# Домашнее задание №2

## Постановка

Реализовать две утилиты: locate и updatedb, для индексации и поиска файлов. Утилита updatedb создает индекс файлов в заданном каталоге. Утилита updatedb создает индекс файлов и каталогов начиная с указанного корня. Индекс должен быть сохранен в один файл, имя которого указывается, как параметр при вызове утилиты. Вызов утилиты updatedb осуществляется командой:

```
updatedb --database-root PATH --output FILE  ,
```

где PATH - каталог для индексации (если указанного каталога не сущесвтует, программа должна выдавать соответствующую ошибку);

FILE - имя файла для сохранения индекса (файл может уже существовать, в этом случае он должен быть перезаписан);

Утилита locate ищет файлы и каталоги в индексе созданном утилитой updatedb по подстроке. Утилита locate выводит в стандартный поток вывода список полных путей файлов и каталогов, содержащих заданную подстроку в имени (именно в имени, а не в полном пути). Вызов утилиты locate осуществляется командой:

```
locate --databse FILE PATTERN      ,
```

где FILE - имя файла индекса (если файла не существует, должна быть выдана соответствующая ошибка);

PATTERN - подстрока для поиска;

## Требования к выполнению

При выполнении задания разрешено, и настоятельно рекомендуется использовать Boost (как минимум Boost.Filesystem, так же можете обратиться к Boost.Asio и Boost.Synchronization).

Для ускорения построения индекса используйте многопоточное исполнение. Если не уверены в эффективности алгоритма, проконсультируйтесь с преподавателем.

Индекс должен ускорять поиск файла (например, индекс, который только дублирует структуру каталогов бесполезен).

При индексации нужно игнорировать символьные ссылки, но жесткие ссылки могут присутствовать, учитывайте это и избегайте бесконечных циклов.

Оперативной памяти у вас достаточно, пока преподаватель практики не сказал обратного (поэтому, если сомневаетесь по-поводу алгоритма, проконсультируйтесь с преподавателем).

После поиска файла в индексе, убедитесь, что файл все еще присутствует в системе перед тем как выводить его (с момента построения индекса файлы могли быть удалены).

Поиск файла в индексе тоже может быть многопоточным, если это поможет.

**Требования к поставке**

Задание сдается в виде исходных кодов на языке C++ и Makefile, который их собирает. Разрешено использовать только компиляторы g++ и clang++. Разрешено использовать стандарты 1998, 2003 и 2011 годов. Makefile должен быть составлен таким образом, чтобы после вызова команды make в каталоге с Makefile появились два исполняемых файла updatedb и locate соответственно. Все исходные коды должны находиться в каталоге src, а побочные продукты компиляции (*.o) в каталоге obj, который должен создаваться при вызове make. Makefile должен содержать цель clean, которая удаляет все результаты компиляции (каталог obj и его содержимое, файлы updatedb и locate).

**Сроки сдачи**

Срок сдачи задания ограничен экзаменом. Учтите, что то, что вы сдали задание, не значит, что преподаватель его проверил и одобрил, постарайтесь сдать его заранее, чтобы избежать проблем перед экзаменом. К экзамену допускаются только студенты второе домашнее задание, которых проверено и одобрено преподавателем.

**Подсказки**

Все структуры данных дружат с многопточностью пока вы их не изменяете. Хештаблицы дружат с многопточностью, даже если вы их изменяете, если правильно их реализовать. std::unordered_[multi](map|set) - неправильная реализация хештаблиц. Древовидные структуры тяжело строить эффективно в несколько потоков.

Библиотека Boost представляет более богатую реализацию потоков и примитивов синхронизации (в том числе и shared_mutex, который появится в только в 14 стандарте).

Правильное использование мьютексов лучше, чем неправильное использование атомиков, не зависимо от производительности.