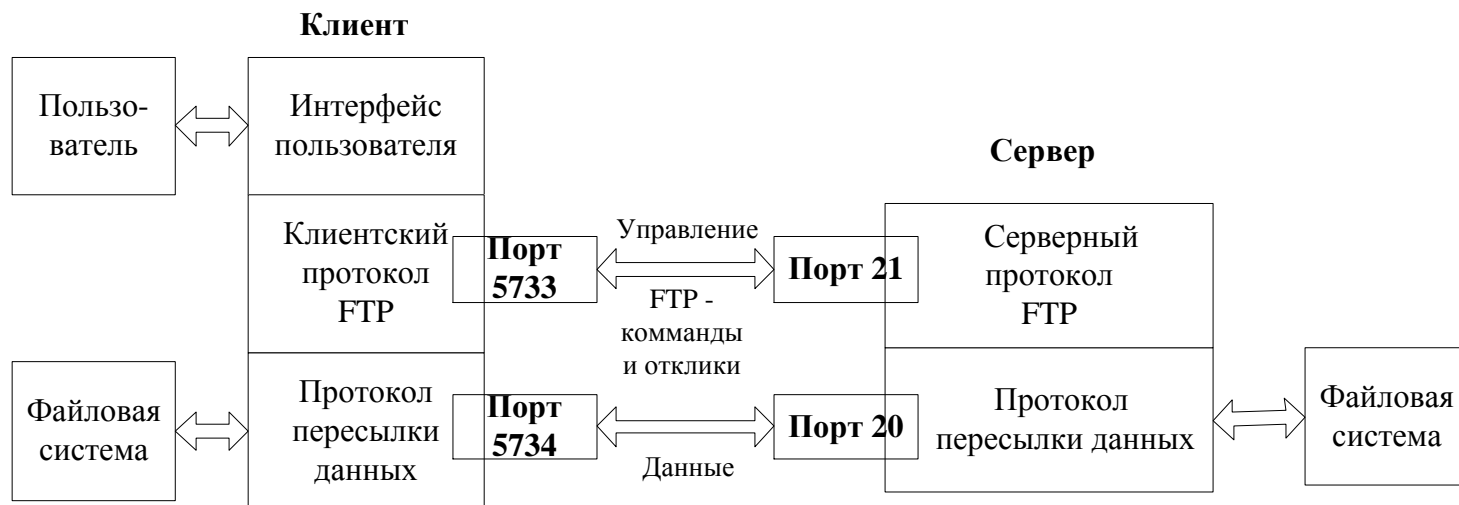


Протокол FTP

Протокол пересылки файлов (**File Transfer Protocol – FTP**, RFC-959) реализует обмен файлами между удаленными пользователями, поверх протокола транспортного уровня **TCP**.

- **Общий алгоритм работы FTP** на пользовательском уровне содержит несколько этапов:
 - Авторизация (ввод имени-идентификатора и пароля).
 - Выбор каталога.
 - Определение режима обмена (поблочный, поточный, ASCII или двоичный).
 - Выполнение команд обмена (get, ls или dir, del, put или send).
 - Завершение процедуры (quit или close).
- **Спецификация FTP определяет общедоступный или частный доступ к файлам:**
 - доступ к общедоступным файлам через анонимную регистрацию (логин – ANONYMOUS);
 - доступ к личным файлам, разрешенный только для определенных пользователей (требуется логин и пароль в режиме авторизации).
 - Протокол FTP использует **два порта**: один порт используется для **передачи команд** (например, 21 порт), второй порт используется для **передачи данных**.



Протокол FTP

Команды пересылки файлов позволяют:

- копировать одиночный файл между хостами;
- копировать несколько файлов между хостами;
- добавлять содержимое локального файла к удаленному файлу;
- копировать файл и добавлять к его имени номер для формирования уникального имени (например, файлы ежедневной регистрации получают имена log. 1, log.2 и т.д.).

Команды обслуживания файлов разрешают:

- просмотреть список файлов каталога;
- узнать текущий каталог и изменить его на другой;
- создавать и удалять каталоги;
- переименовывать или удалять файлы.

Управляющие команды служат для:

- идентификации пересылки файлов ASCII, EBCDIC или двоичных файлов;
- проверки структурирования файла (как последовательность байт или как последовательность записей);
- указания способа пересылки файла (например, как поток октетов).

