

Програмчлалын үндэс

Лекц № 10

Си хэл дээр файльтай ажиллах

А. Хүдэр

Агуулга

- ▶ Бүтцийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Бүтцийг функц руу утгаар нь болон хаягаар нь дамжуулах (typedef)
- ▶ Нэгдлийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Тоочих төрлийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Файлыг үүсгэх, унших, бичих, шинэчлэх
- ▶ Файл боловсруулах хоёр арга:
 - Дараалсан хандалт
 - Санамсаргүй хандалт

Оршил

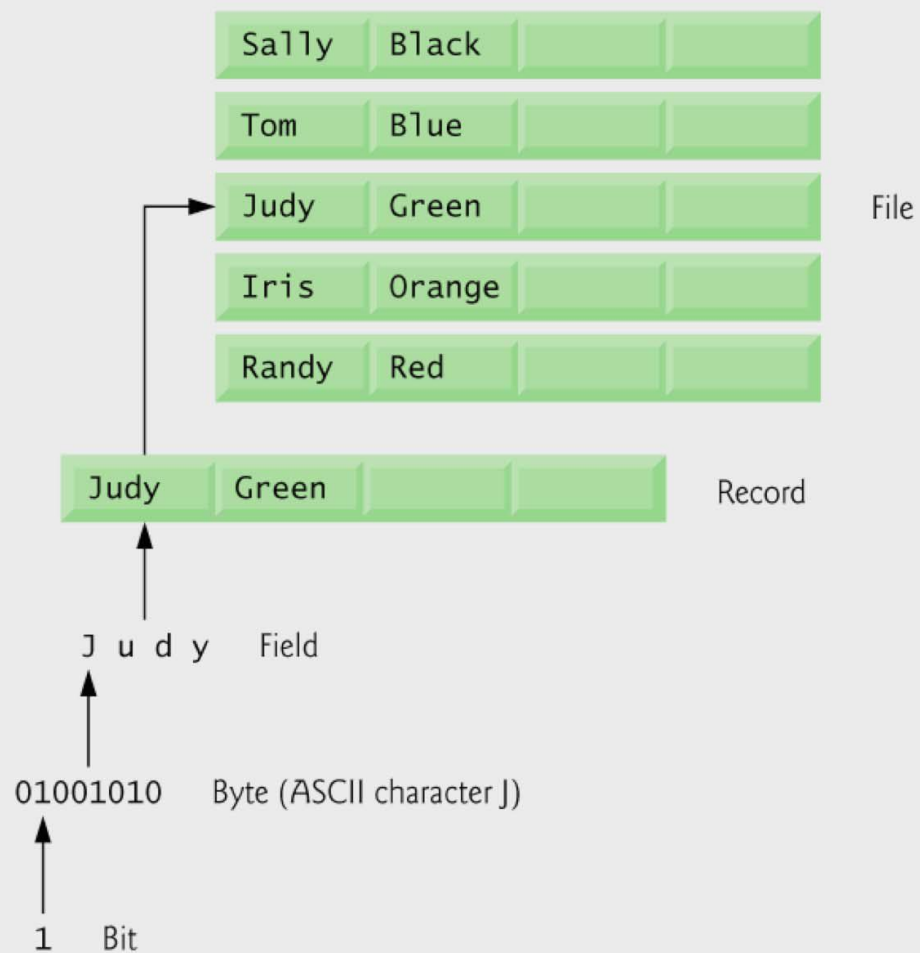
- ▶ Өгөгдлийн файлууд
 - Си програмаар үүсгэж, шинэчилж, боловсруулж болно
 - Их хэмжээний өгөгдлийг хадгалахад хэрэглэгдэнэ
 - Өгөгдлийг хувьсагч болон хүснэгтэнд хадгалахад түр зуур хадгалагддаг

Өгөгдлийн шатлал

► Өгөгдлийн шатлал:

