Bilgisayar Bilimlerine Giriş-II

-11-

BIL 1002

Dokuz Eylül Üniversitesi, Fen Fakültesi,
Bilgisayar Bilimleri Bölümü

Üst düzey dosya yapısı - tekrar.

- C programlama dilinde, disk dosyasına erişme (okuma veya yazma için) iki farklı yöntemle yapılır.
- Üst düzey G/Ç (high level I/O) ya da tamponlanmış (buffered I/O) G/Ç olarak;
- Alt düzey G/Ç (low level (I/O) ya da UNIX benzeri.
- Üst düzey G/Ç yönteminde okuma ve yazma işlemi, temelde, karakter düzeyinde yapılır ve kullanımı alt düzey yöntemine göre daha kolaydır. Programcının tampon bellek tanımlaması gerekmez, veri dönüşümü ve formatlı okuma yazma için programcının ayrıca ek bir şeyler yazmasına da gerek yoktur.

Üst düzey dosya yapısı.

- Bu yöntemde programcıya kolaylık sağlayan ve her biri değişik uygulamaya yönelik bir çok kütübhane fonksiyonu kullanılır. Bunlardan bazıları:
- fopen() Dosya oluşturmak ve açmak için,
- fclose() Dosyayı kapatmak için,
- putc() Dosyaya karakter yazmak için,
- getc() Dosyadan karakter okumak için,
- feof() Dosya sonuna gelindiğini sorgulamak için,
- fprintf() Dosyaya formatlı yazmak için,
- fscanf() Dosyaya formatlı okumak için,
- fputs() Dosyaya katar yazmak için,
- fgets() Dosyadan katar okumak için,
- fwrite Dosyaya diziyi yazmak için,
- fread() Dosyadan diziyi okumak için,
- fseek() Verilere rastgele erişim için.

Dosyanın Açılması ve Kapatılması

Önceki sunumumuzda işlediğimiz gibi,

- Bir dosyaya okuma ve yazma yapmak (erişmek) için onun açılması gerekir. Bu işlem için ***fopen()*** fonksiyonu kullanılır.
  - ……
  - FILE *di;
  - ……..
  - di=fopen(dosya adı, modu);

fopen() fonksiyonuyla açılan bir dosya, gerekli erişimler yapıldıktan sonra, kapatılır.

Dosyanın Açılması ve Kapatılması

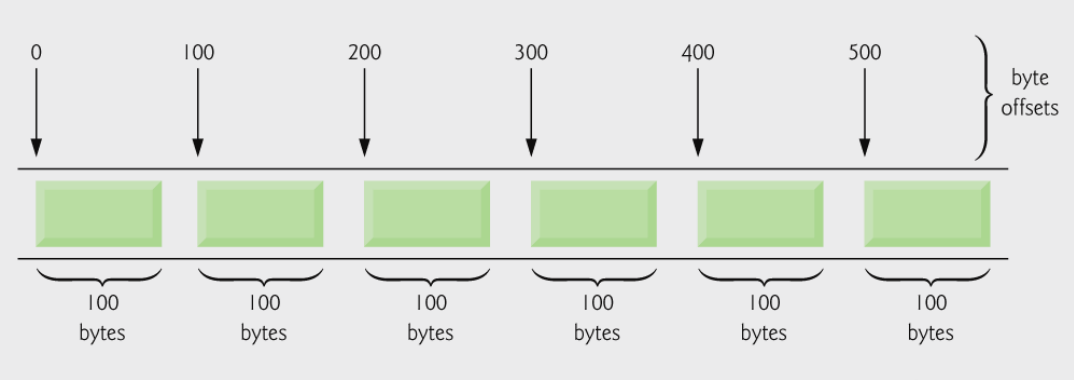
Dosyanın kapatılmasıyla; yazma ve ekleme modunda açılmışsa, ve tampon bellekte henüz disk'e yazılmamaış veri varsa, disk'e yazılır; ardından o dosya için ayrılmış geçici bellek alanı (tampon bellek…) serbest bırakılır. Herhangi bir modda açılmış bir dosya ***fclose()*** fonksiyonu kullanılarak kapatılır.

- ……
- (dosya üzerine işlemler yapan program parçası)
- ….
- fclose(di);
- …….

