C PROGRAMMING FOR ENGINEERS

Input & Output Data Files

Data Files – File Pointer

- Each data file used in a program must have a file pointer associated with it.
- A file pointer is declared by:

```
FILE *pointername;
```

The word FILE must be fully capitalized.

The asterisk before the pointer name indicates that the variable is a pointer variable.

Example: `FILE *weather;`

fopen Function

- The fopen function is used to assign a specific file to a file pointer.
- Example:

```
weather = fopen("weather.txt","r");
```

File pointer

File name

File status:
r for read.
w for write.
Must use lower case.

NOTE: "weather.txt" only defines the file name, not the location of the file.

fopen Function (continued)

- To use the fopen function effectively, a file location must be predefined. This is generally done using a preprocessor directive that specifies the file location.
- Example:

```
/* Processor directive */
#define inputfile "u:\\engr 200\\weatherdata.txt"
```

Must use double backslashes. The first backslash is the escape character, and the second backslash is the directory path.

Putting it all Together

Example:

```
/* Preprocessor directives */
#include <stdio.h>
#include <math.h>
#define  inputfile  "u:\\engr 200\\weatherdata.txt"

/* Main function */
int main(void)
{
        /* Declare variables  */
        FILE *weather;

        /* Open input file */
        weather = fopen(inputfile, "r");
```

Defines location of input file.

Declares pointer.

Opens input file and assigns address to pointer.

Input File Status

- To ensure that the input file successfully opens, a verification needs to be performed. This is accomplished by using an **if** structure to error trap.

Example:

```
weather = fopen(inputfile, "r");
if(weather == NULL)
{
   printf("\n\nERROR OPENING INPUT FILE\n\n"
          "\n\nPROGRAM TERMINATED\n\n);
   return 0;
}
```

If an input data file cannot be opened, **fopen** returns a value of NULL. NULL is a symbolic constant defined in stdio.h, and it has a value of the character zero.

Note: To ensure program success, the input and output files should be stored in the same working directory that the source program resides.

NOTE: NULL must be fully capitalized.

Closing Files

- The fclose function is used to close a file.

Example:

```
fclose(weather);
```

File pointer name is all that is needed between parentheses. This statement will close the input file from the previous examples.

Note: Input and output files can be closed at any stage in a program, but once closed they cannot be accessed until they are reopened again. Thus, most programmers will not close any files until the end of the program.

Reading From Data Files

- The fscanf function is used to read data from an input file.

Example:

```
fscanf(weather,"%i,%f",&year,&inches);
```

Input file pointer.

Input variables.

Control string containing input file specifiers.

The comma is the input file delimiter. This must match with the input file data separation delimiter.
Common types are:
- spaces
- commas
- tabs

Writing to an Output File

- The fprintf function is used to write data to an output file.

Example:

```
fprintf(results,"Year      Inches");
fprintf(results,"\n%4i     %4.1f",year,inches);
```