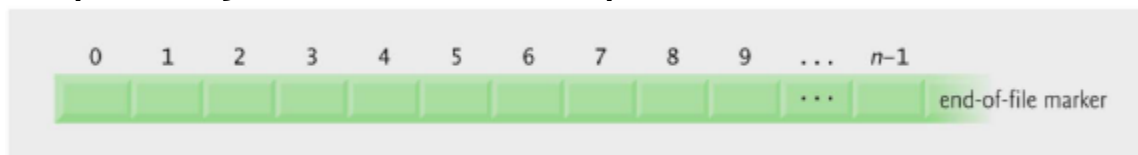
- Бит – хамгийн бага нэгж
 - 0 эсвэл 1 гэсэн утгатай байна
- Байт – 8 бит
 - Нэг тэмдэгт хадгалахад хэрэглэгдэнэ
 - Аравтын цифр, үсэг болон тусгай тэмдэгтүүд
- Талбар – утга бүхий тэмдэгтүүдийн бүлэг
 - Жишээ: хүний нэр
- Бичлэг – холбоотой талбаруудын бүлэг
 - struct – аар илэрхийлнэ
 - Жишээ: Цалингийн системд ямар нэг ажилчны тухай **бичлэг нь түүний дугаар, нэр, хаяг** гэх мэтийг агуулна.
- Файл – холбоотой бичлэгүүдийн бүлэг
 - Жишээ: цалингийн файл
- Өгөгдлийн сан – холбоотой файлуудын бүлэг

Өгөгдлийн шатлал



Файл ба урсгал

- ▶ Си нь файл бүрийг байтуудын урсгал хэлбэрээр авч үздэг
 - Файлын төгсгөлд файлын төгсгөлийн тэмдэгт байна
 - эсвэл файл тусгай тэмдэгтээр төгсөнө



- ▶ Файл нээгдэхэд түүнтэй харгалзах урсгал үүснэ
 - Файл болон програмын хоорондын харилцааг хангана
 - Файлыг нээхэд FILE төрлийн бүтцийн заагчийг буцаадаг
 - Файлын заагчийн жишээ:
- | Заагч | Файл |
|----------|--------------------------|
| • stdin | стандарт оролт (гар) |
| • stdout | стандарт гаралт (дэлгэц) |
| • stderr | стандарт алдаа (дэлгэц) |

Файл ба урсгал

- ▶ Стандарт сангийн унших/бичих функцууд
 - `fgetc`
 - Файлаас нэг тэмдэгт уншина
 - `FILE`-ын заагчийг аргументаар авна
 - `fgetc(stdin)` нь `getchar()`-тай адил
 - `fputc`
 - Файл руу нэг тэмдэгт бичнэ
 - `FILE`-ын заагч болон бичих тэмдэгтээ аргументаар авна
 - `fputc('a', stdout)` нь `putchar('a')` – тай адил
 - `fgets`
 - Файлаас нэг мөр уншина
 - `fputs`
 - Файл руу нэг мөр бичнэ
 - `fscanf/fprintf`
 - файл боловсруулах `scanf`, `printf`-тэй адил функцууд

Дараалсан хандалттай файл ҮҮСГЭХ

- `fprintf`
 - Файл руу хэвлэхэд хэрэглэгдэнэ
 - Эхний аргумент нь `FILE`-ын заагч (бичих гэж байгаа файлын заагч) байгаагаас бусад нь `printf`-тэй адил
- `feof(FILE-ын заагч)`
 - Файлын төгсгөлийн тэмдэгт (цаашид боловсруулах өгөгдөл байхгүй) таарсан үед үнэн утга буцаана
- `fclose(FILE-ын заагч)`
 - Заасан файлыг хаана
 - Програм дуусахад автоматаар хийгдэнэ
 - Файлаа хааж байх хэрэгтэй
- ▶ Бусад
 - Програм файлтай ажиллахгүй, нэг файлтай эсвэл олон файлтай ажиллаж байж болно
 - Файл бүр өөрийн ялгаатай нэртэй, өөрийн файлын заагчтай байна