Dosyayı kapatmak için dosya işaretçisi, ***di***, yeterlidir. Dosya hatasız kapatılmışsa, ***fclose()*** fonksiyonu **0** sayısal değerini gönderir.

Rasgele Erişimli Dosyalar

- **Rasgele erişimli dosyalar**
  - Tüm kayıtlar sabit uzunluğa sahiptir ve diğer kayıtların aranmasına gerek kalmadan doğrudan erişilebilir.
  - Her kayıt sabit uzunlukta olduğu için kaydın dosya başlangıcına göre konumu anahtar kaydın bir fonksiyonu olarak bulunabilir.
  - Dosyadaki diğer verilere zarar vermeden yeni veriler eklenebilir.
  - Daha önceden depolanmış veriler tüm dosyanın yeniden yazılmasına gerek kalmadan güncellenebilir, silinebilir.



Rasgele Erişimli Dosyalar

- C'de rastgele erişimli dosyalar için ayrı bir dosya açma kipi tanımlanmamıştır, fakat metin türü değil ***ikili(binary) tür*** dosyalar Rastgele Erişim için tercih edilir.
- İkili dosyalar üzerinde işlem yapmak genelde daha hızlıdır. Aynı verileri tek tek yazdırmak yerine, toplu şekilde yazabilir.
- İkili dosyalar, genellikle veri iyi organize edildiğinde, metin dosyalarından daha az yer kaplar.

Mode	Açıklama
rb	İkili bir dosyayı okumak için aç
wb	Yazmak için ikili bir dosya yarat. Eğer dosya daha önceden varsa,o andaki içeriğini siler.
ab	Ekle;ikili bir dosyayı okumak için ya da dosyanın sonuna yazma yapmak için aç
rb+	İkili bir dosyayı güncellemek için aç (okuma ve yazma yapmak için)
wb+	Güncellemek için ikili bir dosya yarat. Eğer dosya daha önceden varsa,o andaki içeriğini sil.
ab+	Ekle; ikili bir dosyayı güncellemek için aç ya da yarat. Tüm yazma dosyanın sonuna yapılır.

Rasgele Erişimli Dosyalar Yaratmak

- Rasgele erişimli dosyalarda veriler
  - Formatsızdır (ham byte olarak)
    - Aynı veri tipindeki tüm veriler aynı miktar bellek kullanır.
    - Aynı tip tüm kayıtlar eşit ve sabit uzunluktadır.
    - Veriler insan okumasına uygun değildir.

Rasgele Erişimli Dosyalar Yaratmak

- Formatsız I/O (Girdi/Çıktı) fonksiyonları
  - **fwrite**
    - Hafızada belirlenmiş bir konumdan aldığı belli sayıdaki byte'ı dosyaya aktarır.
  - **fread**
    - Dosya içinde dosya pozisyon göstericisi ile belirlenen konumdan aldığı belli sayıdaki byte'ı belirlenen adresten başlayarak hafızaya aktarır.
  - Örneğin;

    fwrite(&number, sizeof(int), 1, myPtr);
    - &number – transfer edilecek byteların adresi
    - sizeof(int) – transfer edilecek byte sayısı
    - 1 – dizilerde transfer edilecek eleman sayısı
      - Bu durumda bir dizinin bir elemanı yazdırılıyormuş gibi düşünülebilir.)
    - myPtr – transferin gerçekleşeceği dosya göstericisi

İkili Dosya Üzerinde Okuma ve Yazma

- Bir ***ikili dosya*** üzerinde ***toplu/rastqele*** yazma için:

  fwrite(isaretci, boyut, adet, dosya_isaretcisi)

- Bir ikili dosya üzerinde toplu/rastgele okuma için:

  fread(isaretci, boyut, adet, dosya_isaretcisi)

  fonksiyonları kullanılır.

- Her iki fonksiyonda da ***isaretci*** parametresi okuma/yazma yapılan değişkenin adresini belirtir.
- ***boyut*** ilgili veri tipinin byte cinsinden boyutu, ***adet*** ise o tipte kaç tane veri kullanıldığıdır.
- ***fread*** fonksiyonu, hata oluşmadığı veya dosya sonuna gelinmediği sürece adet parametresiyle aynı değeri döndürür.