Протокол FTP

```
csc.sibsutis> ftp
```

```
ftp> help
```

Commands may be abbreviated. Commands are:

!	?	acct	append	ascii	binary	bye	cd	debug
delete	dir	drive	exit	fcd	fdir	fpwd	get	help
iget	image	iput	lcd	ldir	lmkdir	local	login	lpwd
ls	mdelete	mget	mkdir	mput	option	parent	passive	put
pwd	quit	quote	rename	retrieve	rmdir	send	server	show
stat	store	take	tenex	tget	tput	type	user	verbose

Протокол FTP

Стоит отметить, что всякий раз, как только выполняется вызов на просмотр списка директории, копирование файла и т.д., то для пересылки этой информации открывается и используется соединение для данных. Процесс открытия этого соединения зависит от выбранного режима: *активный* или *пассивный*. Выбор режима определяется клиентом. В случае активного режима, клиент посылает команду – **PORT** *<IP адрес, порт>*, после чего серверная сторона иницирует соединение на указанную оконечную точку. В случае пассивного режима, клиент посылает команду **PASV**, ждет от сервера *<IP адрес, порт>* для инициации соединения на указанный адрес. После пересылки данных соединение по второму порту закрывается, а общение клиента с сервером продолжается по первому порту.

Пример пересылки файлов в активном режиме

```
csc-sibsutis>ftp
```

```
ftp>open sibsutis.ru
```

```
Connected to sibsutis.ru.
```

```
220 sibsutis.ru FTP server ready.
```

Далее проводим авторизацию (USER, PASS).

```
ftp>dir
```

```
-->PORT 128,36,4,22,10,53
```

Здесь IP-адрес 128.36.4.22 будет записан как 128,36,4,22, а порт 2613 — как 10,53

```
200 PORT command successful.
```

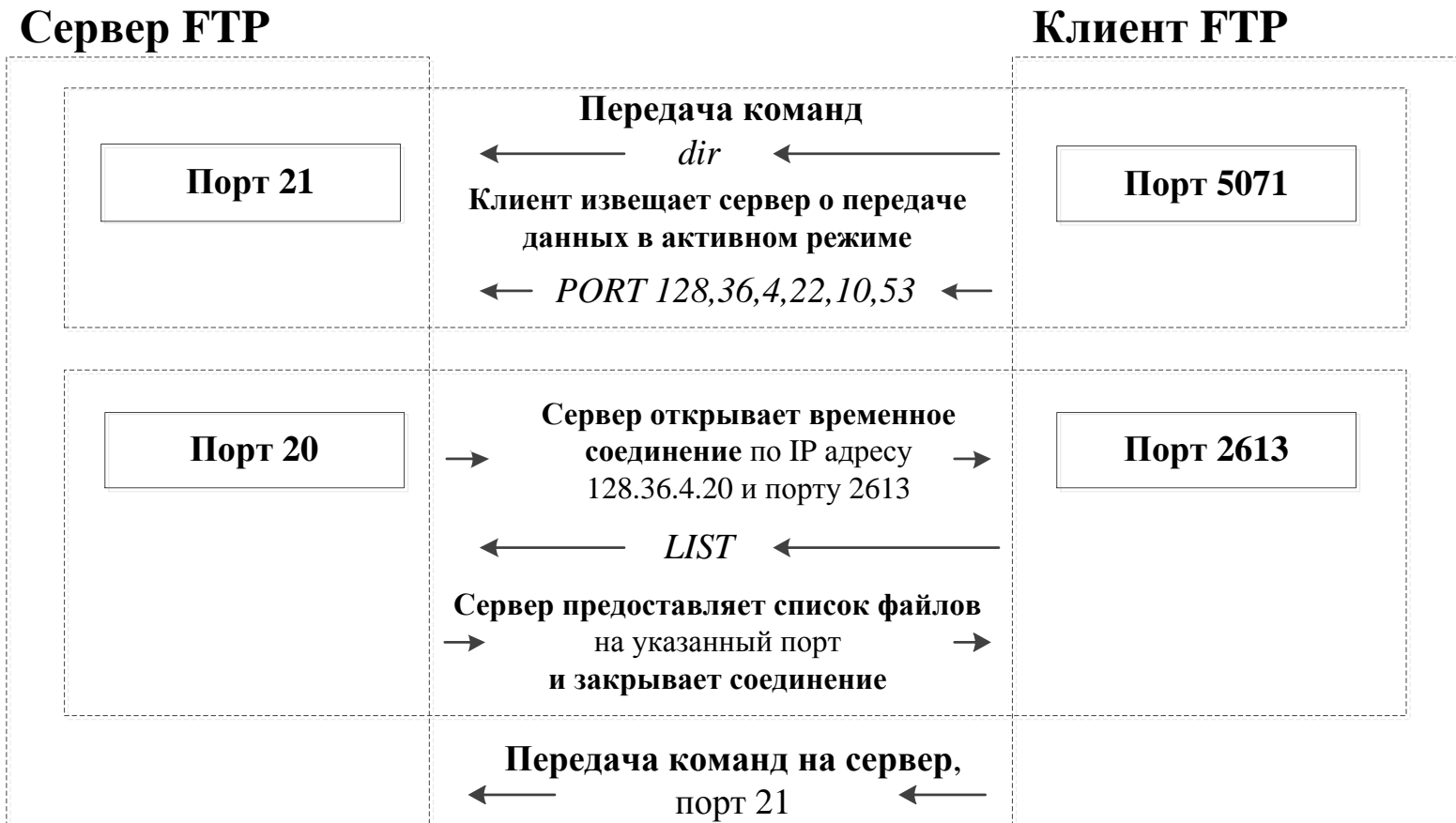
```
—> LIST
```

Далее сервер открывает соединение с объявленным клиентом портом и предоставляет список файлов.

