

Задание 2. Параллельный поиск в файлах.

Вы никогда не теряли информацию на своём компьютере? Не было такого, что вы точно знаете, что она где-то есть, но в каком файле — не помните? Вы помните только кусочек из текста, который вы набирали и, возможно, директорию с вашими проектами, в глубине которой хранился этот файл. Поможем сами себе, написав простой поисковик.

Программа должна получить из командной строки то, что вы будете искать (pattern) — обыкновенную текстовую строку, для простоты содержащую только символы из нижней половины таблицы ASCII (латинские буквы, цифры, знаки препинания и прочее). Кириллических и прочих символов мы искать не будем (если хотите, попробуйте, но я этого не требую). Другой аргумент — имя директории, с которой поиск начинается. Программа должна распознавать также число вычислительных потоков, которые будут заниматься операциями поиска.

Программа должна обрабатывать два флага.

- `-t#` — поиск в `#` потоков.
- `-n` — поиск только в указанной директории.

Эти флаги могут находиться в любом месте командной строки. При их отсутствии программа осуществляет поиск в 1 поток и в указанной директории и всех её поддиректориях.

Примеры вызова:

- `./psearch -t4 FILE /usr/include`
Искать образец `FILE` в директории `/usr/include` и глубже в 4 потока.
- `./psearch FILE`
Искать образец `FILE` в текущей директории и глубже в 1 поток.
- `./psearch FILE -t4`
Искать образец `FILE` в текущей директории и глубже в 4 потока.
- `./psearch FILE -n -t4`
Искать образец `FILE` только в текущей директории в 4 потока.

Для каждой найденной строки с образцом нужно вывести имя файла, в котором она была найдена, её номер и её саму. Не выводите одну и ту же строку несколько раз.

Максимальную оценку получает решение, которое быстрее всего выдаст результат на заданном наборе файлов и строке поиска. Желательно использовать хороший алгоритм поиска подстрок в строках. Идеально — построить конечный автомат, но, видимо, я много хочу...

Постарайтесь задействовать все потоки по-полной. В идеальном решении протаивания потоков не должно быть или оно должно быть как можно меньшим.

Последний срок сдачи задачи №2 — 15 ноября.

Задание 3. Лифт.

В корпусе прикладной математики имеется лифт. Похоже, что программе, которая его обслуживает, писал не первокурсник заборостроительного института. Нет, её писал третьеклассник из вспомогательной школы..

А как бы повёл себя идеальный лифт? Попробуем написать программу для его моделирования.

Итак, что у нас есть:

- Здание, состоящее из N этажей, от 1-го до N -го включительно.
- Количество лифтов K .
- Вместимость лифта C .
- Время перемещения лифта на один этаж T_{stage} , будем считать его постоянной и не зависящей от количества этажей, на которые перемещается лифт.
- Время открытия дверей T_{open} после прибытия на этаж. Только по прошествии этого времени после прибытия лифта в него можно зайти и из него можно выйти.
- Время нахождения лифта в пустом состоянии T_{idle} . Если лифт пуст, то через это время двери закроются автоматически.
- Время закрытия дверей T_{close} . После того, как последний зашедший в лифт нажмёт кнопку закрытия дверей, он отправляется именно через это время.
- Время T_{in} , требуемое для того, чтобы в лифт зашёл один человек.
- Время T_{out} , требуемое для того, чтобы из лифта вышел один человек. Ни при входе, ни при выходе паники не наблюдается, всё идёт строго по очереди.

Интересно, что в нашей модели лифта все его пользователи имеют одинаковый вес и вместимость лифта определяется сугубо количеством заполнивших его людей.

Так как случайных процессов у вас ещё не было, входные данные будут детерминированы, то есть заданы при запуске программы.

Вначале следует описание модели. $N K C T_{stage} T_{open} T_{idle} T_{close} T_{in} T_{out}$.

Далее i -я строчка описывает прибытие одного человека и состоит из следующих полей:

1. T_i — время прибытия. Записи подаются в порядке неубывания в формате HH:MM:SS.
2. S_{from} — этаж прибытия.
3. S_{to} — этаж назначения.

Каждый прибывший немедленно нажимает кнопку нужного направления (не так, как это делают в КПМ многие сотрудники, нажимая обе кнопки).

