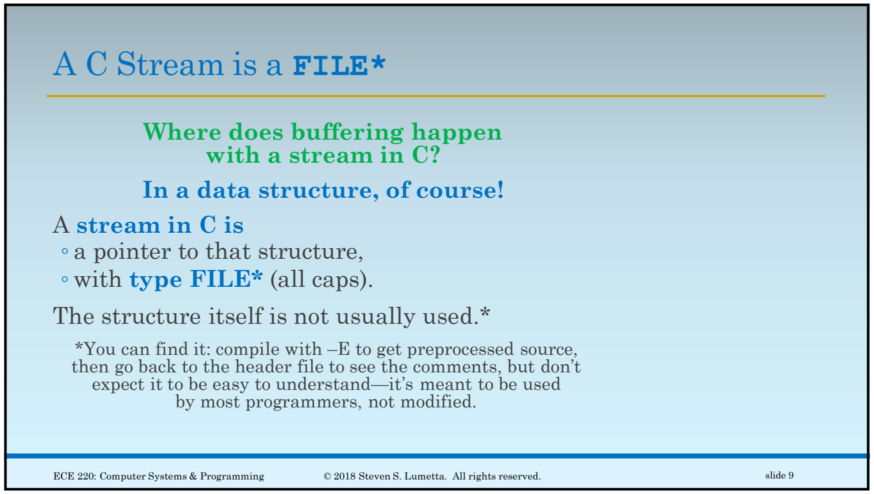


A C Stream is a **`FILE*`**

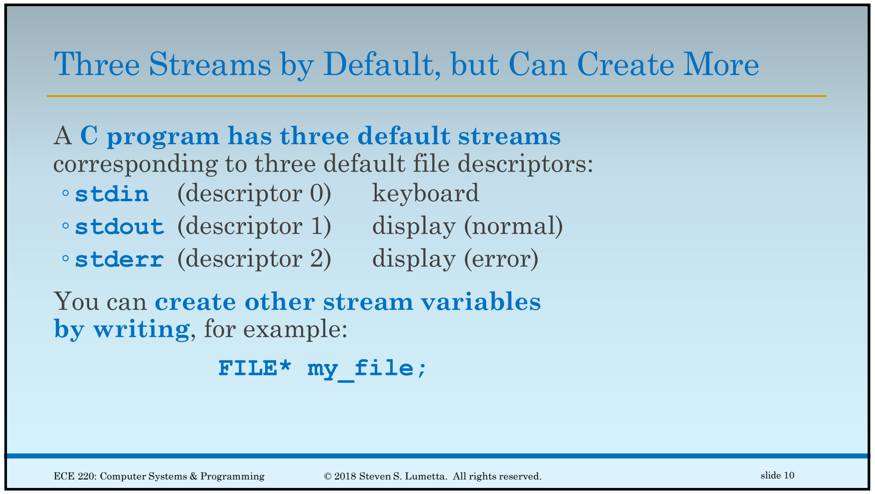
Where does buffering happen with a stream in C?

In a data structure, of course!

A **stream in C is**
- a pointer to that structure,
- with **type FILE*** (all caps).

The structure itself is not usually used.*

*You can find it: compile with –E to get preprocessed source, then go back to the header file to see the comments, but don't expect it to be easy to understand—it's meant to be used by most programmers, not modified.



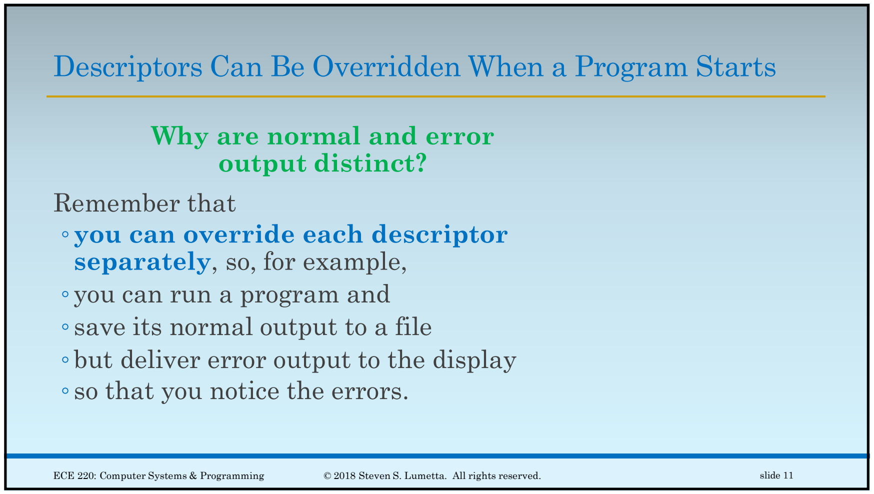
Three Streams by Default, but Can Create More

A **C program has three default streams** corresponding to three default file descriptors:

`stdin`	(descriptor 0)	keyboard
`stdout`	(descriptor 1)	display (normal)
`stderr`	(descriptor 2)	display (error)

You can **create other stream variables by writing**, for example:

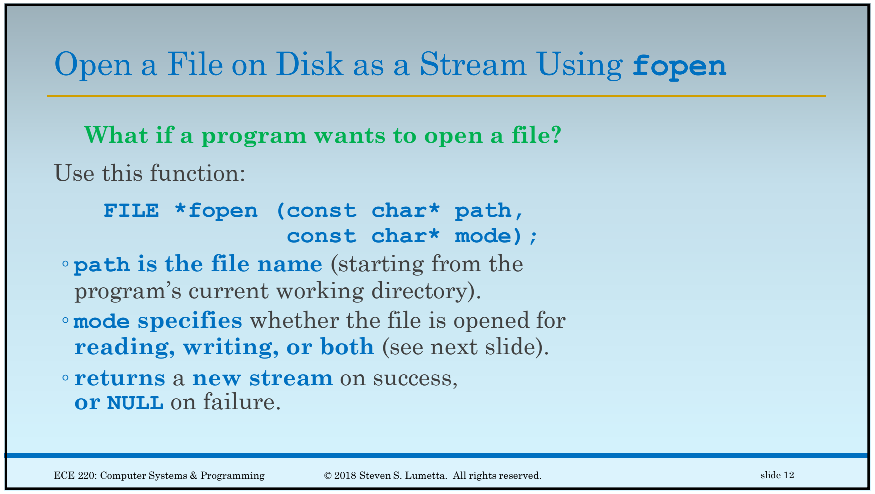
```
FILE* my_file;
```



Descriptors Can Be Overridden When a Program Starts

Why are normal and error output distinct?

Remember that
- **you can override each descriptor separately**, so, for example,
- you can run a program and
- save its normal output to a file
- but deliver error output to the display
- so that you notice the errors.



Open a File on Disk as a Stream Using **`fopen`**

What if a program wants to open a file?

Use this function:

```
FILE *fopen (const char* path,
             const char* mode);
```

- **path is the file name** (starting from the program's current working directory).
- **mode specifies** whether the file is opened for **reading, writing, or both** (see next slide).
- **returns** a **new stream** on success, **or `NULL`** on failure.