SEQUENTIAL vs RANDOM FILE DATA ACCESS

Page :
Date :

Difference b/w Random and Sequential File data Access

Sequential	Random
① Sequential Access to a data file means that the computer reads or writes information to the file sequentially, starting from the begining of the file and proceeding step by step.	Random Access to a file means that the Computer System reads or write information anywhere in the data file. This type of operation is also called "Direct Access" because the Computer System knows when the data is stored (using indexing) and hence goes "directly" and reads the data.
② Sequential access has advantages when you access information in the same order all the time. Also it is faster than random access.	Random Access file has the advantage that you can search through it and find the data you need more easily (using indexing for example). Random Access Memory (RAM) in computer works like that.

Page :
Date :

Why? Sequential Access is faster than Random Access.

1# Accessing data sequentially is much faster than accessing it randomly because of the way in which disk hardware works. The seek operations, which occurs in when the disk had positions itself at the right disk cylinder to access data takes more time than any other part of the I/O process.

OR

2# Because reading randomly involves a higher number of seek operations than does sequential reading, random reads and deliver a lower rate of output. The same is true for random writing.

Random Access Files in C

It can be allowed by 3 functions

(i) ftell()
(ii) rewind()
(iii) fseek()

(1) **ftell()** : The function returns the value of current pointer position in the file. The value is count from the beginning of file.

Syntax:

```
ftell(ptr);
```

(2) rewind() : The function is used to move the file pointer to the beginning of the given file.

Syntax:

```
rewind(ptr);
```

(3) **fseek()**: The function is used for seeking the pointer position in the file at specified byte.

Syntax:

fseek(filepointer, displacement(int), pointer-position);

Filepointer: it is the pointer which points to the file.

displacement: it is +ve or -ve value. This no. of byte which are skipped backward (if -ve) or forward (if +ve) from current position. This is attached with L because this is a long integer.

Pointer Position: This is set the pointer position in the file.

Value	Pointer Position
0	Beginning of file
1	Current position.
2	End of file.

eg- fseek(p, 10L, 0)

0 means position is at beginning of file. from this statement pointer position is skipped 10 bytes from beginning of file.

feof (Test for End of file)

Page :
Date :

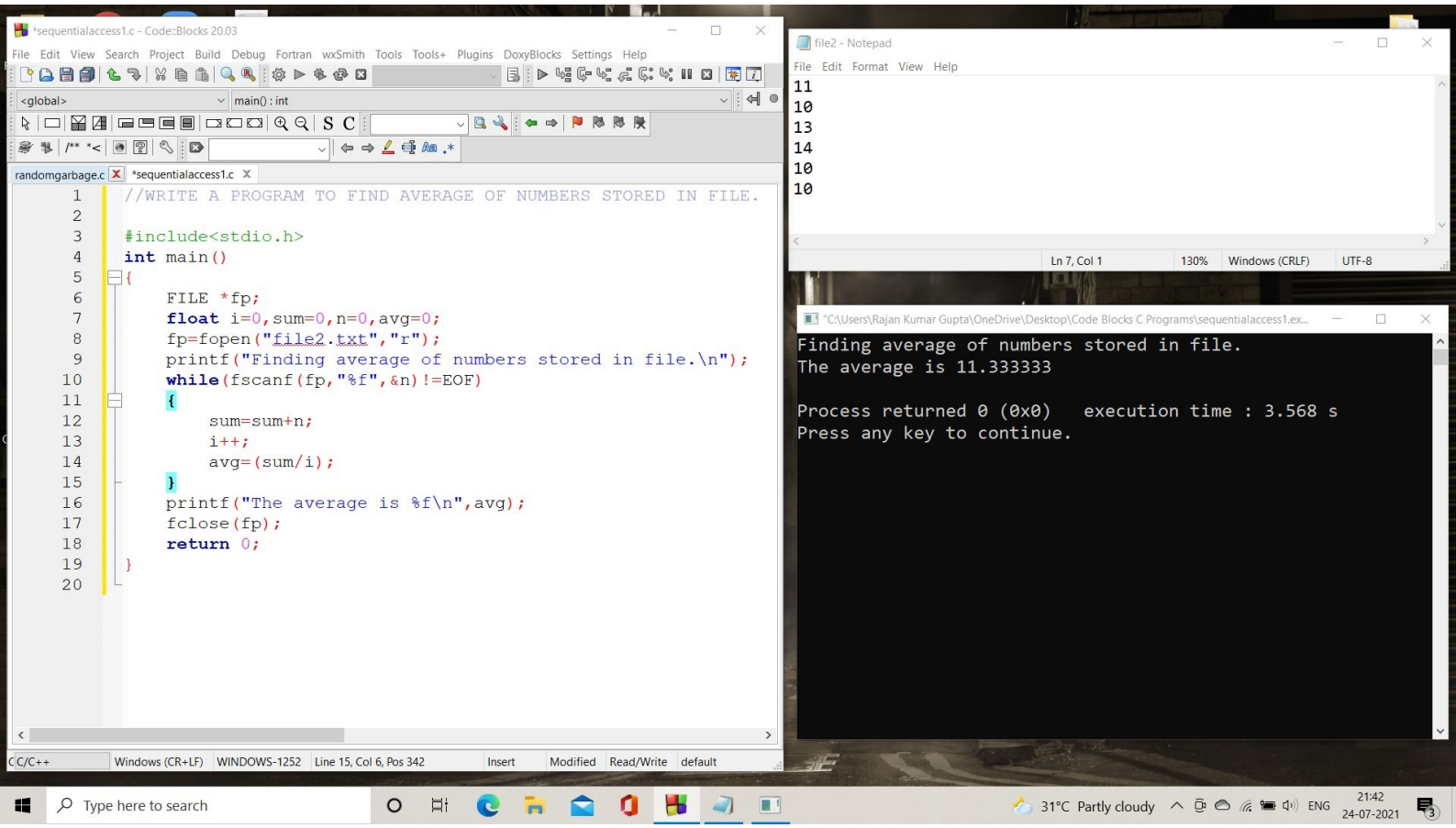
e.g. fseek(p, -5L, 1)
from this statement pointer position is skipped by 5 bytes backward from the current position.