Агуулга

- ▶ Бүтцийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Бүтцийг функц руу утгаар нь болон хаягаар нь дамжуулах (typedef)
- ▶ Нэгдлийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Тоочих төрлийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Файлыг үүсгэх, унших, бичих, шинэчлэх
- ▶ Файл боловсруулах хоёр арга:
 - Дараалсан хандалт
 - Санамсаргүй хандалт

Дараалсан хандалттай файл ҮҮСГЭХ

- ▶ Си хэлэнд ямар нэг файлын бүтэц заадаггүй
 - Файлд бичлэгийн тухай мэдээлэл агуулагдахгүй
 - Файлын бүтцийг програм зохиогч хангах ёстой
- ▶ Файл ҮҮСГЭХ
 - `FILE *cfPtr;`
 - `cfPtr` гэсэн нэртэй `FILE`-ын заагч зарлаж байна
 - `cfPtr = fopen("clients.dat", "w");`
 - `fopen` функц файл руу заах `FILE` төрлийн заагч буцаана
 - Хоёр аргумент авна – **нээх файл** ба **файл нээх горим**
 - Хэрэв нээж чадахгүй бол `NULL`-ыг буцаана

```

1  /* Example 91:
2     Create a sequential file */
3  #include <stdio.h>
4
5  int main( void )
6  {
7     int account;      /* account number */
8     char name[ 30 ]; /* account name */
9     double balance;   /* account balance */
10
11     FILE *cfPtr;      /* cfPtr = clients.dat file pointer */
12
13     /* fopen opens file. Exit program if unable to create file */
14     if ( ( cfPtr = fopen( "clients.dat", "w" ) ) == NULL ) {
15         printf( "File could not be opened\n" );
16     } /* end if */
17     else {
18         printf( "Enter the account, name, and balance.\n" );
19         printf( "Enter EOF to end input.\n" );
20         printf( "? " );
21         scanf( "%d%s%lf", &account, name, &balance );
22

```

FILE төрлийн заагчийг зарласнаар
файлын шинэ заагчийг тодорхойлно

fopen функц нь файлыг нээх ба w аргумент
нь уг файлыг бичихээр нээж байгааг заана

```

23  /* write account, name and balance into file with fprintf */
24  while ( !feof( stdin ) ) {
25      fprintf( cfPtr, "%d %s %.2f\n", account, name, balance );
26      printf( "? " );
27      scanf( "%d%s%f", &account, name, &balance );
28  } /* end while */
29
30  fclose( cfPtr ); /* fclose closes file */
31 } /* end else */
32
33 return 0; /* indicates successful termination */
34
35 } /* end main */

```

feof функц нь файлын төгсгөл хүрэхэд
үнэнг буцаана

fprintf функц нь тэмдэгт мөрийг файл руу бичнэ

fclose функц нь файлыг хаана

```

Enter the account, name, and balance.
Enter EOF to end input.
? 100 Jones 24.98
? 200 Doe 345.67
? 300 White 0.00
? 400 Stone -42.16
? 500 Rich 224.62
? ^Z

```

Зарим үйлдлийн систем дэх файлын төгсгөлийн тэмдэгт

Operating system	Key combination
Linux/Mac OS X/UNIX	<i><Ctrl> d</i>
Windows	<i><Ctrl> z</i>

Файл нээх горимууд

Mode	Description
r	Open an existing file for reading.
w	Create a file for writing. If the file already exists, discard the current contents.
a	Append; open or create a file for writing at the end of the file.
r+	Open an existing file for update (reading and writing).
w+	Create a file for update. If the file already exists, discard the current contents.
a+	Append: open or create a file for update; writing is done at the end of the file.
rb	Open an existing file for reading in binary mode.
wb	Create a file for writing in binary mode. If the file already exists, discard the current contents.
ab	Append; open or create a file for writing at the end of the file in binary mode.
rb+	Open an existing file for update (reading and writing) in binary mode.
wb+	Create a file for update in binary mode. If the file already exists, discard the current contents.
ab+	Append: open or create a file for update in binary mode; writing is done at the end of the file.

