

# 一、作业名称

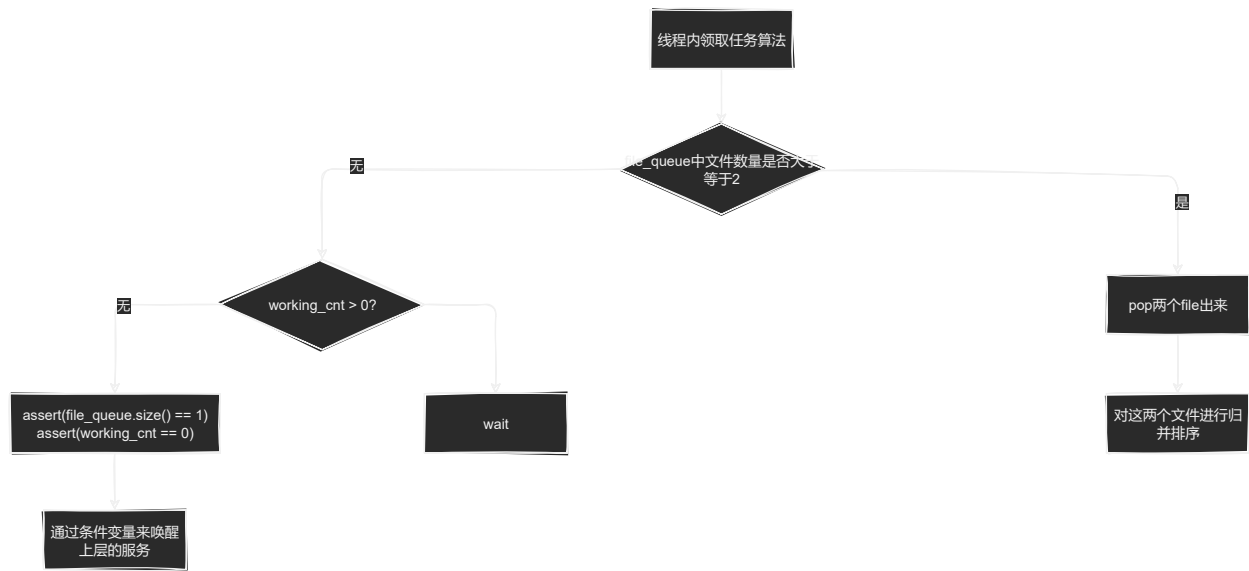
归并排序

# 二、作业要求

- 1. 给定一个目录，其中包含大量的普通文件
- 2. 每个文件中又包含了大量的数据
- 3. 每个数据都是64位的有符号数
- 4. 文件/数据格式可自由设定，数据随机生成

