C: Strings and Advanced Topics

CS 62 - Spring 2016
Michael Bannister

This Week

- Weekly Assignment
  - 20 Questions Animal Game
- Weekly Lab (today)
  - Working with strings

C Strings

No real string objects in C

"C Strings" are just arrays of characters!

- Type of string is `char[]` or more typically `char*`
- `NULL` terminated: "ABC" = {'A','B','C','\0'}
- Array length is one more than string length

Working with C Strings

Working with C strings is difficult and error prone! Whenever possible you should only interact with C strings using the functions `string.h`.

Even better is to use a third party string library!

Exploring string.h

(In browser)

Posix Extensions

```
ssize_t getline(char **lineptr, size_t *n, FILE *stream);
```

Reads an entire line from stream and returns a heap allocated copy via lineptr.

```
int asprintf(char **strp, const char *fmt, ...);
```

Uses printf style format strings to generate a heap allocated string.

```
char *strdup(const char *s);
```

Returns a heap allocated copy of s.

Warning!

These POSIX standard functions allocate memory that you are responsible for freeing!

Const Qualifier

The `const` keyword is used to make a variable read only, i.e., it is used to define constants.

With points it is tricky to tell exactly what is being made constant!

Type Unsafe Generics

- `void*` can point to anything!
- In C you can implement generic algorithms and data structures with `void*`
- Dangerous as types are never checked!
- Typically need to use function pointers to define meaning
- Example: quick sort in `stdlib.h`

Preprocessor Macros

- The preprocessor language
  - Simple directives used to modify source files before being compiled
  - Directives: `#define`, `#undef`, `#include`, `#if`, `#ifdef`, `#ifndef`, `#else`, `#elif`, `#endif`, `#line`, `#error`, `#pragma`
  - Allow for conditional compilation
  - Macro functions
  - Implementation defined behavior

GOTO

```
#include <stdio.h>

int main(void) {
   for (int x = 0; x < 3; x++) {
      for (int y = 0; y < 3; y++) {
         printf("(%d;%d)\n", x, y);
         if (x + y >= 3) goto endloop;
      }
   }
endloop:;
}
```

Unions

(on board)