Дараалсан хандалттай файлаас өгөгдөл унших

- ▶ Дараалсан хандалттай файлаас унших
 - FILE-ын заагч зарлан түүнийг унших гэж байгаа файлтайгаа холбох
`cfPtr = fopen("clients.dat", "r");`
 - `fscanf` – ийг хэрэглэн файлаас унших
 - Эхний аргумент нь FILE-ын заагч гэдгээс бусдаараа `scanf`-тэй адил `fscanf(cfPtr, "%d%s%f", &account, name, &balance);`
 - Өгөгдөл эхлэлээс төгсгөл хүртэл уншигдана
 - Файлын байрлалын заагч
 - Уншигдах/бичигдэх гэж байгаа дараагийн байтын дугаарыг заана
 - Заагч биш харин бүхэл тоон утга юм (байтын байрлал заана)
 - **Байтын шилжилт** гэж бас нэрлэдэг
 - `rewind(cfPtr)`
 - Файлын байрлалын заагчийг файлын эхэнд авчирна (0-р байт)

```

1  /* Example 92:
2     Reading and printing a sequential file */
3  #include <stdio.h>
4
5  int main( void )
6  {
7     int account;      /* account number */
8     char name[ 30 ]; /* account name */
9     double balance;   /* account balance */
10
11     FILE *cfPtr;      /* cfPtr = clients.dat file pointer */
12
13     /* fopen opens file; exits program if file cannot be opened */
14     if ( ( cfPtr = fopen( "clients.dat", "r" ) ) == NULL ) {
15         printf( "File could not be opened\n" );
16     } /* end if */
17     else { /* read account, name and balance from file */
18         printf( "%-10s%-13s%\n", "Account", "Name", "Balance" );
19         fscanf( cfPtr, "%d%s%lf", &account, name, &balance );
20

```

fopen функц файлыг нээнэ; r аргумент нь уг файлыг уншихаар нээснийг заана


```

21     /* while not end of file */
22     while ( !feof( cfPtr ) ) {
23         printf( "%-10d%-13s%7.2f\n", account, name, balance );
24         fscanf( cfPtr, "%d%s%lf", &account, name, &balance );
25     } /* end while */
26
27     fclose( cfPtr ); /* fclose closes the file */
28 } /* end else */
29
30 return 0; /* indicates successful termination */
31
32 } /* end main */

```

fscanf функц файлаас тэмдэгт мөр уншина

Account	Name	Balance
100	Jones	24.98
200	Doe	345.67
300	White	0.00
400	Stone	-42.16
500	Rich	224.62