Rasgele Erişimli Dosyalar Yaratmak

- Yapıları (struct) yazma

fwrite(&myObject, sizeof (struct myStruct), 1, myPtr);

  - **sizeof** – parantez içindeki objenin byte olarak boyutu

- Dizinin birden çok elemanını yazma
  - İlk argüman dizi göstericisi
  - Üçüncü argüman dizi eleman sayısı

```c
#include <stdio.h>
struct musVer {
   int hesapNo;
   char soyad[15];
   char ad[10];
   double bakiye;
};
int main()
{
   int i;
   struct musVer bos = { 0, "", "", 0.0 };
   FILE *cfPtr;
   if ( ( cfPtr = fopen( "credit.dat", "wb" ) ) == NULL )
         printf( "Dosya acilamadi.\n" );
   else {
          for ( i = 1; i <= 100; i++ )
          fwrite( &bos, sizeof( struct musVer ), 1, cfPtr );
          fclose ( cfPtr );
   }
   return 0;
}
```

Rasgele Erişimli Dosyaya Rasgele Veri Yazmak

- ***fseek***
  - Dosya pozisyon göstericinin dosyada belirli bir konuma götürür.

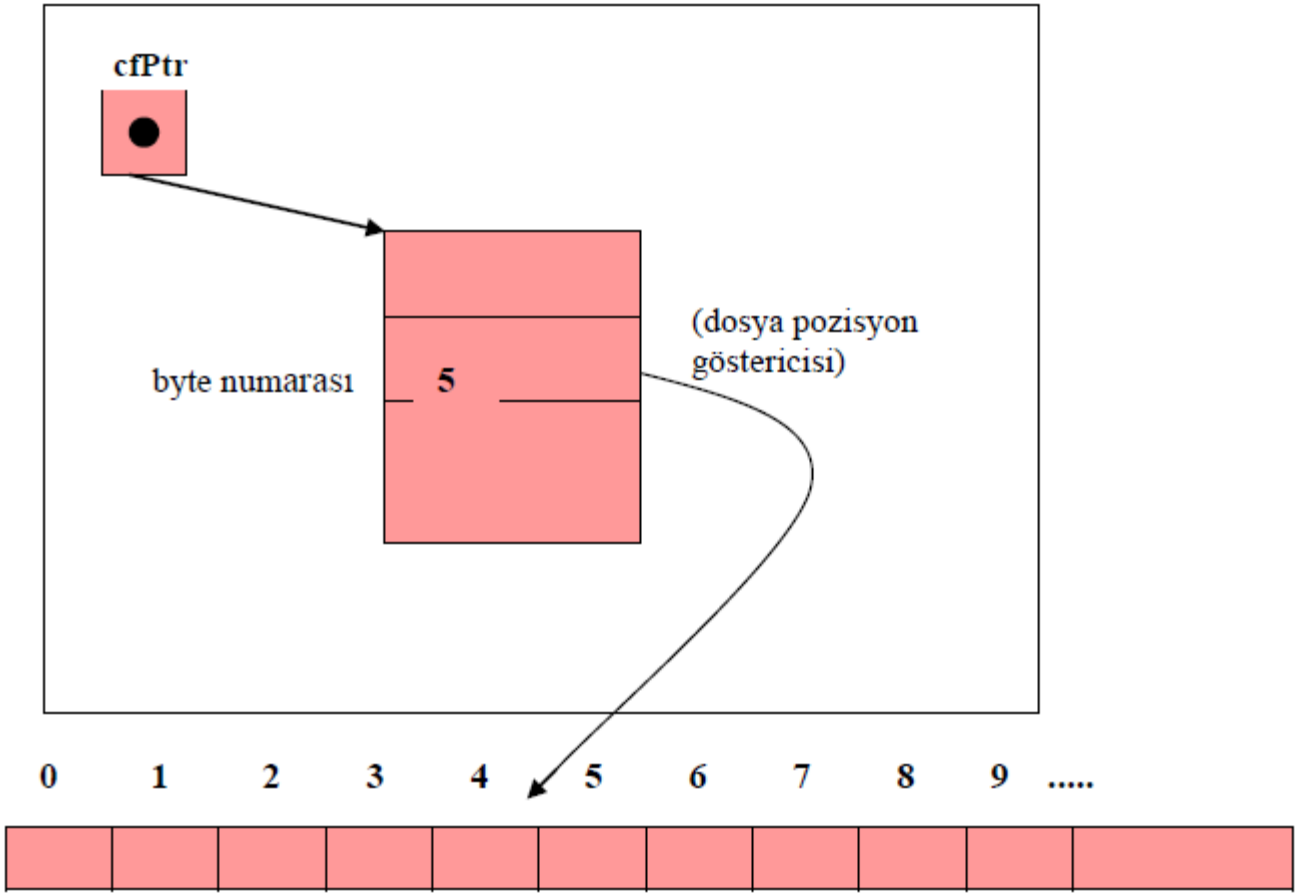
    fseek(pointer, offset, symbolic_constant);

