Lecture 35: C Language
“Features”

Fall 2016
Kim Bruce & Peter Mawhorter

Thanksgiving

- Last week’s assignment is due today
  - Resources for help will be limited after noon on Wednesday
- No lab or assignment this week
- Graphs next week

#NoDAPL

---

Goto

- Go directly to a label
- Also known as “Progenitor of 1000 Bugs”
- Your processor only has goto

Goto Example

```c
int dont_use_goto(int x) {
  one:
    x += 5;
  two:
    x += 3;
    if (x < 12) goto three;
    return x;
  three:
    if (x % 8) goto two;
    goto one;
}
```
Break and Continue

- break gets you out of a single loop.
  - Everyone knows where you’re going
  - Can still be confusing
- continue takes you to the start of the current loop

No easy way to break out of 2+ loops at once

Goto Example

```c
int lazy_sum(int *ar, int width, int height) {
    int sum = 0;
    for(int x = 0; x < width; ++x) {
        for(int y = 0; y < height; ++y) {
            if (sum >= 30) {
                goto done;
            }
            sum += ar[x+y*width];
        }
    }
    return sum;
}

union int_and_float_u {
    int i;
    float f;
};
typedef union int_and_float_u int_and_float;
```

What happens if we set i or f?

Unions

- Did you ever want to declare two data structures that shared the same memory?
- Imagine threads, but for memory
  - Yup.
- Don’t use them.
Unions

- Multiple ‘views’ of the same data
- Used for data manipulation and either/or situations

Makefiles

- Contain rules:
  - target: dependencies
  - commands to build target from dependencies
  - Checks modification time of resources
  - Figures out build order
  - Commands are just shell commands

Example

```make
myprogram: myprogram.c myprogram.h  
clang myprogram.c myprogram.h -o myprogram
```

Can use wildcards:

```make
%.exe: %.c %.h  
clang $< -o $@
```

Makefile Variables

- Variables store common configuration
- Use $() syntax to get value
- Often used for common arguments, e.g., *how to compile*
Our Makefile