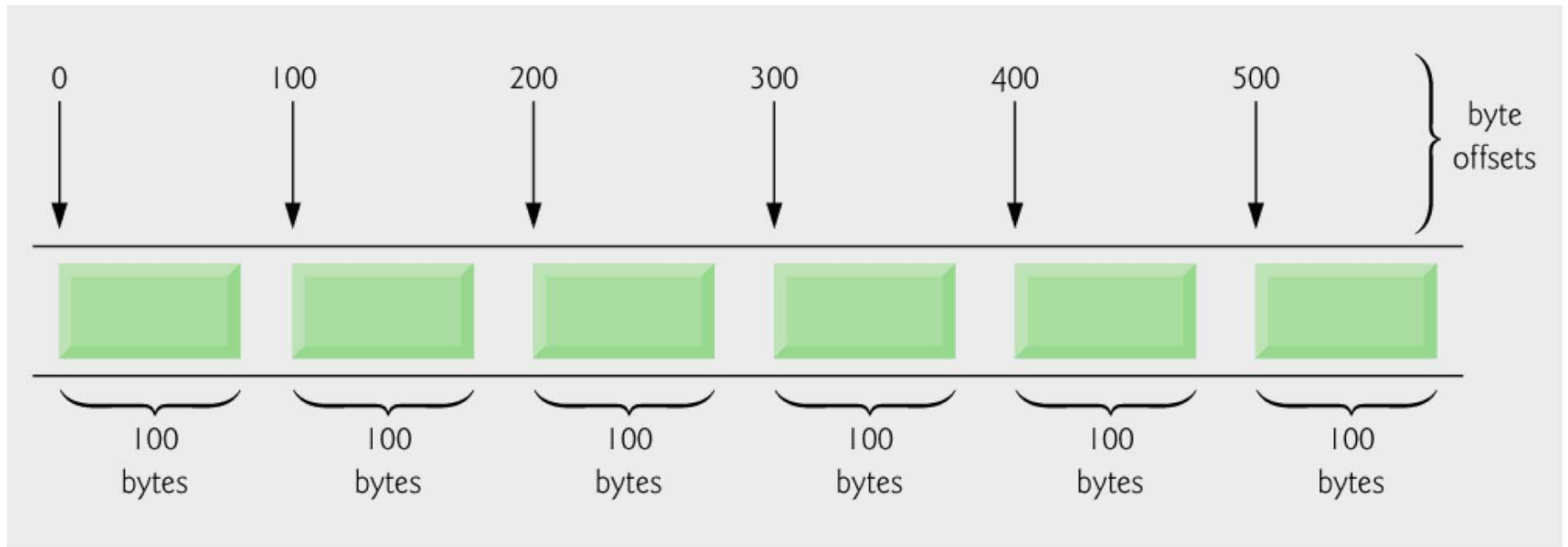
Агуулга

- ▶ Бүтцийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Бүтцийг функц руу утгаар нь болон хаягаар нь дамжуулах (typedef)
- ▶ Нэгдлийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Тоочих төрлийг үүсгэх, анхны утга олгох, хэрэглэх
- ▶ Файлыг үүсгэх, унших, бичих, шинэчлэх
- ▶ Файл боловсруулах хоёр арга:
 - Дараалсан хандалт
 - Санамсаргүй хандалт

Санамсаргүй хандалттай файл

- ▶ Санамсаргүй хандалттай файл
 - Бичлэг рүү хандахын тулд бусад бичлэгийг уншиж өнгөрөөх хэрэггүй
 - Файл дах бичлэг рүү шууд хандана
 - Өгөгдлийг бусад бичлэгийг устгалгүйгээр өөрчилж болно
 - Өмнө нь хадгалсан өгөгдлийг давтан бичилгүйгээр шинэчилж болно
- ▶ Тогтмол урттай бичлэг
 - Дараалсан хандалттай файлын бичлэгийн урт тогтмол байдаггүй

Си хэлэн дэх санамсаргүй хандалттай файлын дүрслэл



Санамсаргүй хандалттай файл ҮҮСГЭХ

- ▶ Санамсаргүй хандалттай файл дахь өгөгдөл
 - Хэлбэргүй (“цэвэр байт” байдлаар хадгалагдсан)
 - Өгөгдлүүд нь бүгд нэг төрлийнх (жишээ нь int) байх тул санах ойд ижил хэмжээний зай эзэлнэ
 - Бичлэгүүд тогтмол урттай байна. Иймд ямар нэг бичлэгийн байрлалыг файлын эхлэлээс эхлэн бичлэгийн түлхүүрийг ашиглан тооцоолж болно.
 - Хүнд уншигдахааргүй байна

Санамсаргүй хандалттай файл ҮҮСГЭХ

► Хэлбэржүүлэлтгүй Оролт/Гаралтын функцууд

- fwrite

- Санах ой дахь байрлалаас файл руу байтуудыг бичнэ

- fread

- Файлаас санах ой руу байтуудыг бичнэ

- Жишээ:

`fwrite(&number, sizeof(int), 1, myPtr);`

- &number – байтуудыг унших байрлал
- sizeof(int) – бичих байтын тоо
- 1 – хүснэгтийн хувьд бичих элементийн тоо
 - Энэ жишээнд хүснэгтийн нэг элементийг бичиж байна
- myPtr – бичих файл