- Если имеется свободный лифт, то он направляется к прибывшему, если он стоял на этаже — двери сразу открываются.
- Если имеются несколько свободных лифтов, заявку начинает исполнять ровно один, время прибытия которого к пассажиру минимально. Направление движения этого лифта фиксируется и если поступают заявки с промежуточных этажей на это направление, он их исполняет. При этом может оказаться, что человек, вызвавший лифт, за время обслуживания заявки уедет на другом.
- Если свободных лифтов не оказывается, регистрируется заявка на этаж и направление.
- Проходящий мимо в нужном направлении лифт останавливается на этаже всегда, если имеется соответствующая заявка. В этом случае заявка всегда считается выполненной, даже если он оказывается, что лифт переполнен и в него никто не зашёл.
- При исполнении заявки в него заходит максимальное количество людей с этого этажа, движущихся в нужном направлении. На это требуется время $n \times T_{in}$, где n — количество человек, зашедших в лифт.
- Каждый человек, зайдя в лифт, нажимает кнопку нужного ему этажа, формируя лифту заявку на выход.
- Лифт останавливается только на тех этажах, для которых имеются заявки на вход или выход.
- После остановки лифта на этаже вначале выходят все, для которых этот этаж оказывается конечным, после чего заходят все, для которых этот этаж начальный.
- Если из лифта вышли все люди, лифт после выхода последнего пассажира переходит в пустое состояние и закрывает двери самостоятельно через T_{idle} . Если нет заявок, он остаётся в этом состоянии вечно.

Ваша задача — написать модель поведения лифта и его пассажиров. Каждый лифт должен быть описан отдельным вычислительным потоком (`thread`), остальные потоки можете распределять как вам заблагорассудится. Можете даже для каждого пассажира заводить свой поток. Можно завести поток диспетчера. Можно — потоки этажей. Фантазируйте. Учтите, что на каждом этаже кнопка направления обслуживается всеми лифтами одновременно, но не должно быть таких ситуаций, что на вызов к вам поспешили все свободные лифты.

Чего не должно быть:

- Лифт не должен упасть от того, что в него набьются все желающие попасть в 705-ю аудиторию.
- К одному пассажиру не должно придти два или более лифта.
- Проще сказать, что в вашей программе все взаимодействующие потоки не должны попасть в состояние гонок, в тупиковую ситуацию, все пассажиры должны быть обслужены (иначе они умрут от голода), должны быть соблюдены все инварианты по загрузке лифта.

Что нужно выводить: в программе, которую вы мне будете демонстрировать, все основные события. По окончании работы — вывести статистику: среднее время ожидания пассажиров, средняя загрузка лифта, словом, всё то, что вы посчитаете важным.

Примеры входных файлов — за вами. Для тестирования я буду использовать несколько своих.

Последний срок сдачи задачи №3 — 9 декабря.

Вопросы/ответы

:

Q: Если человек вызвал лифт, к нему начал движение свободный, но пассажира подобрал мимо идущий лифт, свободный лифт продолжает движение или останавливается?

A: Продолжает движение и останавливается на заказанном этаже.

Q: Если на этаж приехал лифт, начал набирать людей, и в этот момент на этаже появился другой мимоидущий — то люди начинают входить в оба и пассажиры делятся?

A: Заказавшие лифт пассажиры хотят уехать в своём направлении. Поэтому они заходят в любой идущий в нужном направлении лифт.

Q: Если на одном этаже вызвали лифт много людей и целесообразно приехать двум лифтам, почему не приехать двум? Почему если первую порцию заберет первый приехавший, остальные "сгорят"?

A: Лифт не знает про количество людей и на заказ на этаже всегда приезжает один лифт. Если люди не смогли уехать — они вызывают ещё раз.

Q: В лифт люди входят по справедливости, кто первой вызвал?

A: Пассажиры обслуживаются по дисциплине FIFO.

Q: Вася заказал вниз с 5 этажа — лифт поехал к нему. Пока лифт ехал, на 4 этаже Петя заказал вверх — ему нужно до 10 этажа - лифт его подберёт, проездом? Если да, то куда он поедет, подобрав Васю?

A: Как и все лифты, если текущее движение направление вверх — он его продолжит, даже если есть заказ вниз. Просто он не остановится у Васи, пока едет вверх.

Q: Мы на 5-том вызвали лифт вверх. На 9 был свободный лифт. Он начал движение в нашу сторону, вниз. На седьмом появился человек, который хочет на третий, вниз, направление с движением лифта совпадало, и лифт взял пассажира. Лифт доехал до нас, мы его вызывали. Но мы хотим вверх, а пассажир вниз. Что делать?

A: Если мы его вызвали, то он доедет до нас, но не возьмёт человека с 7-го этажа, так как у него в планах именно мы с движением вверх.