    - ***pointer*** – dosya göstericisi
    - ***offset*** – dosya pozisyon göstericisi (başlangıçta sıfır)
    - ***symbolic_constant*** – Dosyada aramanın başlayacağı konumu belirten aşağıdaki değerlerden birini alabilir.
      - **SEEK_SET** – aramanın dosyanın başlangıcından başlayacağını
      - **SEEK_CUR** – aramanın dosyanın o anda bulunduğu konumdan başlayacağını
      - **SEEK_END** – aramanın dosyanın sonundan başlayacağını

```
#include <stdio.h>
struct musVer {
   int hesapNo;
   char soyad[ 15 ];
   char ad[ 10 ];
   double bakiye;
};
int main()
{
  int i;
   struct musVer mus = { 0, "", "", 0.0 };
   FILE *cfPtr;
   if ( ( cfPtr = fopen( "credit.dat", "rb+" ) ) == NULL )
      printf( "Dosya acilamadi.\n" );
   else {
      printf( "Hesap No:"
            " ( 1 -- 100, 0 sonlandirmak icin )\n? " );
      scanf( "%d", &mus.hesapNo );
      while ( mus.hesapNo != 0 ) {
            printf( "Soyad, ad, bakiye\n? " );
            fscanf( stdin, "%s%s%lf", mus.soyad,
                  mus.ad, &mus.bakiye );
            fseek( cfPtr, ( mus.hesapNo - 1 ) *
                  sizeof( struct musVer ), SEEK_SET );
            fwrite( &mus, sizeof( struct musVer ), 1,
                   cfPtr );
            printf( "Hesap no:\n? " );
            scanf( "%d", &mus.hesapNo );
         }
         fclose( cfPtr );
     }
   return 0;
}
```

```
Hesap No: ( 1 -- 100, 0 sonlandirmak icin )
? 45
Soyad, ad, bakiye
? tezel baris 450
Hesap no:
?
```

Dosyanın başlangıçtan itibaren 5 byte'lık bir yer değiştirmeyi belirten pozisyon göstericisi



Rasgele Erişimli Dosyadan Veri Okumak

- ***fread***
  - Bir dosyadan hafızaya belirli bir sayıda byte okur.

    fread(&mus, sizeof (struct musVer), 1, cfPtr);

  - Diziden sabit boyuttaki birden çok elemanı okuyabilir.
    - Dizinin göstericisi belirtilir.
    - Okunacak eleman sayısı belirtilir.
  - Üçüncü argüman okunacak eleman sayısı

```
#include <stdio.h>
#include <conio.h>
struct musVer {
   int hesapNo;
   char soyad[ 15 ];
   char ad[ 10 ];
   double bakiye;
};
int main()
{
   int i;
   struct musVer mus = { 0, "", "", 0.0 };
   FILE *cfPtr;
if ( ( cfPtr = fopen( "credit.dat", "rb" ) ) == NULL )
      printf( "Dosya acilamadi.\n" );
   else {
      printf( "%-6s%-16s%-11s%10s\n", "H.No", "Soyad",
            "Ad", "Bakiye" );
while (fread( &mus, sizeof( struct musVer ), 1,cfPtr )==1) {
                if ( mus.hesapNo != 0 )
                printf( "%-6d%-16s%-11s%10.2f\n",
                mus.hesapNo, mus.soyad,
                mus.ad, mus.bakiye );
      }
      fclose( cfPtr );
   }
   getch();
   return 0;
}
```

