POSIX Threads

- Original Unix process: address space + one thread
  - When OS runs a program, it creates address space and one thread that runs the main function of the program

- A Unix process can create additional threads using the POSIX Threads (pthreads) system calls
  - POSIX Threads is a standard threads API for Unix variants
POSIX Threads API

- **pthread_create(thread, attr, start_routine, arg)**
  - Returns new thread ID in `thread`
  - Executes function specified by `start_routine(arg)`

- **pthread_exit(status)**
  - Terminates current thread, returns `status` to a joining thread

- **pthread_join(thread_id, status)**
  - Blocks thread until thread specified by `thread_id` terminates
  - Return value from `pthread_exit` is passed in `status`

- **pthread_yield()**
  - Thread gives up processor
POSIX Threads Mutex API

- Pthreads provides mutex, semaphores, and condition variables
- Mutex

```c
pthread_mutex_t mut = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_lock(&mut);
pthread_mutex_unlock(&mut);
```
POSIX Threads Synchronization API

- Pthreads provides mutex, semaphores, and condition variables
- Semaphores

```c
sem_t sem_name;
sem_init(&sem_name, 0, 0); /* 2nd arg is flag, 3rd arg is init value */
sem_wait(&sem_name);    /* down operation */
sem_post(&sem);         /* up operations */
```

- Condition variables

```c
pthread_cond_t cond = PTHREAD_COND_INITIALIZER;
pthread_cond_wait(&cond, &mut); /* mut is a lock */
pthread_cond_signal(&cond);
```
Say a thread wishes to wait until \( x > y \)

```c
int x,y; // shared variables
pthread_mutex_t mut = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t cond = PTHREAD_COND_INITIALIZER;

pthread_mutex_lock(&mut);
while (x <= y) {
    pthread_cond_wait(&cond, &mut);
}
/* operate on x and y */
pthread_mutex_unlock(&mut);
```

Another thread signals when \( x > y \)

```c
pthread_mutex_lock(&mut);
/* modify x and y */
if (x > y) pthread_cond_signal(&cond);
pthread_mutex_unlock(&mut);
```
Some POSIX Threads examples are available on the web site

- pthread-example.c
  - Several threads are created and run without synchronization
- pthread-example-sync.c
  - Several threads are created and run with synchronization
Summary

- Unix OS provides various system calls for managing processes
- POSIX threads is a standardized API for managing threads in Unix operating systems
- Next lecture: scheduling policies