

Report-Project3

付昊源

517021910753

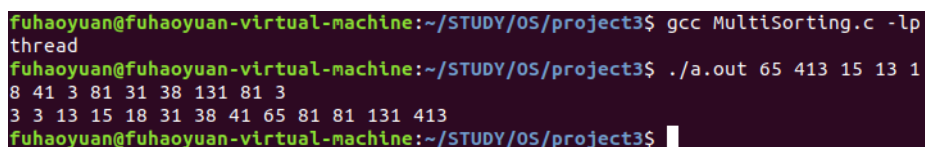
2019 年 6 月 11 日

1 Multithreaded Sorting Application

1.1 程序设计思想

这一 project 是让我们使用多线程对输入的数组进行排序。我的设计是先通过两个线程分别使用遍历排序法对前一半和后一半的数组数据进行排序，再通过第三个线程将这两个线程归并起来，从而完成整个数组的排序。

1.2 运行结果截图



```
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$ gcc MultiSorting.c -lpthread
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$ ./a.out 65 413 15 13 1 8 41 3 81 31 38 131 81 3
3 3 13 15 18 31 38 41 65 81 81 131 413
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$
```

图 1: Multithread Sorting

1.3 核心数据结构及代码解释

需要排序的数据通过命令行传入数组 array，array 使用 malloc 函数动态申请的空间，做到了空间的合理利用。之后使用三个线程，分别为 tid1，tid2 及 tid3，前两者用于排序前一半和后一半数据，最后一个线程用于归并数据。

首先需要对变量进行初始化，然后调用 pthread_create 函数，使 tid1 和 tid2 执行 sorting 函数，对数据进行排序。在 tid1 和 tid2 结束之前，主

函数中的 `pthread_join` 函数会等待他们的结束，再创建 `tid3`，执行 `merging` 函数将数组归并起来。

2 Fork-Join Sorting Application

2.1 程序设计思想

2.1.1 Quicksort

Quicksort 排序方式的基本思想是随机选取数组中的一个数据作为“中轴”（在我的设计中即选数组的第一个元素为“中轴”），遍历数组中其余的每一个元素，若比中轴小，则放到左侧数组 (`larray`) 中，否则放入右侧数组 (`rarray`) 中。阈值设为 2，即传递到线程中的数组个数小于 2 时，才对其进行排序，否则继续调用两个线程，分别对 `larray` 和 `rarray` 进行 Quicksort。

2.1.2 Mergesort

Mergesort 排序方式的基本思想是 `divide-and-conquer` 算法，根据主定理公式，它的时间复杂度是 $O(n\log n)$ ，也是所有排序算法中时间复杂度最低的。每次排序，它会首先检验数组个数是否低于阈值（在我的设计中为 2），低于阈值则直接排序，大于阈值则将数组分为左右两部分再递归调用本身。等左右两部分数组排好后，它将把两个数组归并到一起，完成排序。

2.2 运行结果截图

2.2.1 Quicksort

```
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$ gcc Quicksort.c -lpthread
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$ ./a.out 354 13 351 351
35 384 97 78 43 87
13 43 78 87 97 351 354 384 35135
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$
```

图 2: Quick Sort

2.2.2 Mergesort

```
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$ gcc Mergesort.c -lpthread
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$ ./a.out 34 38 78 1 456
56 78 12 67
1 12 34 38 56 67 78 78 456
fuhaoyuan@fuhaoyuan-virtual-machine:~/STUDY/OS/project3$
```

图 3: Merge Sort

2.3 核心数据结构及代码解释

2.3.1 Quicksort

为了便于传参，我在 Quicksort.c 中定义了一个结构体 parameters，包含待排序的数组的首地址 start 和数组中元素个数 cnt。主函数中，首先从命令行中读取待排序数据并存入以 start 为首的地址中，然后新建一个线程调用 sorting 函数。

在 sorting 函数中，即线程的工作过程中，首先判断待排序数组的个数是否小于二。个数由 parameters 结构体中的 cnt 作为参数传递。若个数小于 2 则直接排序，个数大于 2，以首地址的数据作为中轴，分别形成 larray 和 rarray，并调用两个线程分别对其进行 sorting 函数的执行。pthread_join 后，再将 larray、rarray 和中轴数分别复制到原数组的对应位置，排序结束。

2.3.2 Mergesort

与 Quicksort 类似，在 Mergesort.c 中我也定义了同样的结构体 parameters，主函数中的操作也与 Quicksort 类似，在此不再赘述。

在 sorting 函数中，我们首先检验数组的元素个数。高于阈值时，将数组均分为左右两部分，分别存入两个 parameters 变量 para1 和 para2 中，并调用两个线程对其排序。在 pthread_join 后，在这里我们调用第三个线程，对数组进行归并，归并时只需要归并前一半数组和后一半数组，即可完成排序。