

# Project 3: 多线程编程

**Chentao Wu 吴晨涛**

Professor

Dept. of CSE, SJTU

wuct@cs.sjtu.edu.cn

# 课程目标

- 使用 Pthreads, java 线程 api 设计多线程应用程序

# Pthreads Api

- Pthreads 指的是 POSIX 标准，该标准定义了用于线程创建和同步的 API。POSIX 只定义了线程行为的规范，不包含实现。内核设计者可以按照 POSIX 标准自行实现相应的 API。

- ```
int pthread_create(pthread_t *thread,  
                  const pthread_attr_t *attr,  
                  void *(*start_routine) (void *),  
                  void *arg);
```

第一个参数是线程描述符，表示创建的线程。第二个参数为创建的线程属性，本次作业我们直接用 `pthread_attr_init` 初始化为默认属性，我们将在下一次课程设计中使用线程属性。第三个参数是一个函数指针，指明新建线程需要执行的函数。第四个参数是传递给 `start_routine` 函数的实参。

如果成功创建线程，`pthread_create` 返回 0，反之返回非 0。

# Pthreads Api

- `void pthread_exit(void *retval);`

在线程执行过程中，我们可以手动调用 `pthread_exit` 来结束线程。

参数 `retval` 是 `void*` 类型的指针，可以指向任何类型的数据，它指向的数据将作为线程退出时的返回值。如果线程不需要返回任何数据，将 `retval` 参数置为 `NULL` 即可。

- `int pthread_join(pthread_t thread, void ** retval);`

`pthread_join` 将阻塞主线程并等待子线程终止，我们可以使用 `pthread_join` 等待子线程并获取子线程的返回值。如果创建子线程后没有调用 `pthread_join`，子线程的资源将不会被回收，并最终成为僵尸线程。我们也可以调用 `pthread_detach` 将子线程分离，这样子线程结束时会自行释放资源。

第一个参数为等待的子线程的线程描述符。第二个参数参数表示接收到的返回值，如果没有返回值或不需要返回值，可以设置为 `NULL`。

# 作业及评分

自行阅读课本第四章的 Programming Projects 部分，并完成以下四个任务，完成后共计11分。

- （课本习题 2分）使用 `pthread_create` 创建11个子线程，其中1个子线程校验数独中的所有行，1个子线程校验数独中的所有列，9个子线程分别校验一个3x3的区域。使用一个对所有线程可见的 `int` 类型数组来为主线程传递校验结果。
- （课本习题 3分）使用 `pthread_create` 创建3个子线程，其中2个子线程做一半数组的排序，第3个线程使用 **Mergesort** 算法将两个有序的子数组合并为一个有序的数组。
- （课本习题 4分）学习并使用 java 的 `fork-join api` 来实现两种并行排序算法 Quicksort 和 Mergesort。

注：可以使用 `sudo apt install openjdk-8-jdk` 安装 java8，并使用 `javac Quicksort` 和 `javac Mergesort` 来编译你写好的代码文件。最后使用 `java Quicksort` 和 `java Mergesort` 来运行代码。

- （报告 2分）做一个简单的报告解释你的代码，报告建议不超过2页（防内卷）。