Санамсаргүй хандалттай файл ҮҮСГЭХ

► Бичлэгийг бичих

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

- sizeof – хаалтанд байгаа объектын хэмжээг буцаана

► Хүснэгтийн хэд хэдэн элементийг бичих

- Эхний аргумент нь хүснэгтийн заагч
- Гурав дахь аргумент нь бичих элементийн тоо

```

1  /* Example 93:
2     Creating a random-access file sequentially */
3  #include <stdio.h>
4
5  /* clientData structure definition */
6  struct clientData {
7     int acctNum;          /* account number */
8     char lastName[ 15 ]; /* account last name */
9     char firstName[ 10 ]; /* account first name */
10    double balance;        /* account balance */
11 }; /* end structure clientData */
12
13 int main( void )
14 {
15     int i; /* counter used to count from 1-100 */
16
17     /* create clientData with default information */
18     struct clientData blankClient = { 0, "", "", 0.0 };
19

```



```

20 FILE *cfPtr; /* credit.dat file pointer */
21
22 /* fopen opens the file; exits if file cannot be opened */
23 if ( ( cfPtr = fopen( "credit.dat", "wb" ) ) == NULL ) {
24     printf( "File could not be opened.\n" );
25 } /* end if */
26 else {
27
28     /* output 100 blank records to file */
29     for ( i = 1; i <= 100; i++ ) {
30         fwrite( &blankClient, sizeof( struct clientData ), 1, cfPtr );
31     } /* end for */
32
33     fclose ( cfPtr ); /* fclose closes the file */
34 } /* end else */
35
36 return 0; /* indicates successful termination */
37
38 } /* end main */

```

fopen функц файлыг нээнэ; **wb** аргумент нь уг файлыг хоёртын горимд, бичихээр нээж байгааг заана

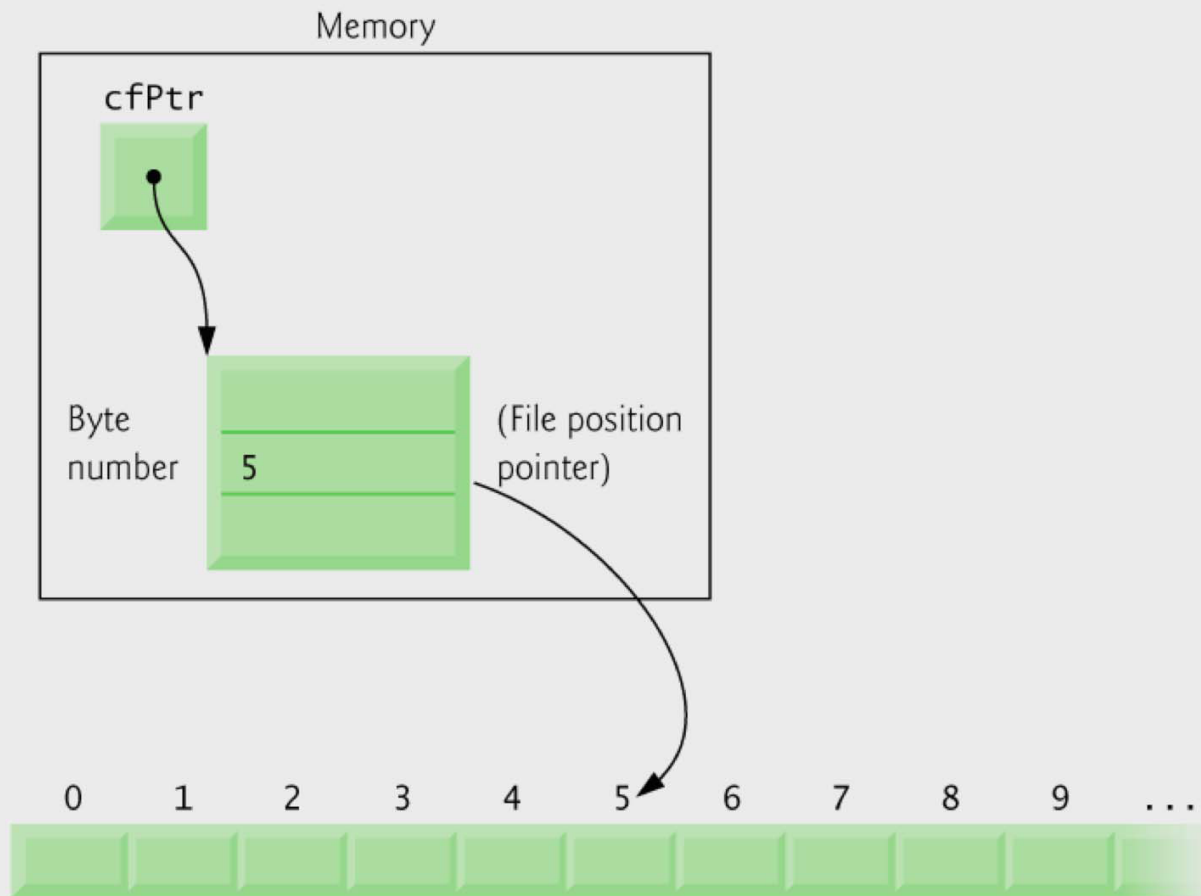
fwrite нь санамсаргүй хандалттай файл руу байтуудыг дамжуулна

