This lecture talks about the fifth programming assignment.

This assignment asks you to write a program that can read a file of numbers, adds these numbers, and saves result in another file.

This program is different from the fourth programming assignment. In the fourth assignment, the program reads one character at a time. In this, the fifth, assignment, the program needs to read integers.

For this input, the first number is -598. It should be treated as a number negative five hundred ninety eight. This should not be treated as four characters: negative sign, five, nine, and eight.

Fortunately, the C language provides different methods to read files. We have seen how to use F get C to read one character. C also provide F scan F to read data in different formats. You have learned scan F in CS 1 5 9. F scan F is very similar.

The only difference is that F scan F’s first argument is a file pointer.

The second argument is the format. If you use % d, F scan F can read an integer.

It is possible to read two or more integers, if you add % d multiple times between the quotation marks.

The function returns an integer, indicating how many have been read. If nothing can be read, the function returns zero.

Another way to read from a file is the F get S function. This function can read a string from a file. This function takes three arguments:

The first argument is the memory space that will store the data. It is necessary to ensure that there is enough space to store the data.

The second argument is the largest amount of data to read.

The third argument is a file pointer.

What does F get S do? It will read at most the number of size, the second argument, of characters from the file and store the data in the buffer.

F gets S stops in one of the following conditions. First, F get S stops if it reaches the end of the file. F get S may stop if it reads the new line character.

Please notice that F get S does not stop when it encounters space.

Let’s see an example using three different ways reading the content of a file.

This program opens a file for reading using the argument A R G Vee one as the file name.

This program uses three different methods to read the content of the file. The first method uses F get C. F get C reads one character at a time. The program keeps reading until it reaches the end of the file.

Then, the program uses F seek to return to the beginning of the file. F seek takes three arguments. The first is the file pointer. The second is the distance, also called offset. The third argument is the reference point. In this example, the offset is zero and the reference is seek set, meaning the beginning of the file.

The program then reads one integer at a time using F scan F until reaching the end of the file.

Please notice that the conditions inside while are different for F get C and F scan F. .

For F get C, when it reaches the end of the file, it returns E O F . . E O F is defined as -1.

For F scan F, when it reaches the end of the file, it returns zero.

We use F seek to return to the beginning of the file again.

The third method uses F get S. . This method has three arguments, the first is the space to store the data. The second argument is the length to read from the file. The third is the file pointer.

Please notice that condition for while is different again. When F get S reaches the end of the file, it returns NULL.

This slide shows the execution of the program. Suppose the input file has these numbers: 1

23

456

78 on the first line. -365

202

642 on the second line. The third line has only one number 3. The fourth line has one number -7.

The output is shown in the middle and the right side of the slide. In the middle, it can be seen that F get C treats each character as individual letter. F get C reads the letter one and its ass kee value is 49. Then, F get C reads the space. The ass kee value of space is 32. F get C breaks 23 into two characters: 2 and 3. F get C does not have the concept of twenty three. Each character is treated independently.

The right side of this slide shows using F scan F to read integers. F scan F can read an integer. The first integer is a number one. The second integer is the number twenty three. Please notice the difference. F scan F treats this as an integer twenty three, not two characters 2 and 3.

The integers are divided by spaces or new lines.

When the program uses F get S to read the file, the spaces are not used to stop reading. F get s stops reading in only one of three conditions: First, it reaches the end of the file. Second, it reaches the end of a line. Third, it has read enough specified by the second argument.

C has another way of reading data. In this case, it reads from keyboard using get S. . However, as the document has said, get S is considered insecure. Thus, you should not use get S. .

Earlier, we use F seek to go to the beginning of a file. This slide shows the document of F seek and F tell. As mentioned in an earlier lecture, each file has a marker. The marker is set to the beginning of the file after F open. F seek can move the marker to specific locations. The location is specified by two arguments. One argument is the offset. The second argument is the reference point. SEEK SET means the beginning of the file. SEEK C U R means the current location. SEEK E N D means the end of the file.

F tell returns the location of the marker from the beginning of the file. The beginning of the file is zero.

This is an example how F tell may do. The program uses F get C to read one character. Then, the program calls F tell and prints the marker’s location. Next, the program uses F scan F to read one integer and uses F tell to report the marker’s location again. Finally, the program uses F get S to read the rest of the line and reports the marker’s location.

If the input is shown here, what is the program’s output?

The output of F tell is 1, 4, and 12. Let me explain why the output is these three numbers.

At the very beginning, the marker’s location is zero. F get C reads only one characters. Thus, at this moment, F tell reports the marker’s location as one.

F scan F finds an integer. The integer is 23. F scan F has to skip the space and reads two characters, 2 and 3. Thus, the marker’s location is 4.

F get S reads as many as 79 characters, or stops at the new line character. The input file has space, 456, space, 78, and new line character. If you add all of them together, that is 8 characters. Since the marker was at location 4 earlier, the marker’s new location should be 12.

The last topic of this lecture is how to write to a file.

There are several ways to write to a file. This slide uses F print F. . Before writing to a file, we have to open the file for writing using W as the mode in the second argument of F open.

F print F is the same as print F with only one difference. The first argument of F print F is the file pointer.

To print an integer, use % D. . To print one character, use % C. . To print a string, use % S . .