Concurrent Programming (II)

Threads
- In a multithreaded program, all threads execute the same piece of code. They share the heap, but each thread has its own stack frame.
- main() function comprise a single, default thread. Threads other than the default one can be created by programmers.

Threads in C
- To create a multithreaded C program, you need to know these basic:
  - First, include the header file pthread.h
  - Second, a worker function should be defined. A worker function is a C routine that the thread will execute once it is created.

  - pthread (POSIX thread): threads that use the POSIX standard programming interface
    - #include <pthread.h>
    - Define a worker function: a C routine that the thread will execute once it is created
      - void *foo (void *args) () {}
    - Initialize pthread_attr_t: you can use NULL for the default values
      - pthread_attr_t attr;
      - pthread_attr_init (attr);
    - Create a thread
      - pthread_t thread;
      - pthread_create (&thread, &attr, foo, arg);
    - Thread management
      - pthread_join (thread, status); //suspend execution of the calling thread until the target thread terminates
      - pthread_exit (status); // terminates the calling thread
    - Compiling: using -pthread

- Third, the thread attribute should be initialized. You can use NULL for the default values. (attr: scope, joinable, size and etc.)
- Now you can create a thread by using the pthread_create function.
- After create threads, you can use pthread_join to suspend execution of the calling thread until the target thread terminates.
- Or you can use pthread_exit to terminates the calling thread.
- When compiling, remember to include the pthread library. You may no need this on Mac. If you use other machines, double check if the library path has been set already.
Show the `helloThreads.c`, and run the code. Comment out the join for loop, run the program again, and ask students why the results are different. [pthread_join will suspend the calling thread (main thread) until the target thread terminates.]

```c
#include <stdio.h>
#include <pthread.h>

#define NUM_THREADS 5
typedef struct {
  int id;
} threadInfo;

void *hello_thread(void *threadinfo) {
  threadInfo *ti = (threadInfo *) threadinfo;
  printf("Thread %d saying Hello!\n", ti->id);
  pthread_exit(NULL);
}

int main () {
  int i;
  threadInfo ti[NUM_THREADS];
  pthread_t thread[NUM_THREADS];

  // Set up the parameters for each thread
  for (i = 0; i < NUM_THREADS; i++)
    ti[i].id = i;

  // Get the threads going
  for (i = 0; i < NUM_THREADS; i++)
    pthread_create(&thread[i], NULL, hello_thread, &(ti[i]));

  // Join up with them. This will wait until they are done.
  for (i = 0; i < NUM_THREADS; i++)
    pthread_join(thread[i], NULL);

  return 0;
}
```

- with join for loop, the output is several "saying Hello!"
- without join for loop, the output is empty.