

# Operating System CH4

Susie Glitter

2025 年 7 月 7 日

注：本次实验使用了 VMware 中的 ubuntu-16.04.6-desktop

## 1 任务一：多线程数独校验

创建了二十七个线程，分别进行九行九列九宫的判断，使用类似掩码的方法判断是否重复，最后通过传入的地址向主程序报告判断结果，使用pthread\_join()等待所有线程输入完毕后进行展示。结果如下：其中对

```
gg@ubuntu:~/Desktop/final-src-osc10e/ch4/sudoku$ ./sudoku <in
6 2 4 5 3 9 1 8 7
5 1 9 7 2 8 6 3 4
8 3 7 8 1 4 2 9 5
1 4 3 8 6 5 7 2 9
9 5 8 2 4 7 3 6 1
7 6 2 3 9 1 4 5 8
3 7 1 9 5 6 8 4 2
4 9 6 1 8 2 5 9 3
2 8 5 4 7 3 9 1 6
```

有问题的行、列、宫进行了黄色标注，若某一格所在的行、列、宫全部有误，则很有可能是这一格的数字有误，在这里使用红色进行标注

## 2 任务二：多线程归并排序

使用归并排序进行分治排序，对于每一个子任务，创建三个线程，前两个分别递归地对左右两边进行归并排序，第三个线程等待前两个排序完成后进行归并，使用到一个辅助数组，由于同时运行的线程排序部分不重叠，因此统一申请并且传入使用。

```
gg@ubuntu:~/Desktop/final-src-osc10e/ch4/mergesort$ ./mergesort <in
6 5 9 2 3 1 0 4 8 7
0 1 2 3 4 5 6 7 8 9
```

## 3 任务三：Java 多线程归并排序与快速排序

### 3.1 Java 多线程

Java 中使用ForkJoinPool实现 pthread,使用invoke()与invokeAll()创建新的线程执行任务，任务继承自RecursiveAction类，通过重写初始化函数实现参数的传入，通过重载其compute()函数定义线程所执行的任务。即可实现多线程编程。

### 3.2 归并排序

类似 c 语言实现的归并排序，使用三个线程分别对左右进行排序后进行归并。使用invokeAll()可以同时执行多个线程的任务并且进行等待，相比于 c 语言的实现，不需要额外使用pthread\_join()进行等待，更加方便，但是不适用于非立即进入等待的情况。

```
gg@ubuntu:~/Desktop/final-src-osc10e/ch4/java/mergesort$ java MergeSort <in
6 5 0 9 2 7 1 8 4 3
0 1 2 3 4 5 6 7 8 9
```

### 3.3 快速排序

快速排序也同样可以使用分治，使用多线程进行优化，不同的是快速排序先进行排序再进行分治，所以需要调换执行的顺序，先使用枢轴值将数列分为大小两部分，再递归地调用以枢轴值为分界的两部分分别进行排序。

```
gg@ubuntu:~/Desktop/final-src-osc10e/ch4/java/quicksort$ java QuickSort <in
6 5 0 9 2 7 1 8 4 3
0 1 2 3 4 5 6 7 8 9
```