File Handling in C

The slides here are not comprehensive. And covers some part of the lecture

File Reading list…

Read the following pages from Herbert Sheild Teach yourself book:

Pages: 257 to 297

What is a File?

- A *file* is a collection of related data that a computers treats as a single unit.
- Computers store files to secondary storage so that the contents of files remain intact when a computer shuts down.
- When a computer reads a file, it copies the file from the storage device to memory; when it writes to a file, it transfers data from memory to the storage device.
- C uses a structure called `FILE` (defined in `stdio.h`) to store the attributes of a file.

Steps in Processing a File

1. Create the stream via a pointer variable using the `FILE` structure:
   ```
   FILE *p;
   ```
2. Open the file, associating the stream name with the file name.
3. Read or write the data.
4. Close the file.

The basic file operations are

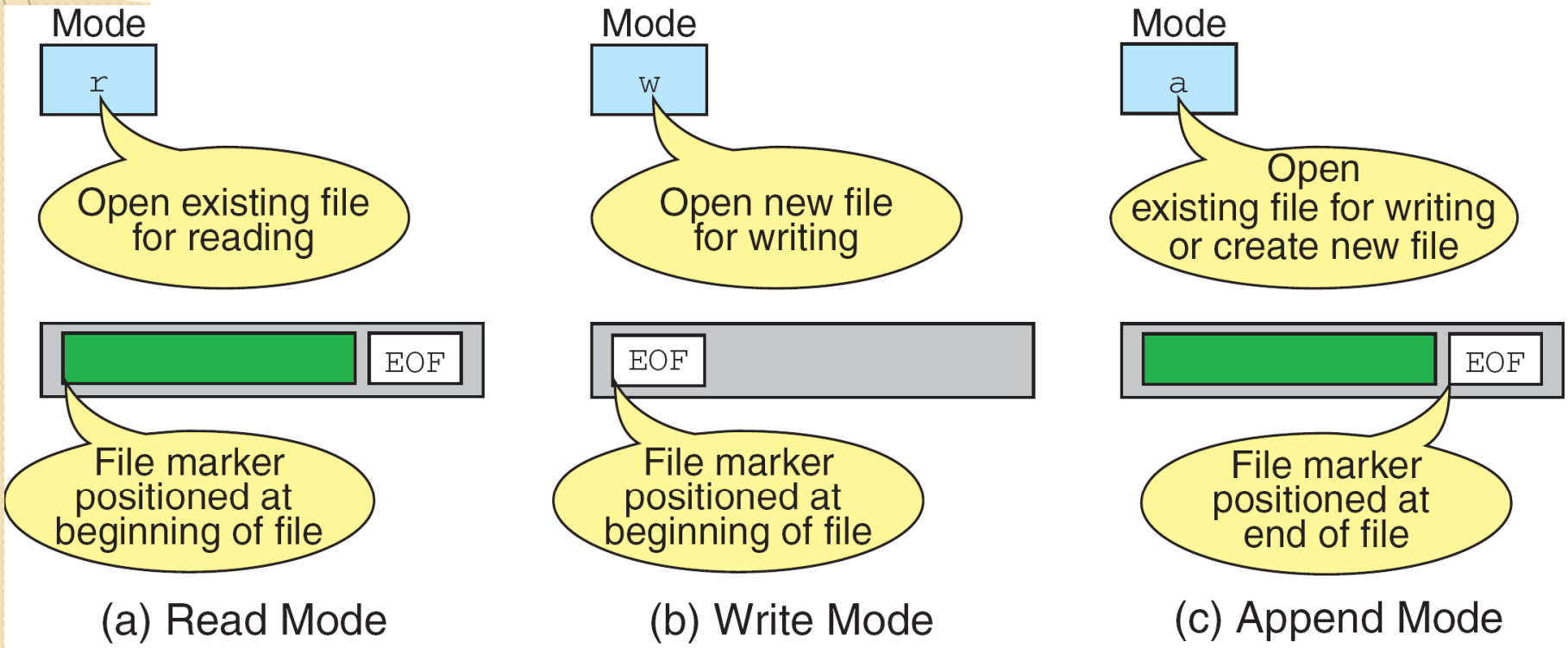
- fopen - open a file- specify how its opened (read/write) and type (binary/text)
- fclose - close an opened file
- fread - read from a file
- fwrite - write to a file
- fseek/fsetpos - move a file pointer to somewhere in a file.
- ftell/fgetpos - tell you where the file pointer is located.

File Open Modes

Mode	Meaning
r	Open text file in read mode • If file exists, the marker is positioned at beginning. • If file doesn't exist, error returned.
w	Open text file in write mode • If file exists, it is erased. • If file doesn't exist, it is created.
a	Open text file in append mode • If file exists, the marker is positioned at end. • If file doesn't exist, it is created.

from Table 7-1 in* Forouzan & Gilberg, *p. 400

More on File Open Modes



from Figure 7-4 in* Forouzan & Gilberg, *p. 401

Additionally,

- r+ - open for reading and writing, start at beginning
- w+ - open for reading and writing (overwrite file)
- a+ - open for reading and writing (append if file exists)

File Open

- The file open function (`fopen`) serves two purposes:
  - It makes the connection between the physical file and the stream.
  - It creates "a program file structure to store the information" C needs to process the file.
- Syntax:

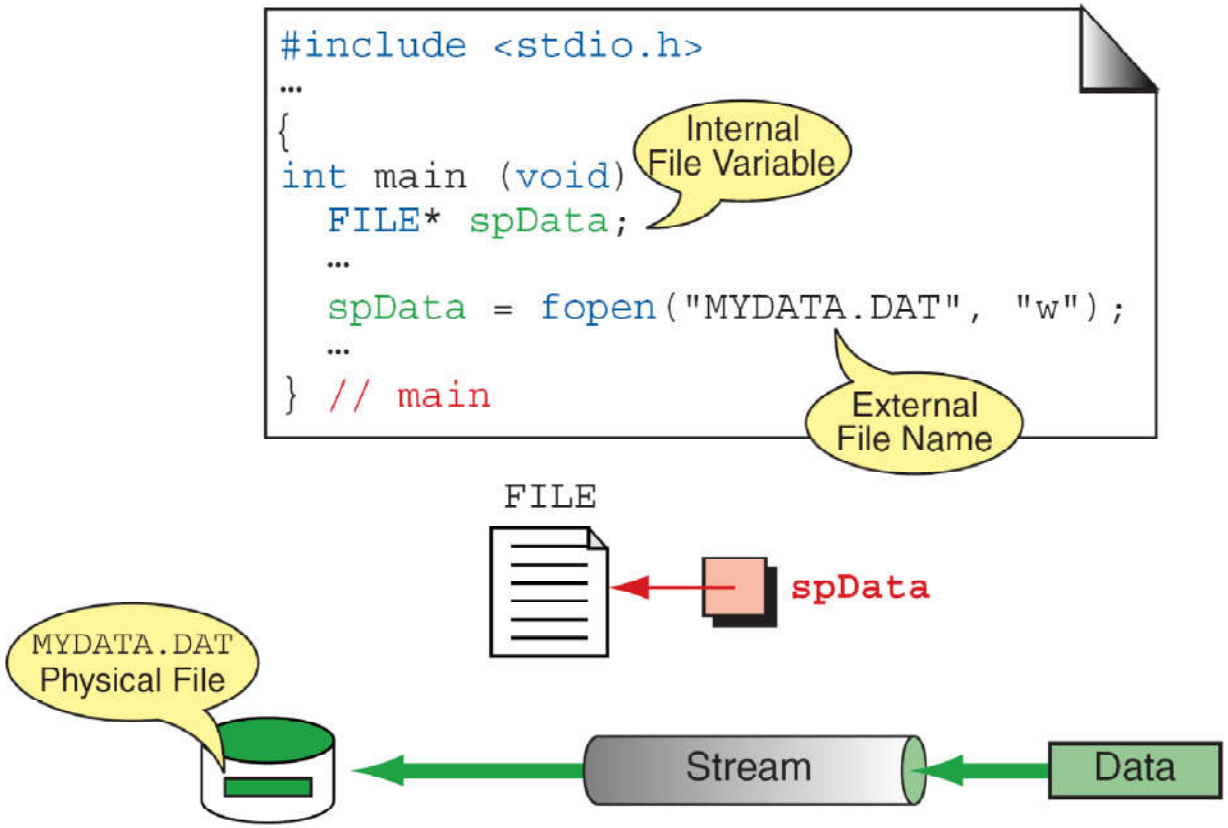
```
filepointer=fopen("filename",
"mode");
```

More On `fopen`

- The file mode tells C how the program will use the file.
- The filename indicates the system name and location for the file.
- We assign the return value of **fopen** to our pointer variable:

```
spData = fopen("MYFILE.TXT","w");
spData = fopen("d:\\MYFILE.TXT","w");
```

More On fopen



from Figure 7-3 in* Forouzan & Gilberg, *p. 399

Closing a File

- When we finish with a mode, we need to close the file before ending the program or beginning another mode with that same file.
- To close a file, we use `fclose` and the pointer variable:

```
fclose(spData);
```

fprintf()

Syntax:

fprintf (fp,"string",variables);

Example:

```
int i = 12;
float x = 2.356;
char ch = 's';
FILE *fp;
fp=fopen("out.txt","w");
fprintf (fp, "%d %f %c", i, x, ch);
fclose(fp);
```

fscanf()

Syntax:

fscanf (fp,"string",identifiers);

Example:

```
FILE *fp;
fp=fopen(“input.txt”,”r”);
int i;
fscanf (fp,“%d", &i);
printf(“%d”, i);
fclose(fp);
```

fgetc()

Syntax:

identifier = fgetc (file pointer);

Example:

```
FILE *fp;
fp=fopen("input.txt","r");
char ch;
ch = fgetc (fp);
printf("%c",ch);
fclose(fp);
```

fputc()

write a single character to the output file, pointed to by fp.

Example:

```
FILE *fp;
char ch;
fp=fopen("input.txt","r");
putc (ch, fp);
fclose(fp);
```

End of File

- There are a number of ways to test for the end-of-file condition. Another way is to use the value returned by the *fscanf* function:

```
FILE *fptr;
int istatus ;
fptr=fopen(“input.txt”,”r”);
istatus = fscanf (fptr, "%d", &var) ;
if ( istatus == feof(fptr1) )
{
        printf ("End-of-file encountered.\n”) ;
}
```

Reading and Writing Files

```
#include <stdio.h>
int main ( )
{
   FILE *outfile, *infile ;
   int b = 5, f ;
   float a = 13.72, c = 6.68, e, g ;

   outfile = fopen ("testdata", "w") ;
   fprintf (outfile, " %f %d %f ", a, b, c) ;
   fclose (outfile) ;
   infile = fopen ("testdata", "r") ;
   fscanf (infile,"%f %d %f", &e, &f, &g) ;
   printf (" %f %d %f \n ", a, b, c) ;
   printf (" %f %d %f \n ", e, f, g) ;
   fclose (infile) ;
}
```

Example

```
#include <stdio.h>
#include<conio.h>
void main()
{
    char ch;
    FILE *fp;
    fp=fopen("out.txt","r");
    while(2==2)
       {
       ch=getc(fp);
       if(feof(fp))
          break;
       printf("\n%c",ch);
    }
    getch();
 }
```

THANK YOU