File I-O

File IO

- Stands for File Input - Output
- Used to read and write files
- Use for when scanning in text isn't feasible, or if the output needs to be saved

Creating Files

- Files can be anything - .c, .txt, .pdf, etc...
- In this class, files we read/write to will be mainly .txt files
- To create a .txt file, use the command “nano filename.txt”

Files

- Files have a specific data type: FILE

- Stores pointers to memory addresses not used by the .c program

> **FILE *fp;** /*Makes a pointer, called fp, which will eventually store the memory location of a file*/

Opening Files

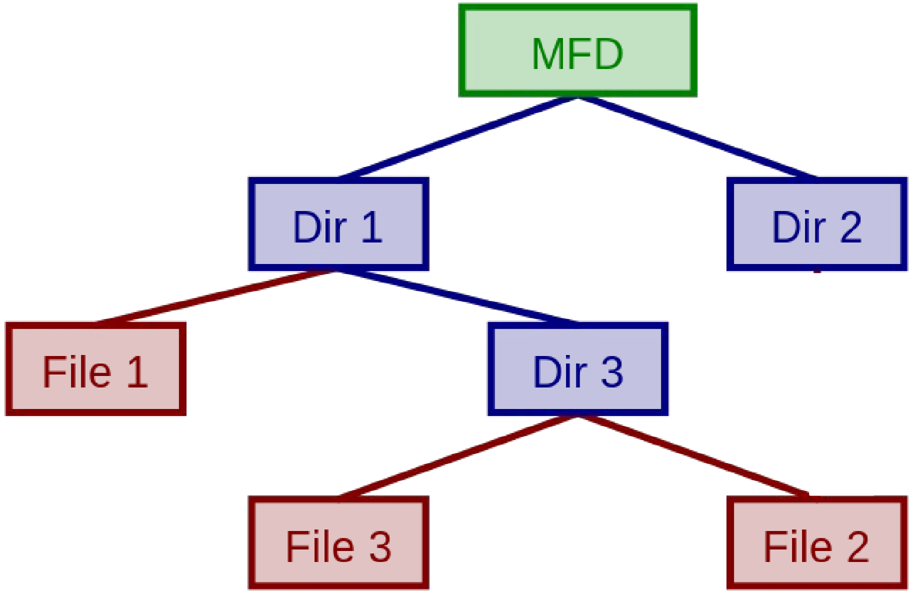
- Opening command: *fopen()*

- Takes two parameters:
  - The name of the file (or the path to a file)
  - The *mode* of the file

File Names

- Given as the *relative path* to the file

- Relative path = location of a file, *relative* to the current file



File Modes

- Three main modes
  - **“r”** → Read
    - Can only read a file
  - **“w”** → Write
    - Can only write to a file, but will overwrite anything previously written
  - **“a”** → Append
    - Can only write to the END of a file

Opening Files

- Opening command: *fopen()*

- Takes two parameters:
    - The name of the file (or the path to a file)
    - The *mode* of the file

```
> fp = fopen(“hello.txt”, “r”);
```

Closing Files

- Use *fclose()* to close the file buffer
- One parameter: the file pointer

```
> fclose(fp);
```

Reading Files

- Three ways to read files
    - By individual characters
    - By line
    - By format (not today)

Reading Files: Characters

- The command *getc()* reads the next character in the file stream
  - One parameter: the file pointer

```
> char next_char = getc(fp);
```

Reading Files: Lines

- Use *fgets()* to read the next line
    - Three parameters: The string which will store the input, the max_size, and the *file pointer*

```
> char name[50];

> fgets(name, 50, fp);
```

Reading Files: Lines

- Will read the line until: the file pointer hits a newline character OR the file pointer reaches max_size

```
> char name[50];

> fgets(name, 50, fp);
```

File IO Coding Challenge

- Open and read the Oriole_sequence.txt file on Github, and count the number of G’s, C’s, A’s, and T’s (same as part 1 of the DNA challenge)