

⇒ C++11

⇒ tomorrow → Friday  
more examples  
in depth

SE3910 – REAL TIME SYSTEMS

⇒ Design

Designing Multithreaded Software For the Beaglebone

⇒ C++11 threads

- How do you start designing a multithreaded application in C using POSIX?

QUESTION

⇒ pthread\_create

pthread\_create

⇒ spawns a new thread.

SE3910 REAL TIME SYSTEMS



⇒ Must have a single static function pointer...

# POSIX THREADS (PTHREADS) INTERFACE

- *Pthreads*: Standard interface for ~60 functions that manipulate threads from C programs
  - Creating and reaping threads
    - pthread\_create, pthread\_join
  - Determining your thread ID
    - pthread\_self
  - Terminating threads
    - pthread\_cancel, pthread\_exit
    - exit [terminates all threads], return [terminates current thread]
  - Synchronizing access to shared variables
    - pthread\_mutex\_init, pthread\_mutex\_[un]lock
    - pthread\_cond\_init, pthread\_cond\_[timed]wait

*Waits for a given thread to terminate*

*Mutex/Synchronization*



# THE PTHREADS "HELLO, WORLD" PROGRAM

```

/*
 * hello.c - Pthreads "hello, world" program
 */
#include "csapp.h"

void *howdy(void *vargp);

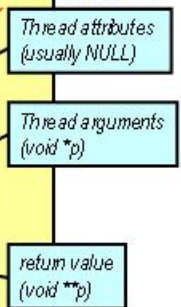
int main() {
    pthread_t tid;

    Pthread_create(&tid, NULL, howdy, NULL);
    Pthread_join(tid, NULL);
    exit(0);
}

/* thread routine */
void *howdy(void *vargp) {
    printf("Hello, world!\n");
    return NULL;
}

```

*Single function*

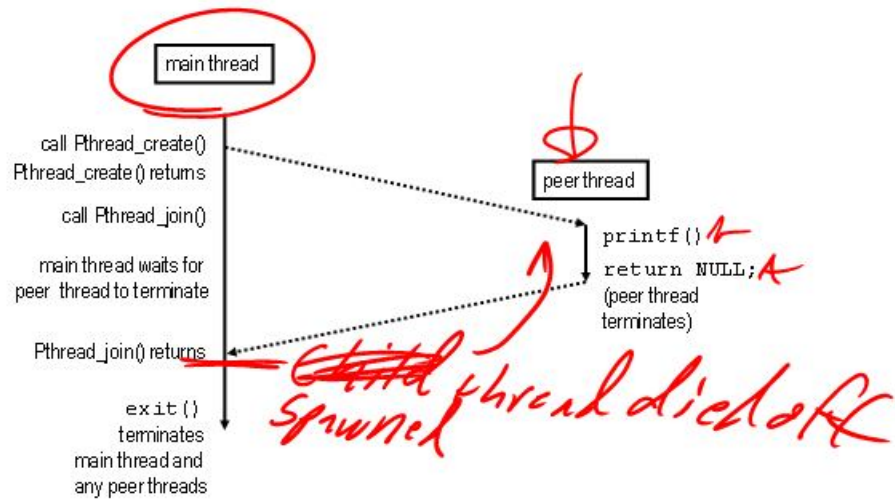


*Function that is executed...*

*Wait for thread to terminate.*



EXECUTION OF THREADED  
"HELLO, WORLD"



C++ POP QUIZ

- Explain to me what a friend function is.



## FRIEND FUNCTION

- A friend function of a class is defined outside that class' scope but it has the right to access all private and protected members of the class. Even though the prototypes for friend functions appear in the class definition, friends are not member functions.

*Used to be a  
method to  
start another  
thread ...*

SE3910 REAL TIME SYSTEMS



## POP QUIZ #2

How do we use multithreading in Java?

SE3910 REAL TIME SYSTEMS



# C++ THREADS

- C++ 11 threads are the new kid on the block
- `std::thread` class now part of standard C++ library
- `std::thread` is an abstraction — maps to local platform threads
  - (POSIX, Windows, etc.)

SE3910 REAL TIME SYSTEMS



## C++11 THREAD EXAMPLE

```
#include <thread>
#include <iostream>

void func()
{
    std::cout << "**Inside thread "
              << std::this_thread::get_id() << "!" << std::endl;
}

int main()
{
    std::thread t( func );
    t.join();
    return 0;
}
```

Static function  
Not a member of  
class

A simple function for thread to do...

class

Create & schedule thread to execute

function to execute

Wait for thread to finish...

pthread\_create(&thread,  
0, func, NULL);

## C++11 THREAD CONSTRUCTORS

- class thread
- {
- thread(); // creates new thread object that does \*not\* represent a thread (i.e. not joinable)
- thread(std::Function&&f, Args&&... args ); // creates new thread to execute f
- thread(thread&&other); // \*move\* constructor
  
- thread(thread&other); // \*copy\* constructor --- not available