Тема: Многопоточность

Вариант: ξ

Задача: Реализовать web crawler: программу для перебора и скачивания интернет страниц. В условии задается адрес первой интернет страницы в формате:

<протокол>://<адрес>

Например:

https://habr.com/

Единственным обязательным для сдачи задачи является протокол "file://", позволяющий проводить тестирование на локальной базе данных:

file://input.html

Crawler должен:

1. Загрузить данную страницу на диск (в случае локального файла просто скопировать файл)
2. Найти в файле все ссылки на другие страницы. Ссылке искать в формате:

   <a href="<протокол>://<адрес>">

   ссылки в другом формате допускается проигнорировать

3. Произвести сканирование найденных страниц по тем же правилам (если они не были просканированы ранее)

4. Когда сканировать больше нечего, crawler должен вывести статистику: сколько времени он потратил на работу, сколько ссылок нашел.

Требования к решению:

1. Решение должно быть многопоточным: обработка графа страниц должно производится множеством рабочих потоков.

2. Должна присутствовать возможность варьирования множества рабочих потоков. Добавление второго рабочего потока (т.е. переход к многопоточности) не должно увеличивать суммарное время работы всей программы.
3. Необходимо найти предел количества рабочих потоков, после которого добавление новых потоков не уменьшает время работы приложения.

Дополнительная задача:

Поддержать другие протоколы кроме "file://" и протестировать решение на реальных страницах в интернете. Для скачивания страниц по сети следует использовать сторонние библиотеки, например: libcurl

Выполнение этой доп. задачи для получения автомата **не требуется**.

Формат входных данных:

<адрес стартовой страницы> <количество потоков работников>

Формат выходных данных:

<количество посещенных страниц> <общее потраченное время>

Тестовая локальная база страниц:

https://drive.google.com/file/d/1OzSUBeEMSNAazwgP397cjNKla0YsZjI1/