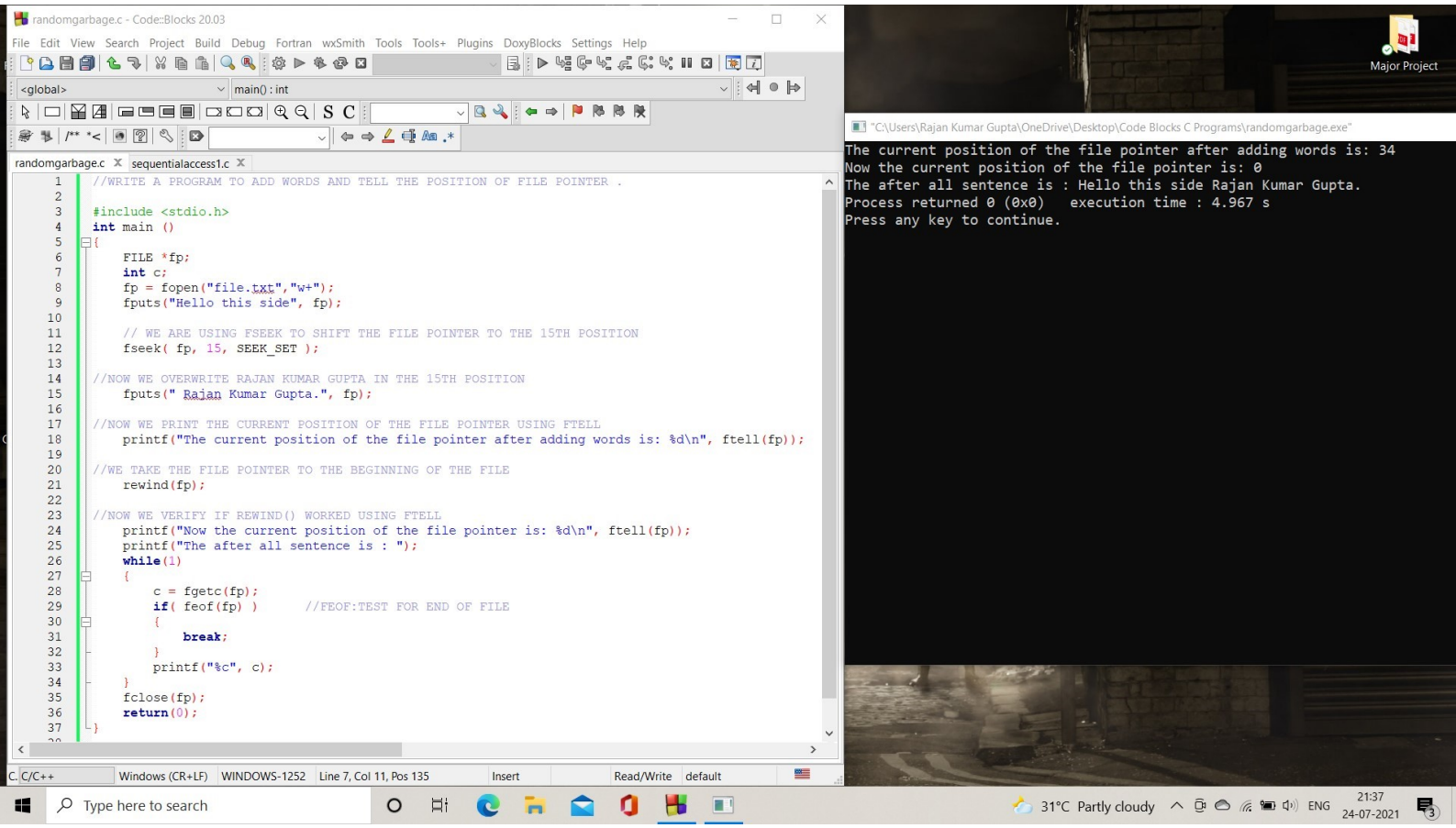
More examples

fseek(fp, 0, 0); → 0 means no movement
→ beginning file. (Rewind)
fseek(fp p, 1); stay at current position.
fseek(fp, 0, 2); go to EOF
Fseek(fp, +m, 0); go forward by +m bytes from beginning
fseek(fp, -m, 1); go backward by m bytes from current position
Fseek(fp, m, 2); go forward by m bytes from EOF.

Imp

SEEK_END : it denotes end of file
SEEK_SET : it denotes starting of file
SEEK_CUR : it denotes file pointer's Current Position


*sequentialaccess1.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : int
randomgarbage.c
*sequentialaccess1.c
//WRITE A PROGRAM TO FIND AVERAGE OF NUMBERS STORED IN FILE.
#include<stdio.h>
int main()
{
FILE *fp;
float i=0,sum=0,n=0,avg=0;
fp=fopen("file2.txt","r");
printf("Finding average of numbers stored in file.\n");
while(fscanf(fp,"%f",&n)!=EOF)
{
sum=sum+n;
i++;
avg=(sum/i);
}
printf("The average is %f\n",avg);
fclose(fp);
return 0;
}
C/C++ Windows (CR+LF) WINDOWS-1252 Line 15, Col 6, Pos 342 Insert Modified Read/Write default
file2 - Notepad
File Edit Format View Help
11
10
13
14
10
10
Ln 7, Col 1 130% Windows (CRLF) UTF-8
"C:\Users\Rajan Kumar Gupta\OneDrive\Desktop\Code Blocks C Programs\sequentialaccess1.ex...
Finding average of numbers stored in file.
The average is 11.333333
Process returned 0 (0x0) execution time : 3.568 s
Press any key to continue.
Type here to search
31°C Partly cloudy
ENG
21:42
24-07-2021


