File I/O - Text

Last updated 12/6/22

These slides introduce text file operations

File I/O - Text

- File Formats
  - Files can contain information in 2 different formats
  - Text
    - Stores characters (numbers are stored as their ascii values)
    - Line terminated by a newline (\n)
  - Binary
    - Raw bytes
    - File terminated by “end of file” EOF

text 257A

0011 0010	0011 0101	0011 0111	0100 0001
ascii 2	5	7	A

binary 257A

0000 0001	0000 0001	0100 0001
binary 257		A

This assumes 257 was a 16b integer
a full sized int would require 4 bytes 0x00000101

File I/O - Text

- File I/O - text
  - Need to create a “stream” to transfer the data to/from the file from/to our program
  - Identify the stream by name
  - Use a pointer

```
FILE* pointer_name;

FILE* Student_Data_ptr;
```


File I/O - Text

- Stream Pointer
  - Need to identify the file we are creating the stream to/from
  - "open" the file
  - assign the pointer to the opened file

```
file_pointer = fopen("filename", "mode");

Student_Data_ptr = fopen("ee1601.dat", "r");

Student_Data_ptr =
        fopen("C:\\users\\tim\\winter\\ee1601.dat", "r");
```

the file extension .dat is commonly used

File I/O - Text

- File I/O - text
  - Open file – modes

r	read only, start at beginning if does not exist → error
w	write only, start at beginning (erase all contents) if does not exist → creates it
a	append only, start at end of current data if does not exist → creates it

Returns address of file or NULL if an error occurs

NULL is defined in the stdio library

File I/O - Text

- Error checking
  - If the fopen() returns a NULL – we have an error

```
// create a stream pointer for the file
FILE * DataFile_strm_ptr;

//create a new file
if((DataFile_strm_ptr = fopen("myDataFile.dat", "w")) == NULL){
     printf("Error opening file myDataFile.dat\n");
     exit (100);    // terminate program
    } // end if
```

exit – exits the program
requires <stdlib.h>

File I/O - Text

- Close a file

```
fclose(file_pointer);

fclose(Student_Data_ptr );
```

File I/O - Text

- Formatting stream data – write
  - Uses the same formatting conventions as printf

```
int fprintf(FILE*                stream_pointer,
            const char* string   "control_string",
            ... )
```

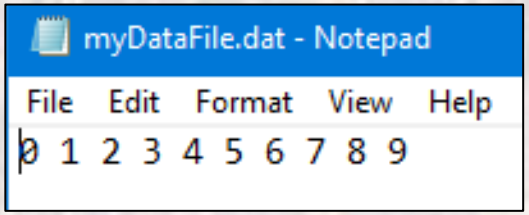
... represents additional arguments

returns the # of characters written

File I/O - Text

- Write a series of integers to a file

```c
/*  file_io_text.c
      Created by johnsontimoj
      Rev 0, 11/15/17
 */
// read and write to a data file

#include <stdio.h>
#include <stdlib.h>

int main(void){
   setbuf(stdout, NULL);  // disable buffering

   // create a stream pointer for the file
   FILE * DataFile_strm_ptr;

   //create a new file
   if((DataFile_strm_ptr = fopen("myDataFile.dat", "w")) == NULL){
        printf("Error opening file myDataFile.dat\n");
        exit (100);    // terminate program
       } // end if

   // write a series of integers - 1 at a time
   int i;
   for(i=0; i<10; i++){
       fprintf(DataFile_strm_ptr, "%i ", i);
   }

   // close the file
   fclose(DataFile_strm_ptr);

   return 0;
 } // end main
```