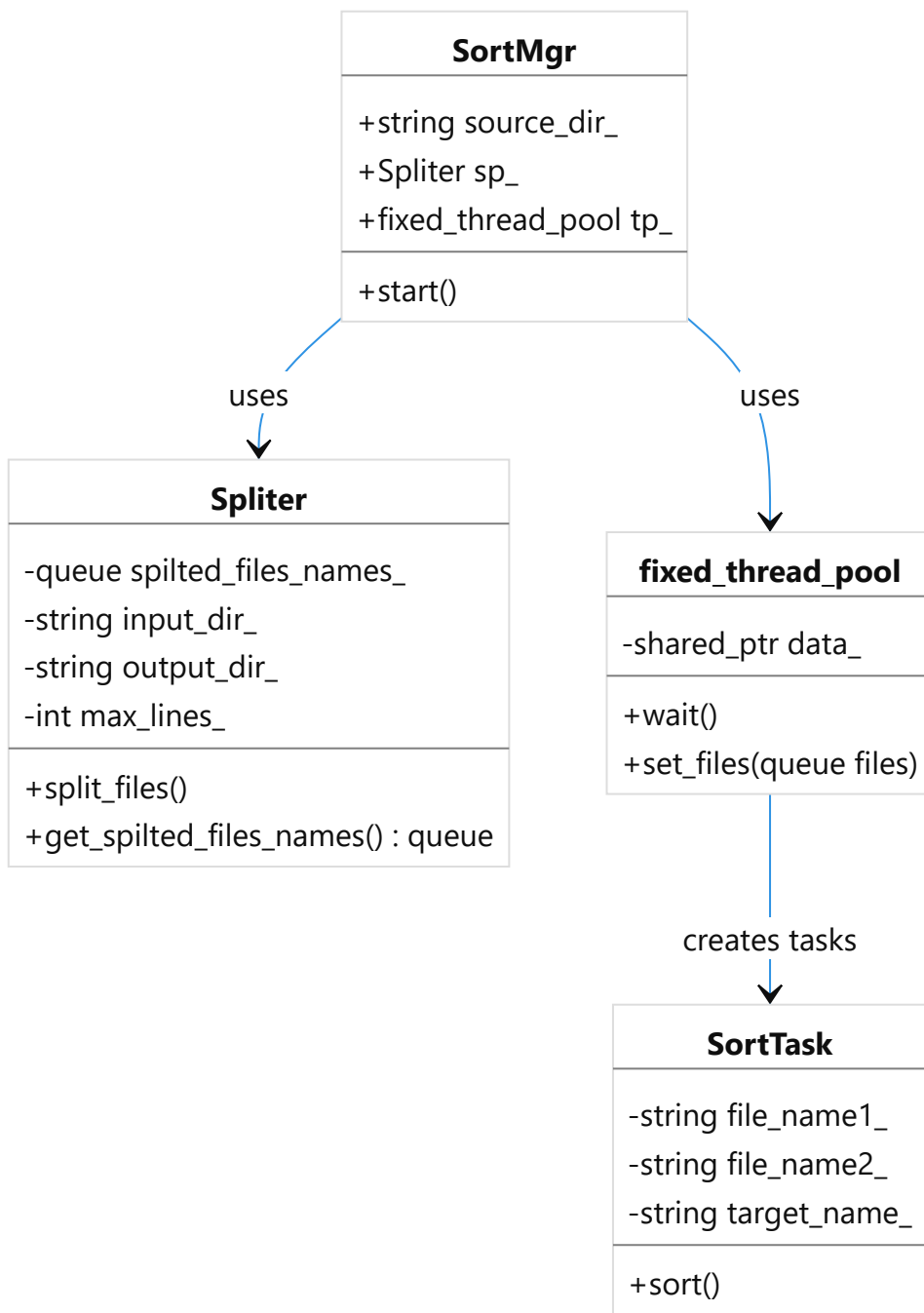
# 三、设计与实现

- 1. 先将目录中的文件分块，将文件分成最大4kb的trunk，并保存到数组中
- 2. 对trunk块内部进行排序，使其有序
- 3. 对文件进行归并排序



# 组件描述

- 1. **SortMgr**: 这是排序管理器，负责整个排序流程的协调。它使用Spliter来拆分大文件，并使用fixed\_thread\_pool来处理排序任务。
- 2. **Spliter**: 负责将大文件拆分成更小的文件块，并对每个小文件进行排序。这有助于并行处理和减少单个文件处理的内存需求。
- 3. **SortTask**: 表示一个排序任务。它从fixed\_thread\_pool接收文件，并执行排序操作。
- 4. **fixed\_thread\_pool**: 管理一个线程池，用于并行执行排序任务。它分配任务给线程，并在所有任务完成后进行清理。
- 5.



