

Домашнее задание 27. P2P network.

Срок сдачи: 23:59 20 июня, 2015

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 (обычно), книга имеется только у одного из клиентов. При подключении, он передает на сервер информацию о книге (ее название) и ее разбиении на «главы» (torrent info; количество «глав» и размер каждой).

Под «главой» будем понимать $1/N$ часть файла, где N – заданная **на сервере** константа.

Среднестатистический клиент ведет себя примерно так:

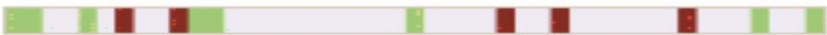
- подключается к серверу
- сообщает о том, что у него уже есть (мог раньше уже скачать половину)
- запрашивает информацию об интересующей его книге
- для каждой «главы», которой у него нет
 - запрашивает информацию о тех, у кого она есть
 - подключившись к одному из них, качает «главу»
 - не забывает сообщить на сервер, что у него появилась новая глава

Задача каждого клиента – скачать **все** книги, доступные для скачивания.

Клиент и сервер представляют из себя графические приложения.

При запуске клиент показывает окно со строкой для ввода IP адреса сервера и кнопкой Login.

После успешного соединения с сервером открывается основное окно программы, которое содержит следующие графические компоненты:

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; text-align: center;">Add file</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; text-align: center;">Stop seeding</div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px; text-align: center;">Quit</div>	Название файла	Скачено	Доступно
	Filename1.txt	100%	100%
	Filename2.txt	80%	56%
	...		
			

- кнопка Add file. При нажатии открывает дополнительное окно, в котором можно выбрать файл для раздачи
- кнопка Stop seeding. При нажатии выделенный файл перестает раздаваться (файл в таблице помечается красным)
- Кнопка Quit. Выходит из приложения.
- Выделенный файл в таблице помечается синим цветом.
- В нижней части находится прогресс-бар. Красным помечены «главы» недоступные для скачивания. Зеленым – скаченные.
- В таблице: Скачено – процент скаченных глав. Доступно – процент доступных для скачивания глав.

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

Его графический интерфейс выводит в виде таблице список клиентов. В нижней части выводятся имена файлов и «прогресс-бар» для файлов находящихся на выбранном клиенте.

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. Необходимо поддерживать загрузку нескольких различных книг одновременно.
6. Подумайте над поведением клиента в случае, если качать ту или иную главу сейчас не с кого.
7. Вам могут пригодиться ваше воображение, *Blocking Queue*, *ScheduledExecutorService*, *ThreadPoolExecutor* и т.п.

Если что-то не понятно – пишите ☺