

Файловый сервер (JAVA)

Требуется разработать приложение "Виртуальный файловый сервер/клиент". Программа должна состоять из двух частей: клиент и сервер.

Серверная часть

Сервер обрабатывает запросы клиентов для работы с виртуальной файловой системой (ВФС): создание, удаление, перемещение и копирование файлов и папок. Все клиенты должны работать с одной и той же ВФС, сервер должен обеспечивать синхронизацию работы. Когда один из клиентов делает изменения в ВФС (например создает новый файл), остальные клиенты должны получить уведомление об изменениях.

Обязательные требования: Сервер должен представлять из себя JAVA приложение.

Сервер должен поддерживать возможность одновременной работы как минимум со 100 клиентами.

Настройки сервера должны задаваться в конфигурационном файле.

Сервер не должен работать с реальной файловой системой (за исключением конфигурационного файла).

Протокол взаимодействия клиента и сервера – на усмотрение разработчика. Код программы должен быть документирован.

Желательные требования:

□ Использование UnitTest'ов.

Клиентская часть

Клиент представляет из себя консольное JAVA приложение, позволяющее работать с сервером из командной строки.

Команды работы с сервером:

`Connect` – соединение с сервером. После вызова команды сервер должен вернуть сообщение с количеством подключенных клиентов.

Формат: `connect server_name[:port] UserName`

Примечание: В системе не может быть два пользователя с одинаковым именем. Если пользователь пытается соединиться, указав имя, с которым кто-то соединился до него, сервер должен вернуть ошибку.

Примеры:

1. connect localhost Dennis 2. connect 127.0.0.1:123 Andrew

Quit – завершает работу с сервером MD –
создание директории.

Формат: MD [Drive:]Path

Примечание: команда не должна создавать промежуточных директорий.

Примеры:

1. MD c:\test – должна создать папку test на диске C.
2. MD temp – должна создать папку temp в текущей директории.
3. MD c:\test1\test2\ - должна создать папку test2, если папка test1 уже существует.

CD – смена текущей директории

Формат: CD [Drive:]Path *Примечание:* У каждого клиента должна быть своя текущая директория, т.е. вызов команды CD на одном из клиентов не должен влиять на работу других клиентов.

Примеры:

1. CD C: - делает диск C текущей директорией
2. CD C:\temp – делает папку temp текущей директорией
3. CD test1 – должен сделать текущей поддиректорию test1 текущей директории.

RD – удаление директории

Формат: RD [Drive:]Path

Примечание: команда не может удалить текущую директорию и директорию, в которой есть поддиректории.

Примеры:

1. RD test1 удаляет поддиректорию

DELTREE – удаление директории и всех ее поддиректорий *Формат:*

DELTREE [Drive:]Path

Примеры:

1. DELTREE c:\test

MF – создание файла

Формат: MF [[DRIVE:]Path]FileName

Примеры:

1. MF c:\test\1.txt – создает файл 1.txt в папку c:\test,
2. MF test\2.txt – создает файл 2.txt в поддиректории test текущей директории,
3. MF 3.txt – создает файл 3.txt в текущей директории.

DEL – удаление файла

Формат: DEL [[DRIVE:]Path]FileName

Примеры:

1. DEL 1.txt – удаляет файл 1.txt в текущей директории.

LOCK – запрещает удаление файла

Формат: LOCK [[DRIVE:]Path]FileName

Примечание: несколько пользователей могут заблокировать один и тот же файл, и его нельзя будет удалить до тех пор, пока все пользователи не снимут блокировку. Пользователь не может заблокировать файл 2 и более раз. Директорию нельзя удалить, если в ней есть заблокированные файлы. Файл нельзя удалить, если он заблокирован.

Примеры:

1. LOCK 1.txt – блокирует файл 1.txt в текущей директории.

UNLOCK – снимает запрет на удаление с файла

Формат: UNLOCK [[DRIVE:]Path]FileName

Примечание: пользователь может снять только свою блокировку.

Примеры:

1. UNLOCK 1.txt – разблокирует файл 1.txt в текущей директории.

COPY - копирует файл или директорию в другую директорию

Формат: COPY [drive:]source [drive:]destination

Примечание: Директория должна копироваться со всем содержимым. Destination не может указывать на имя файла. *Примеры:*

1. COPY test1 c:\temp – копирует поддиректорию test1 из текущей директории в директорию c:\temp

MOVE - перемещает файл или директорию в другую директорию

Формат: MOVE [drive:]source [drive:]destination

Примечание: Директория должна перемещаться со всем содержимым. Destination не может указывать на имя файла. Заблокированный файл не может быть перемещен. *Примеры:*

1. `MOVE test1 c:\temp` – копирует поддиректорию test1 из текущей директории в директорию c:\temp

`PRINT` – выводит на экран дерево каталогов

Формат: `PRINT`

Примеры:

1. `PRINT` – выводит на экран дерево каталогов, как показано ниже.

```
C:
|_DIR1
| | |_DIR2 | |
| | |_DIR4
| | | |_DIR3
| | | |_1.txt
| |
| | |_DIR5
| | | |_ 2.txt[LOCKED by Me]
| | | |_ 3.txt[LOCKED by TestUser]
```

Дополнительная информация:

1. По умолчанию в системе присутствует диск `C:`
2. Все команды регистро-независимы: т.е. команды `CD`, `cD`, `Cd`, `cd` – это одно и то же.
3. Нельзя удалять, перемещать заблокированные файлы, и директории, в которых есть такие файлы.
4. Нельзя удалять текущую директорию.
5. Любая команда, кроме `CD`, не должна менять текущую директорию.
6. Если в пути отсутствует префикс `C:\`, все действия должны выполняться в текущей директории.
7. В случае любой ошибки программа должна вывести сообщение на экран.
8. При выводе папки и файлы должны быть отсортированы лексикографически.
9. При выводе заблокированных файлов необходимо указать, что он заблокирован, и каким пользователем.
10. При изменении ВФС все клиенты (кроме того, кто сделал изменение) должны получить уведомление в виде текстового сообщения о том, кто и какое изменение сделал, например в таком виде: `UserName performs command: RD c:\test`, где `UserName` – это имя пользователя, заданное в команде `connect`.

