

# Домашнее задание 10. P2P network.

Срок сдачи: 15 мая

Прежде чем приступить к написанию, прочтите раздел **Замечания**.

## 1 Условие

Попробуем написать программу, предназначенной для p2p обмена книгами.

Итак, наша система будет состоять из: сервера (трекера) и клиентов (сиды, пиры). все они – отдельные приложения, общающиеся по сети.

Все клиенты хотят обладать некоторой особо ценной книгой.

Проблема в том, что в свободном доступе книги нигде нет, но есть сервер, который знает о том, какие главы уже есть у подключенных к нему клиентов (про еще не подключившихся, он, естественно, понятия не имеет).

В начале, книга имеется только у одного из клиентов. При подключении, он передает на сервер информацию о книге и ее разбиении на главы (torrent info; в простейшем варианте просто количество глав).

Среднестатистический клиент ведет себя примерно так:

- подключается к серверу
- сообщает о том, что у него уже есть (мог раньше уже скачать половину)
- запрашивает информацию об интересующей его книге
- для каждой главы, которой у него нет
  - запрашивает информацию о тех, у кого она есть
  - подключившись к одному из них, качает главу
  - не забывает сообщить на сервер, что у него появилась новая глава
- Когда вся книга целиком получена, клиент заканчивает работу (клиент, изначально обладавший книгой, естественно, так не поступает)

Сервер отвечает на запросы клиентов и обновляет индекс распределения глав по клиентам.

В итоге все клиенты должны иметь по собственному экземпляру книги.

Напишите, скрипт (на любом привычном для вас скриптовом языке), запускающий сервер и 5 клиентов, у одного из которых есть достаточно объемная интересная книга, которую всем срочно захотелось прочесть.

## 2 Замечания

1. Старайтесь избегать создания новых тредов через *new Thread*, предпочитайте, где это возможно, использование различных *ExecutorService*.
2. Не забудьте про логирование.
3. Клиент не должен иметь более одного открытого соединения с сервером, иначе серверу может поплохеть.
4. На сервере изменения индекса должны выполняться в фиксированном малом количестве потоков.
5. Было бы здорово поддерживать закачку нескольких различных книг одновременно.
6. Можете подумать над поведением клиента в случае, если качать ту или иную главу сейчас не с кого.
7. Вам могут пригодиться ваше воображение, *BlockingQueue*, *ScheduledExecutorService*, *ThreadPoolExecutor* и т.п.
8. Если вам кажется, что работа системы должна выглядеть как-то иначе, пишите – обсудим.

GLHF!