

# CS 222: File I/O, Bits, Review

Chris Kauffman

Week 6-2

# Logistics

## Reading

- ▶ Ch 9 (file i/o)
- ▶ Ch 11 finish (practical programming)

## Exam 2 Thursday

Practice Problems posted tomorrow morning

## Homework

- ▶ HW 6 up sometime soon

## fgets(): Read a Whole Line

**Question:** Where does `fscanf(fin, "%s", buf)` stop reading?

`fgets(buf, size, file)`

- ▶ Read a whole line into `buf`
- ▶ Only reads in `size` characters
- ▶ `size` is a limitation for long lines

`gets(buf)`

- ▶ Read a whole line into `buf` from `stdin`
- ▶ **VERY DANGEROUS**

DESCRIPTION

Never use this function.

## fgets() Example

- ▶ Examine `fgets.c`
- ▶ Read lines from user interactively via `stdin`
- ▶ Read lines from a file
- ▶ Note limitations of needing to know the maximum line `SIZE` ahead of time

## Reading Into Structs: File says how many

- ▶ Will need to read read into a struct for an upcoming HW
- ▶ Examine this with `planet_t`, now with descriptions
- ▶ `read_planets_count_in_file.c`

## rewind(file) and 2-pass I/O

```
void rewind(FILE *f);
```

- ▶ Go back to the beginning of a file
- ▶ Use case
  1. Read whole file to count objects it contains
  2. Allocate memory for objects
  3. rewind() to beginning of file
  4. Read whole file storing objects in memory

## Reading Into Structs: Determine how many from file

read\_planets\_rewind.c

1. Open file for reading
2. Count lines in file -> number of planets
3. Allocate memory for planets
4. Rewind to beginning of file
5. Read each planet into allocated space
6. Print planets that have been read
7. Free space, close files, end

## Number Systems and Specifying Masks/Constants

Common number systems in computing and how to specify constants in C in that system:

```
/* All these have the same value, same bits */
int xb = 0b1011;    /* Binary (language extension) */
int xo = 013;      /* Octal */
int xd = 11;       /* Decimal */
int xh = 0xB;      /* Hexidecimal */

printf("%d %d %d %d\n",xb,xo,xd,xh);
```

## Recall Binary Ops

- ▶ Recall are the C symbols for LOGICAL versus BITWISE AND
- ▶ Describe the difference
- ▶ Symbols for BITWISE OR, XOR, INVERSION
- ▶ What's another set of bitwise operations that are cool?
- ▶ How do those work?
- ▶ How can I check if bit 5 is set (is a 1)?



## Review for Exam 2

Questions and Discussion now