EECS 2031

Software Tools

Module 9 – File Handling

YORK UNIVERSITÉ UNIVERSITY

File handling in C

- **`stdio.h`** contains several functions that allow us to read from and write to files
- Their names typically start with f:
  `fopen()` `fgets()` `fprintf()` etc.
- **`stdio.h`** also defines the type
  `FILE *f;`
- **`f`** is a pointer to a stream
- A stream could be a file or **`stdin`**, **`stdout`**, **`stderr`**

Streams

- **`printf()`** and **`scanf()`** have variants which operate on any stream:

```
printf(“hello”);
```

is the same as

```
fprintf(stdout, “hello”);
```

Streams

- To write something to stderr:

```
fprintf(stderr, "error!\n");
```

- Note that **`stdin`** is "read-only" and **`stdout`** and **`stderr`** are "write-only", so

```
fprintf(stdin,"will not work");
```

will fail (but will not crash your program)

Streams

- There are similar functions to **`getchar`** and **`putchar`** as well:

```
int fgetc(FILE *stream);

int fputc(int c, FILE *stream);
```

Opening and closing files

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

- Creates a new stream by opening a file whose name is provided in **`path`**
- If it fails, returns **`NULL`**
- To close the stream:

```
int fclose(FILE *fp);
```

Note: When a program exits all open files are automatically closed

Stream Modes

- The mode tells us whether we are reading or writing (it's a string)
- "`r`" - read-only
  - file must exist
- "`w`" – overwrite (write-only)
  - file is created if necessary
- "`a`" - append (write-only)
  - file is created if necessary

Status of Streams

- **`int feof(FILE *f)`** - returns non-zero if EOF has been reached on **`f`**,
  0 otherwise
- **`int ferror(FILE *f)`** - returns non-zero if an error has occurred on **`f`**,
  0 otherwise

- See **`filecopy.c`** for an example of file handling in C