ÖRNEKLER

Soru 1

Dışarıdan 0 girilene kadar girilen belirsiz sayıda pozitif tamsayıyı teker teker bir ikili dosyaya yazdıktan sonra, hepsini tek seferde okuyarak bir diziye aktarıp dizi üzerinden ekrana yazdıran bir C programı yazın.

```c
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main() {
int s, *dizi, k = 0;
FILE *f;
if ((f = fopen("sayilar.bin", "w+")) == NULL)
{
    printf("Dosya Acilamadi!\n"); getchar();
    return 0;
}
do {
    scanf("%u", &s);
    if (s < 0) {
        printf("Negatif sayi girmeyin.\n");
        continue;
    }
  if (s == 0) break;
  fwrite(&s, sizeof(unsigned int), 1, f);
  k++;
}while (1);
  fseek(f, 0, SEEK_SET); // rewind(f);
  dizi = (int*)malloc(sizeof(int) * k);
  fread(dizi, sizeof( int), k, f);
  for (s = 0; s < k; s++)
    printf("%u\n", dizi[s]);
  fclose(f);
  getch();
  return 0;
}
```

Soru 2

- C dilinde bir ***ürün kayıt*** programı yazın.
- Ürün bilgileri,
  - numara,
  - ad ve
  - son kullanma tarihi

  şeklinde olmalıdır.
- Ürünler için bir topluluk değişkeni kullanılmalı ve ikili dosyada saklanmalıdır.
- Kullanıcının seçilmesi için ***yeni kayıt ekleme***, ***tüm kayıtları listeleme*** ve ***ürün sırasına göre tek bir kaydı gösterme*** özelliklerini sunmalıdır.

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

typedef struct {
int no;
char ad[10], skt[15];
} kayit;

int main() {
  FILE *f;
  kayit x;
  int sayi;
  char sec;
  anamenu:
    printf("\n\tURUN KAYIT PROGRAMI\n");
    printf("\nUrun kaydetmek icin\t\t(1)\n");
    printf("\nTum kayitlari okumak icin\t\t(2)\n");
    printf("\nBelirli bir urunu okumak icin\t(3)\n");
   sec = getche();
```

```
        URUN KAYIT PROGRAMI

Urun kaydetmek icin             (1)

Tum kayitlari okumak icin                (2)

Belirli bir urunu okumak icin   (3)
```

```
switch (sec) {
case '1':
     system("cls");
      if ((f = fopen("kayit.dat", "ab")) == NULL) {
          printf("Dosya Acilamadi!\n");
          break;
      }
do {
     printf("\nUrun adi: "); scanf("%s", x.ad);
     printf("Urun numarasi: "); scanf("%d", &x.no);
     printf("Urun SKT: "); scanf("%s", x.skt);
     fwrite(&x, sizeof(kayit), 1, f);
     printf("Kayda devam icin 'e' tusuna basin... ");
} while (getche() == 'e');
fclose(f);
break;
```

```
case '2':
    system("cls");
    if ((f = fopen("kayit.dat", "rb")) == NULL) {
        printf("Dosya Acilamadi!\n");
        break;
     }
sayi = 0;
while (fread(&x, sizeof(kayit), 1, f) == 1) {
      sayi++;
      printf("\n%d. Urun\n", sayi);
      printf("\nAdi: %s\n", x.ad);
      printf("Numarasi: %d\n", x.no);
      printf("SKT: %s\n", x.skt);
}
fclose(f);
break;
```

```
case '3': // ürün sırasına göre tek tek gösterme
    system("cls");
    if ((f = fopen("kayit.dat", "rb")) == NULL) {
    printf("Dosya Acilamadi!\n");
    break;
}
do {
    printf("\nKacinci urunu gormek istiyorsunuz?\n");
    scanf("%d", &sayi);
   fseek(f, sizeof(kayit) * (sayi - 1), SEEK_SET);
   if (fread(&x, sizeof(kayit), 1, f) == 1) {
     printf("\n%d. Urun\n", sayi);
     printf("\nAdi: %s\n", x.ad);
     printf("Numarasi: %d\n", x.no);
     printf("SKT: %s\n", x.skt);
   } else
       printf("Boyle bir urun yok urun yok!\n");
       printf("Urun aramaya etmek icin 'e' tusuna basin... ");
} while (getche() == 'e');
fclose(f);
break;
default:  printf("\n\nGecersiz bir secim yaptiniz!\n\n");
}
printf("\n\nAna menuye donmek icin 'e' tusuna"
"basin... ");
if (getche() == 'e') {
system("cls");
goto anamenu;
}
printf("\n\nPrograma kapatmak icin herhangi "
"bir tusa basin... ");
getch();
return 0;
}
```


SON