Санамсаргүй хандалттай файлын дурын байрлал руу бичих

► fseek

- Байрлалын заагчийг заасан байрлалд аваачна
- `fseek(pointer, offset, symbolic_constant);`
 - `pointer` – файлын заагч
 - `offset` – файлын байрлалын заагч (эхний байрлал 0)
 - `symbolic_constant` (файлын хаанаас уншихыг заана)
 - `SEEK_SET` – хайлтыг файлын эхнээс эхэлнэ
 - `SEEK_CUR` – хайлтыг одоо байгаа байрлалаас эхэлнэ
 - `SEEK_END` – хайлтыг файлын төгсгөлөөс эхэлнэ
- `fseek` функц үйлдлээ гүйцэтгэж чадахгүй бол **тэг биш утга** буцаана

Файлын эхлэлээс 5 байтын шилжилттэй байгаа файлын байрлалын заагч



```

1  /* Example 94:
2     Writing to a random access file */
3  #include <stdio.h>
4
5  /* clientData structure definition */
6  struct clientData {
7     int acctNum;          /* account number */
8     char lastName[ 15 ]; /* account last name */
9     char firstName[ 10 ]; /* account first name */
10    double balance;       /* account balance */
11 }; /* end structure clientData */
12
13 int main( void )
14 {
15     FILE *cfPtr; /* credit.dat file pointer */
16
17     /* create clientData with default information */
18     struct clientData client = { 0, "", "", 0.0 };
19
20     /* fopen opens the file; exits if file cannot be opened */
21     if ( ( cfPtr = fopen( "credit.dat", "rb+" ) ) == NULL ) {
22         printf( "File could not be opened.\n" );
23     } /* end if */
24     else {
25
26         /* require user to specify account number */
27         printf( "Enter account number"
28             " ( 1 to 100, 0 to end input )\n? " );
29         scanf( "%d", &client.acctNum );
30

```

```

31  /* user enters information, which is copied into file */
32  while ( client.acctNum != 0 ) {
33
34      /* user enters last name, first name and balance */
35      printf( "Enter lastname, firstname, balance\n? " );
36
37      /* set record lastName, firstName and balance value */
38      fscanf( stdin, "%s%s%lf", client.lastName,
39              client.firstName, &client.balance );
40
41      /* seek position in file to user-specified record */
42      fseek( cfPtr, ( client.acctNum - 1 ) *
43              sizeof( struct clientData ), SEEK_SET );
44
45      /* write user-specified information in file */
46      fwrite( &client, sizeof( struct clientData ), 1, cfPtr );
47
48      /* enable user to input another account number */
49      printf( "Enter account number\n? " );
50      scanf( "%d", &client.acctNum );
51  } /* end while */
52
53      fclose( cfPtr ); /* fclose closes the file */
54  } /* end else */
55
56  return 0; /* indicates successful termination */
57
58 } /* end main */

