Homework Notes from Sean

- From now on, turn in HW via the online dropbox that is linked from the course website
- On homeworks, make sure you give support and/or specific examples when answering the questions
- OS is about tradeoffs and often there is not a single correct answer (there are incorrect answers)
- Back up your answer with evidence and strong reasoning

Looking Forward... Next Project

- Shells
- System Calls
- Forkbomb (don't use attu!)
  - Read info on web page
  - Try to log in
  - Let us know if there are any problems
- VMWare will be useful

Common C Pitfalls (1)

- What's wrong and how to fix it?

```
char* get_city_name(double latitude,
                    double longitude) {
  char city_name[100];
  …
  return city_name;
}
```

Common C Pitfalls (1)

- Problem: return pointer to statically allocated mem.
- Solution: allocate on heap

```
char* get_city_name(double latitude,
                    double longitude) {
  char* city_name = (char*)malloc(100);
  …
  return city_name;
}
```

- Slightly more subtle example:

```
typedef struct _city_info_t {
      char* name;
      … …
} city_info_t;

city_info get_city_name(double latitude,
      double longitude) {
  city_info_ city_info;
  char city_name[100];
  … …
  city_info.name = city_name;
  return city_info;
}
```

Common C Pitfalls (2)

- What's wrong and how to fix it?

```
char* buf = (char*)malloc(32);
strcpy(buf, argv[1]);
```

Common C Pitfalls (2)

- Problem: Buffer overflow
- Solution:

```
int buf_size = 32;
char* buf = (char*)malloc(buf_size);
strncpy(buf, argv[1], buf_size);
```

- Are buffer overflow bugs important?

Common C Pitfalls (3)

- What's wrong and how to fix it?

```
char* buf = (char*)malloc(32);
strncpy(buf, "hello", 32);
printf("%s\n", buf);

buf = (char*)malloc(64);
strncpy(buf, "bye", 64);
printf("%s\n", buf);

free(buf);
```

Common C Pitfalls (3)

- Problem: Memory leak
- Solution:

```
char* buf = (char*)malloc(32);
strncpy(buf, "hello", 32);
printf("%s\n", buf);
free(buf);
buf = (char*)malloc(64);
…
```

- Are memory leaks important?
  - OS, web server, web browser, your projects?

Bug in all previous examples

- We didn't handle memory allocation failures:

```
char *buf = (char*)malloc(32);
if (buf == NULL) return;
```

- You should do that in your code
- This is the hint that many of you have been asking for

Project 0

- Project 0 Due Friday at 11:59pm
- Submit using the turnin program on Linux machines
- Please follow instructions on projects page
- **Seriously, follow the instructions**
- Note: turnin accepts the last thing you submit
- Good comments help me deliver good grades

Project 0 Discussion

- Why pass a reference to a pointer?
  - Code example
- Hash table discussion
  - Hash function
  - Collision resolution
  - Key equality testing
  - Edge case behavior (sane and well-documented)

Questions and Answers

- OS concepts
- Project 0
- OMG WHAT IS A POINTER?