*Сразу после пересылки списка файлов соединение данных будет закрыто. Затем, аналогично, мы можем получить , например, необходимый файл через команду **get** (*> get article*).*

Протокол FTP

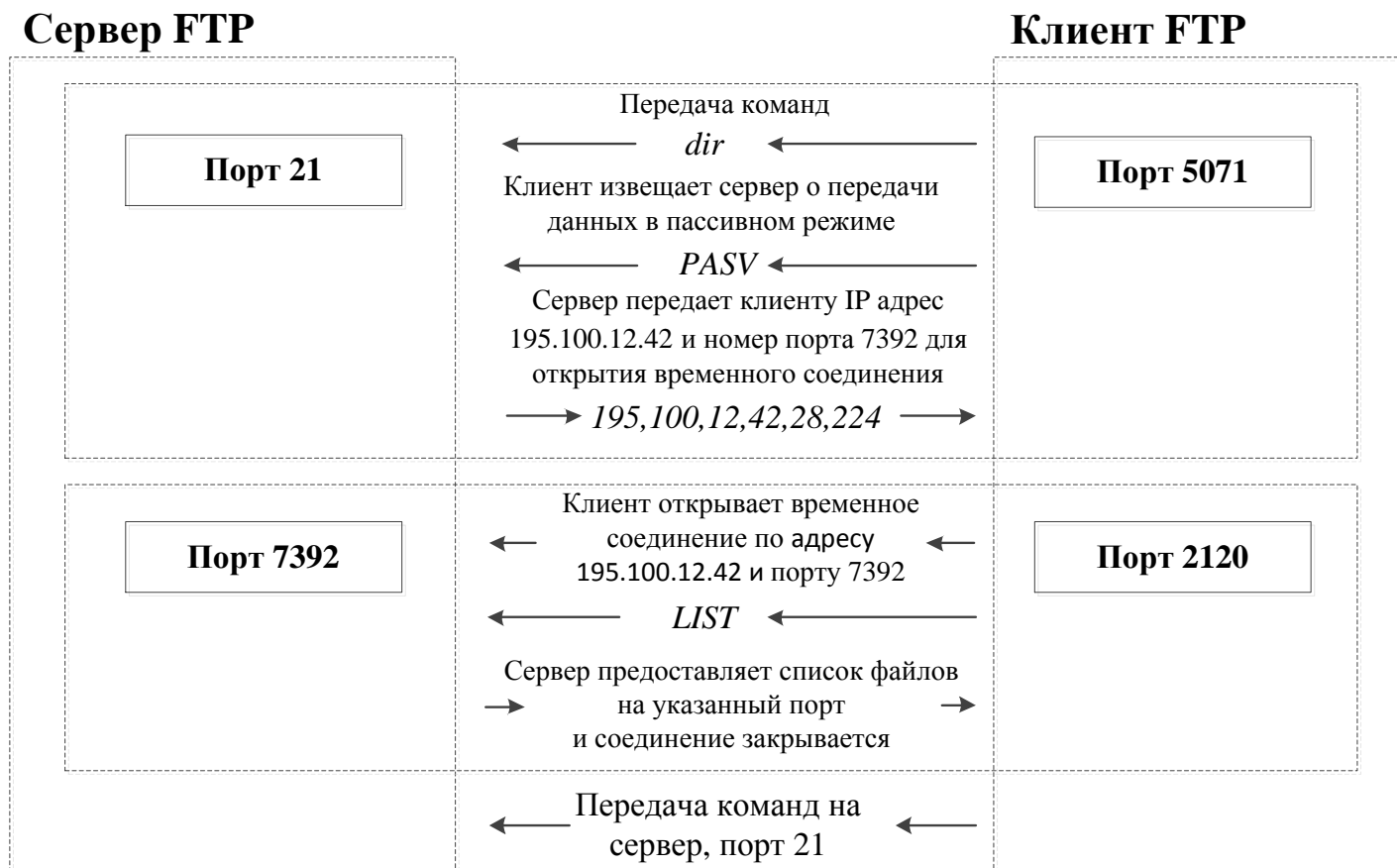
Пример пересылки файлов в активном режиме



Протокол FTP

Пассивный режим

- Как было сказано, существует альтернативный сценарий. Если клиент посылает команду *PASV*, сервер возвращает IPv4 адрес и номер порта (в числовой записи, аналогичной активному соединению) и переходит к прослушиванию установки соединения данных от клиента.
- Ранее преобладало использование команды *PORT*. Однако теперь клиент может послать команду *PASV* для пересылки файлов через простую систему защиты (firewall), которая не разрешает установку соединений из поступающих сообщений.