File I/O - Text

- Write a series of structures to a file

```c
/*  file_io_binary.c
      Created by johnsontimoj
      Rev 0, 11/15/17
 */
// read and write to a data file
#include <stdio.h>
#include <stdlib.h>

// structure definitions
// typedef version
typedef struct{
    int id;
    char name[26];
    float gpa;
} student;

void fprint_student(FILE* data_file, const student the_student);

int main(void){
   setbuf(stdout, NULL);  // disable buffering

   // create a stream pointer for the file
   FILE * DataFile_strm_ptr;

   //create a new file
   if((DataFile_strm_ptr = fopen("myDataFile.dat", "w")) == NULL){
      printf("Error opening file myDataFile.dat\n");
      exit (100);    // terminate program
   } // end if

   // create some student variables and pointers
   student stu1 = {234,
                   "Joe_Smith",
                   3.45
   };
   student stu2 = {.gpa=3.2, .name="Sara_Jones", .id=222};
   student stu3;

   // create an array to hold the students
   student std_ary[3] = {stu1, stu2, stu3};

   // output the array
   int i;
   for(i=0; i<3; i++){
       fprint_student(DataFile_strm_ptr, std_ary[i]);
   }

   return 0;
}// end main
```

```c
void fprint_student(FILE* data_file, const student the_student){
    fprintf(data_file, "%i %s %f\n", the_student.id, the_student.name, the_student.gpa);
    return;
}// end fprint_student
```

myDataFile.dat - Notepad

File Edit Format View Help

```
234 Joe_Smith 3.450000
222 Sara_Jones 3.200000
12520556 ¼þa 8236339266913422800000000000000000.000000
```

File I/O - Text

- Formatting stream data - read
  - Uses the same formatting conventions as scanf

```
int fscanf(FILE*                stream_pointer,
           const char* string   "control_string",
           ... )
```

... represents additional arguments

returns the # of characters written

File I/O - Text

- Read a series of integers from a file

```c
/*  file_io_binary.c
      Created by johnsontimoj
      Rev 0, 11/15/17
 */
// read and write to a data file

#include <stdio.h>
#include <stdlib.h>

int main(void){
   setbuf(stdout, NULL);  // disable buffering

   // create a stream pointer for the file
   FILE * DataFile_strm_ptr;

   //open an existing file
   if((DataFile_strm_ptr = fopen("myDataFile.dat", "r")) == NULL){
        printf("Error opening file myDataFile.dat\n");
        exit (100);    // terminate program
       } // end if

   // create and initialize an array
   int my_array[20];
   int i;
   for(i=0; i<20; i++){
       my_array[i] = 0;
   }
   for(i=0; i<20; i++){
       printf("%i ", my_array[i]);;
   }
   printf("\n");

   // read from the file
   for(i=0; i<10; i++){
       fscanf(DataFile_strm_ptr, "%i", &my_array[i]);
   }

   // print myArray
   for(i=0; i<20; i++){
       printf("%i ", my_array[i]);;
   }

   // close the file
   fclose(DataFile_strm_ptr);

   return 0;
 } // end main
```

Using the integer file from the write example

```
<terminated> (exit value: 0) Class_Cons_Projec
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9 0 0 0 0 0 0 0 0 0 0
```

File I/O - Text

- Read a series of structures from a file

```c
/*  file_io_binary.c
      Created by johnsontimoj
      Rev 0, 11/15/17
 */
// read and write to a data file
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// structure definitions
// typedef version
typedef struct{
    int id;
    char name[26];
    float gpa;
} student;

void fscan_student(FILE* data_file, student* the_student);

int main(void){
   setbuf(stdout, NULL);  // disable buffering

   // create a stream pointer for the file
   FILE * DataFile_strm_ptr;

   //open an existing file
   if((DataFile_strm_ptr = fopen("myDataFile.dat", "r")) == NULL){
      printf("Error opening file myDataFile.dat\n");
      exit (100);    // terminate program
   } // end if

   // create an array to hold the students
   student std_ary[3] = {0};

   // read from the file
   int i;
   for(i=0; i<3; i++){
       fscan_student(DataFile_strm_ptr, &std_ary[i]);
   }

   // print the structure
   printf("%i %s %f", std_ary[1].id, (*(std_ary+1)).name, (std_ary + 1)->gpa);

   return 0;
}// end main
```

```c
void fscan_student(FILE* data_file, student* the_student){
    fscanf(data_file, "%i", &(the_student->id));
    fscanf(data_file, "%s", the_student->name);
    fscanf(data_file, "%f", &(the_student->gpa));
    return;
}// end fscan_student
```

Using the structure file from the write example

```
<terminated> (exit value: 0) Cl
222 Sara_Jones 3.200000
```