)   // CST  Central Standart Time = GMT + 1
#define EET (2)   // Eastern European Time = GMT + 2

   int main ()
        {
           time_t raw_time;
           struct tm *ptr_ts;

           time ( &raw_time );
           ptr_ts = gmtime ( &raw_time );

           printf ("Los Angeles saati : %2d:%02d\n",
               ptr_ts->tm_hour+PST, ptr_ts->tm_min);

           printf ("Amsterdam saati: %2d:%02d\n",
                ptr_ts->tm_hour+CET, ptr_ts->tm_min);

           printf ("Ankara saati: %2d:%02d\n",
                ptr_ts->tm_hour+EET, ptr_ts->tm_min);

               return 0;
         }
```

```
/**
  Los Angeles saati : 13:50
  Amsterdam saati: 22:50
  Ankara saati: 23:50
```