## 工作流程

1. **初始化**: SortMgr初始化，设置源文件目录，并创建Spliter和fixed\_thread\_pool实例。
2. **文件拆分**: SortMgr调用Spliter来拆分大文件。Spliter读取源文件，将其拆分成多个小文件，并对每个小文件进行排序。
3. **任务分配**: 拆分后的文件被传递给fixed\_thread\_pool。fixed\_thread\_pool将这些文件作为排序任务分配给线程池中的线程。
4. **排序处理**: 每个线程使用SortTask来处理分配给它的排序任务。完成后，结果被存储或用于进一步的合并操作。
5. **完成处理**: 所有排序任务完成后，fixed\_thread\_pool确保所有线程都已正确完成，并进行必要的清理。

## 线程池和文件排序实现

线程池（`fixed_thread_pool`）和排序任务（`SortTask`）是多线程文件排序系统的关键组件。下面详细介绍这两个组件的实现。

### `fixed_thread_pool` 实现

`fixed_thread_pool` 类是一个线程池管理器，负责创建和管理线程，以及分配和执行任务。它的主要功能包括：

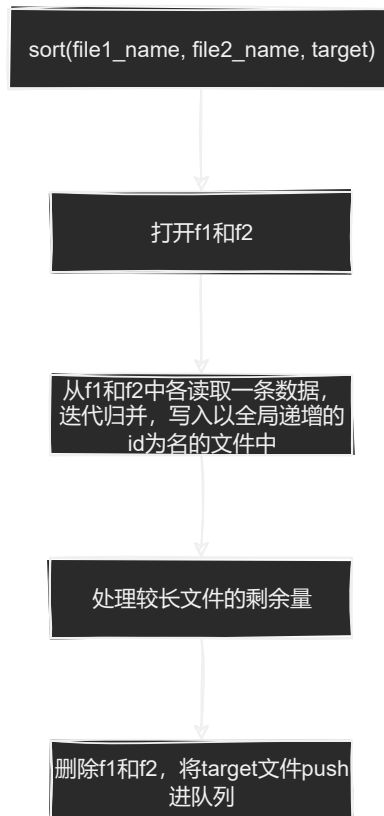
- 线程创建与管理：**
  - 在构造函数中，根据指定的线程数创建线程。
  - 每个线程执行一个循环，等待任务的到来。
- 任务分配：**
  - `set_files` 方法接收一个文件队列，这些文件是由 `Splitter` 拆分并排序后的小文件。
  - 文件被添加到内部任务队列中。
- 任务执行：**
  - 线程从任务队列中取出任务（在这里是文件名），并创建 `SortTask` 对象来执行排序任务。
  - 线程在完成任务后，会将结果文件名放回队列中，供后续的合并或进一步处理。
- 同步与等待：**
  - `wait` 方法用于等待所有任务完成。它会阻塞，直到所有排序任务都完成，并且任务队列中只剩下最后一个文件（最终结果）。
- 资源清理：**
  - 析构函数中确保所有线程都已经完成工作，并进行必要的资源清理。

### `SortTask` 实现

`SortTask` 类表示一个排序任务。它的主要职责是合并两个已排序的文件。实现细节如下：

- 构造函数：**
  - 接收两个要合并的文件名和一个目标文件名。
- 排序与合并：**
  - `sort` 方法打开两个输入文件和一个输出文件。
  - 使用归并排序的方式，从两个输入文件中读取数据，比较并将较小的元素写入输出文件。
  - 当一个文件读取完毕后，将另一个文件的剩余部分写入输出文件。
- 资源管理：**
  - 确保在排序完成后关闭所有文件，并删除原始文件，只保留合并后的文件。

排序算法如下：



这两个组件共同工作，实现了一个高效的多线程文件排序系统。`fixed_thread_pool` 管理线程和任务分配，而 `SortTask` 负责具体的排序逻辑。通过这种方式，系统能够充分利用多核处理器的能力，加快大文件的排序过程。

## 四、测试

测试了三个混乱文件的排序，SortMgr成功将其拆分成了小文件，并对其进行了归并排序。具体测试见homework3/test.cpp

```
[      OK ] MergeSort.SortMgrTest (2 ms)
[-----] 1 test from MergeSort (2 ms total)

[-----] Global test environment tear-down
[=====] 1 test from 1 test suite ran. (2 ms total)
[  PASSED ] 1 test.
```

## 五、附录