

# File IO – Brief Refresher

Dave Ogle

# Opening a File

- The command we will use for opening a file is:

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

- To learn more about fopen, you can use the 'man fopen' command in unix
  - The 'path' will be either a relative path, or a full path to a file
  - The common modes are: r, w, rb, wb, a (see man page for others)
  - If the file can be opened, the return value is a pointer to a FILE
  - If the file CANNOT be opened, the return is NULL and errno is set
- 
- Anytime you do IO in C, you will need to include `<stdio.h>`

# Reading a File

- Common ways to read a file:

- `fscanf()`, `fgetc()`, `fread()`
- For this lab I would recommend `fread()`
- Use 'man fread' to learn more about it

- Usage:

`size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);`

- `ptr` - pointer to buffer where you want to put the data
- `size` – length of the item you want to read (char, int, long, etc.)
- `nmemb` - number of items (members) you want to read per call
- `stream` – a pointer to the file (which should be open for read)
- Return code is a 'size\_t' (unsigned int). It indicates how many 'items' of length 'size' were read.

# Writing a File

- Common ways to write to file:
  - `fprintf()`, `fputc()`, `fwrite()`
  - For this lab I would recommend `fwrite()`
  - Use 'man fwrite' to learn more about it
- Usage:  
`size_t fwrite(void *ptr, size_t size, size_t nmemb, FILE *stream);`
  - `ptr` - pointer to buffer where the data to be written is
  - `size` – length of the item you want to write (char, int, long, etc.)
  - `nmemb` - number of items (members) you want to write per call
  - `stream` – a pointer to the file (which should be open for write)
  - Return code is a 'size\_t' (unsigned int). It indicates how many 'items' of length 'size' were written.

# Closing a File

- All file should be closed after the last read/write.
- This ensures that buffers are flushed properly
- To close a file
  - `int fclose (FILE *fp)`
    - fp – pointer to the open file
    - Return code is 0 if successful and it is EOF if unsuccessful (and errno is set)

# Other file functions to learn

- You should use the man command to familiarize yourself with commands like:
  - fseek – go to a position in a file
  - ftell - tells you the current position in the file
  - frewind – sets pointer to the start of the file
- These might be helpful in lab1!