```

fseek санамсаргүй хандалтат
файл доторх тодорхой
байрлалыг хайж олно

```
Enter account number ( 1 to 100, 0 to end input )
? 37
Enter lastname, firstname, balance
? Barker Doug 0.00
Enter account number
? 29
Enter lastname, firstname, balance
? Brown Nancy -24.54
Enter account number
? 96
Enter lastname, firstname, balance
? Stone Sam 34.98
Enter account number
? 88
Enter lastname, firstname, balance
? Smith Dave 258.34
Enter account number
? 33
Enter lastname, firstname, balance
? Dunn Stacey 314.33
Enter account number
? 0
```

Санамсаргүй хандалттай файлаас өгөгдөл унших

► fread

- файлаас заасан тооны байтыг санах ой руу уншина

`fread(&client, sizeof (struct clientData), 1, myPtr);`

- Тогтмол хэмжээтэй хэд хэдэн хүснэгтийн элементийг уншиж болно
 - Хүснэгтийн заагчийг өгнө
 - Унших элементийн тоог өгнө
- Олон элемент уншихын тулд гурав дахь аргументыг зааж өгнө

```

1  /* Example 95:
2     Reading a random access file sequentially */
3  #include <stdio.h>
4
5  /* clientData structure definition */
6  struct clientData {
7     int acctNum;          /* account number */
8     char lastName[ 15 ]; /* account last name */
9     char firstName[ 10 ]; /* account first name */
10    double balance;        /* account balance */
11 }; /* end structure clientData */
12
13 int main( void )
14 {
15     FILE *cfPtr; /* credit.dat file pointer */
16
17     /* create clientData with default information */
18     struct clientData client = { 0, "", "", 0.0 };
19
20     /* fopen opens the file; exits if file cannot be opened */
21     if ( ( cfPtr = fopen( "credit.dat", "rb" ) ) == NULL ) {
22         printf( "File could not be opened.\n" );
23     } /* end if */

```



```

24 else {
25     printf( "%-6s%-16s%-11s%10s\n", "Acct", "Last Name",
26         "First Name", "Balance" );
27
28     /* read all records from file (until eof) */
29     while ( !feof( cfPtr ) ) {
30         fread( &client, sizeof( struct clientData ), 1, cfPtr );
31
32         /* display record */
33         if ( client.acctNum != 0 ) {
34             printf( "%-6d%-16s%-11s%10.2f\n",
35                 client.acctNum, client.lastName,
36                 client.firstName, client.balance );
37         } /* end if */
38
39     } /* end while */
40
41     fclose( cfPtr ); /* fclose closes the file */
42 } /* end else */
43
44 return 0; /* indicates successful termination */
45
46 } /* end main */

```

fread нь санамсаргүй
хандалттай файлаас байтуудыг
санах ойн байрлал руу уншина

Acct	Last Name	First Name	Balance
29	Brown	Nancy	-24.54
33	Dunn	Stacey	314.33
37	Barker	Doug	0.00
88	Smith	Dave	258.34
96	Stone	Sam	34.98

Дүгнэлт: Си-гийн файл боловсруулалт

- ▶ `FILE *filePtr;` FILE нь бүтэц төрөл
- ▶ `filePtr = fopen("filename", "r");` хэрэв файл байхгүй бол NULL утга буцаана
- ▶ Файлыг нээж болох эсэхийг шалгах
 - `if ((cfPtr = fopen("word.dat", "w")) == NULL) {
else { }`
- ▶ Програм дуусахад файлууд хаагддаг боловч `fclose(filePtr)` гэсэн загвараар файлуудаа хааж байх хэрэгтэй
- ▶ Си програм ажиллаж эхлэхэд `stdin`, `stdout`, `stderr` гэсэн гурван файл автоматаар үүсдэг
- ▶ Талбар->бичлэг->файл->өгөгдлийн сан: файлд мөр бүр нь бичлэг болох ба багана нь талбар болно
- ▶ Дараалсан болон санамсаргүй хандалттай файл