randomgarbage.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : int
randomgarbage.c
sequentialaccess1.c
//WRITE A PROGRAM TO ADD WORDS AND TELL THE POSITION OF FILE POINTER .
#include <stdio.h>
int main ()
{
FILE *fp;
int c;
fp = fopen("file.txt","w+");
fputs("Hello this side", fp);
// WE ARE USING FSEEK TO SHIFT THE FILE POINTER TO THE 15TH POSITION
fseek(fp, 15, SEEK_SET);
//NOW WE OVERWRITE RAJAN KUMAR GUPTA IN THE 15TH POSITION
fputs(" Rajan Kumar Gupta.", fp);
//NOW WE PRINT THE CURRENT POSITION OF THE FILE POINTER USING FTELL
printf("The current position of the file pointer after adding words is: %d\n", ftell(fp));
//WE TAKE THE FILE POINTER TO THE BEGINNING OF THE FILE
rewind(fp);
//NOW WE VERIFY IF REWIND() WORKED USING FTELL
printf("Now the current position of the file pointer is: %d\n", ftell(fp));
printf("The after all sentence is : ");
while(1)
{
c = fgetc(fp);
if(feof(fp)) //FEOF:TEST FOR END OF FILE
{
break;
}
printf("%c", c);
}
fclose(fp);
return(0);
}
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 7, Col 11, Pos 135
Insert
Read/Write
default
Major Project
"C:\Users\Rajan Kumar Gupta\OneDrive\Desktop\Code Blocks C Programs\randomgarbage.exe"
The current position of the file pointer after adding words is: 34
Now the current position of the file pointer is: 0
The after all sentence is : Hello this side Rajan Kumar Gupta.
Process returned 0 (0x0) execution time : 4.967 s
Press any key to continue.
Type here to search
31°C Partly cloudy
ENG
21:37
24-07-2021