CS 240
Computer Systems

## Week 8: Conditional Variables and Act 1 Finale

CS 240, Spring 2021 - Week 8

*Wade Fagen-Ulmschneider*

### Blocking Operations

Throughout synchronization, we have introduced several "blocking calls" that block the CPU from moving forward until a condition is met. A few we've already covered include:

- **`pthread_mutex_lock`**: Blocks until the lock is in the unlocked state,
- **`pthread_join`**: Blocks until the thread exits,
- **`read`** / **`fread`**: Blocks until data is available to be read (ex: waiting for user input, or waiting for data from a file),
- ...and in MP5: **`wallet_chance_resoruce`** blocks until the resource request can be satisfied.

### Reflection on MP5

In MP5, your code may have had a structure similar to the following:

**wallet.c (busy waiting)**

```c
void wallet_change_resource(wallet_t *wallet, const char
*resource, const int delta) {
  int success = 0;

  while (!success) {
    pthread_mutex_lock(&wallet->lock);
    success = _try_wallet_change_resource(wallet,
resource, delta);
    pthread_mutex_unlock(&wallet->lock);
  }
}
```

**Q:** What is this code doing with the CPU?

There are two ways of blocking:

1)

2)

**Q:** What if we need to block until a condition is met?

---

### Conditional Variables (CVs)

Conditional Variables provide a mechanism for the operating system to ________________________________________.

To use a conditional variable:

**1)** [Initialization]:

**2)** [Conditional Wait Loop]:

**3)** [Conditional Signal / Broadcast]:

## PNG File Format

When we discussed the PNG file format, there's a few bits of brilliance in the format that is definitely worth covering:

**1)** PNG Header:

```
0x 89 50 4e 47 0d 0a 1a 0a
```

`0x89`:

`0x50 4e 47`:

**2)** Endianness:

> **Section 2.1**: All integers that require more than one byte must be in network byte order: the most significant byte comes first, then the less significant bytes in descending order of significance (MSB LSB for two-byte integers, B3 B2 B1 B0 for four-byte integers)

"Big Endian":

"Little Endian":

**endianness/endianness.c**

```c
int i = 42;
unsigned char *s = (unsigned char *)&i;
printf("%d %d %d %d\n", s[0], s[1], s[2], s[3]);
```

What do we expect for the output for various endian systems?

---

## Act II: Computer Systems in the Cloud!

After the midterm exam, we will focus on transitioning from understanding systems concepts to applying these concepts in the cloud. There will be a few major changes:

1. We will switch to Python -- handling network communications in C is **a lot** of code. Python provides powerful, widely used libraries for us to build on.

2. We will begin to think of a complete system as just a "node" in a complex network. We will manage entire systems using Docker containers and Virtual Machines (VMs).

3. We will learn how to integrate back-end services with front-end interfaces. In MP6 -- released the week after the exam -- you will use your png-extract to create a web service for extracting hidden images from PNG files.