Random Access in Files

fseek, ftell, rewind

Random Files

- Random file access refers to the ability to read or modify information directly at any given position in a file
- This is done by getting and setting a file position indicator, which represents the current access position in the file associated with a given stream
- We can use functions like ftell, rewind and fseek to have random access in files.

ftell and rewind functions

- To determine where the file pointer is,use:

  long ftell (FILE * fp) ;

  - Returns a long giving the current position of file pointer in bytes.
  - The first byte of the file is byte 0.
  - If an error occurs, ftell () returns -1.

- You can “rewind” and start reading from the beginning of the file by using.

  void rewind (FILE * fp) ;

ftell example

ftell (FILE);

The ftell function returns the current file position of the given FILE.

```c
#include <stdio.h>

int main()
{
     FILE *ptr=fopen("ha.dat","w");
     fprintf(ptr,"saravan");
     int position=ftell(ptr);
     printf("Value of 'position' = %d\n",position);

     fprintf(ptr,"sof");
     position=ftell(ptr);
     printf("Value of 'position' = %d\n",position);

     fclose(ptr);
     return 0;
}
```

program output

```
Value of 'position' = 7
Value of 'position' = 10
```

Random Access using fseek

fseek enables random access in a file

int fseek (FILE * fp, long offset, int whence) ;

- offset is the number of bytes to move the position indicator
- whence – This is the position from where offset is added. It is specified by one of the following constants
- Returns zero if successful, or else it returns a non-zero value

- Three options/constants are defined for whence
  - SEEK_SET
    - move the indicator offset bytes from the beginning
  - SEEK_CUR
    - move the indicator offset bytes from its current position
  - SEEK_END
    - move the indicator offset bytes from the end

Structures and random access of Files

- Normally, `structs` are used in random access files
- Example program
  - A program to process bank accounts
  - Each record has an account number used as the record key, a last name, a first name and a balance
  - First a random access file of 100 records is created with 0 for account number, NULL for first name and last name and 0.0 for balance

Example – creating a random accessed file (from Deitel)

```
/* Creating a randomly accessed file sequentially */
#include <stdio.h>

#define NRECORDS  100

struct clientData {
   int acctNum;
   char lastName[15];
   char firstName[15];
   double balance;
};
```


Example – creating a random accessed file (from Deitel)

```
int main()
{
   int i;
   struct clientData blankClient = {0, "", "", 0.0};
   FILE *cfPtr;

   if ( (cfPtr = fopen("credit.dat", "wb")) == NULL)
      printf("File could not be opened.\n");

     for (i = 1 ; i <= NRECORDS ; i++)
          fwrite(&blankClient, sizeof(struct
   clientData), 1, cfPtr);

     fclose(cfPtr);
     return 0;
}
```


Example – writing data randomly to a random accessed file (from Deitel)

```
/* Writing data randomly to a random accessed file */
#include <stdio.h>

#define NRECORDS  100

struct clientData {
   int acctNum;
   char lastName[15];
   char firstName[15];
   double balance;
};

int main()
{
   struct clientData client = {0, "", "", 0.0};
   FILE *cfPtr;
```


Example – writing data randomly to a random accessed file (from Deitel)

```
if ( (cfPtr = fopen("credit.dat", "r+b")) == NULL)
   printf("File could not be opened.\n");

  printf("Enter account number (1 to %d, 0 to end
input)\n? ", NRECORDS);

scanf("%d", &client.acctNum);
```


Example – writing data randomly to a random accessed file (from Deitel)

```
   while(client.acctNum != 0)
   {
      printf("Enter lastname, firstname, balance\n? ");
      fscanf(stdin, "%s%s%lf", client.lastName,
client.firstName, &client.balance);
      fseek(cfPtr, (client.acctNum - 1) * sizeof(struct
clientData), SEEK_SET);
      fwrite(&client, sizeof(struct clientData), 1,
cfPtr);
      printf("Enter account number (1 to %d, 0 to end
input)\n? ", NRECORDS);
      scanf("%d", &client.acctNum);
   }
   fclose(cfPtr);

   return 0;
}
```


Thank you