Протокол FTP

Типы данных, структуры файлов и методы пересылки

- На обоих концах соединения необходимо обеспечить единый формат для пересылаемых данных. Этот файл текстовый или двоичный? Он структурирован по записям или по блокам?
- Для описания формата пересылки используются три атрибута: *тип данных* (data type), *структура файла* (file structure) и *режим пересылки* (transmission mode). Допустимые значения этих атрибутов рассмотрены ниже. В общем случае применяются:
 - пересылка текста ASCII или двоичных данных;
 - неструктурированный файл, который рассматривается как последовательность байт;
 - режим пересылки рассматривает файл как поток байт.

Протокол FTP

Типы данных

- Файл может содержать текст ASCII, EBCDIC или двоичный образ данных (существует еще тип, называемый *локальным* или *логическим байтом* и применяемый для компьютеров с размером байта в 11 бит). Текстовый файл может содержать обычный текст или текст, форматированный для вывода на принтер.
- Типом данных по умолчанию является не распечатываемый текст ASCII (текст без управляющих символов форматирования). Тип данных может быть изменен стандартной командой *TYPE*, пересылаемой по управляющему соединению.

Пересылка текста ASCII

- Хотя текст ASCII является стандартным, компьютеры интерпретируют его по-разному из-за различия в кодах конца строки. Системы Unix используют для этого <LF>, компьютеры PC — <CR><LF>, а Macintosh — <CR>.

Пересылка текста EBCDIC

- Поддерживающие кодировку EBCDIC хосты обеспечивают весьма полезную команду пользовательского интерфейса, инициирующую пересылку по управляющему соединению команды *TYPE E*.

Пересылка двоичных данных

- С пересылки текстов ASCII легко переключиться на двоичный образ данных. В текстовом пользовательском интерфейсе для этого служит команда *binary*, а в графическом — командная кнопка *binary* (двоичные данные). Клиент меняет тип пересылаемых данных командой *TYPE I*, передаваемой по управляющему соединению.

Протокол FTP

Структуры файлов

- В FTP поддерживаются две структуры:
- *файловая структура*, соответствующая неструктурированному файлу, который рассматривается как последовательность байт;
- *структура записей*, которая применяется для файлов, состоящих из последовательности записей.
- Более распространена *файловая структура*, которая применяется по умолчанию. Перейти на *структуру записей* можно стандартной командой *STRU R*, пересылаемой по управляющему соединению.

Протокол FTP

Режимы пересылки

- Режим пересылки и структура файла определяют, как будут форматированы данные для обмена по соединению. Существуют три режима пересылки: *stream* (поток), *block* (блочный режим) и *compressed* (сжатые данные).
- В режиме потока и файловой структуры файл передается как поток байт. FTP возлагает на ТСПР обеспечение целостности данных и не включает в данные никаких заголовков или разделителей. Единственным способом указания на конец файла будет нормальное завершение соединения для данных.
- Для режима потока и структуры записей каждая запись отделяется 2-байтовым управляющим кодом конца записи (End Of Record — EOR), а конец файла отмечается символами конца файла (End Of File — EOF). EOR кодируется как X'FF 01, а EOF — X'FF02. Для последней записи файла EOR и EOF записываются как X'FF 03. Если файл содержит байт данных из одних единиц, то такой байт представляется при пересылке как X'FFFF.
- В блочном режиме файл пересылается как последовательность блоков данных. Каждый блок начинается 3-байтовым заголовком
- Режим сжатия данных используется крайне редко, поскольку обеспечивает очень неудачный метод архивирования, разрушающий последовательность повторяющихся байт. Обычно пользователю проще применить одну из более удачных программ сжатия, широкодоступных на современных компьютерах, и далее пересылать полученный архивный файл как двоичные данные.

Режим потока наиболее распространен и используется по умолчанию. Изменить его на блочный режим можно стандартной командой *MODE B*, пересылаемой по управляющему соединению.

Преимущество структуры записей или блочного режима, состоит в том, что будет явно отмечен конец файла и после завершения его пересылки можно сохранить соединение для данных, а, следовательно, использовать его для нескольких пересылок.

СПАСИБО ЗА ВНИМАНИЕ