

DRAFT
Version 0.2.0.1 [19.03.2004]



ПРОТОКОЛ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ ПОД КОДОВЫМ НАЗВАНИЕМ
OLYMP

Протокол: Илья Солдатенко

База данных: Евгений Шургалин

Март 2004

Центр Новых Информационных Технологий
Тверской государственный университет

(0822) 36-57-43

Оглавление

Оглавление	2
1. Каналы связи	4
2. Состояния сервера	5
3. Теория кодирования откликов сервера	7
4. Сводная таблица запросов и ответов	8
5. Запросы	9
5.1. Синтаксис запроса	9
5.2. Описание команд	9
LOGIN	9
LOGOUT	10
GET-TASKS.....	11
GET-COMPILERS	11
TASK.....	12
GTT.....	13
READY.....	14
DONE.....	15
START.....	16
DSQ.....	16
REACTIVATE.....	17
INIT.....	17
RATING-UPDATE.....	18
RATING.....	19
RATING-PART	19
6. Ответы	21
6.1. Синтаксис ответа	21
6.2. Описание ответов	21
1) 100 Wait For Beginning	21
2) 101 Program Accepted For Testing.....	21
3) 102 Free Tester Registered.....	21
4) 200 Logged In.....	21
5) 201 Logged Out	21
6) 202 Result Of Testing.....	21
7) 203 Tests And Tasks.....	22
8) 204 Result Accepted.....	23
9) 205 OK.....	23
10) 206 Full Rating.....	23
11) 207 Part Of Rating.....	23
12) 208 Rating Not Changed.....	24
13) 209 Olympiad Started	24
14) 210 Already Solved	24
15) 211 Tasks	24
16) 212 Compilers.....	24
17) 220 <имя сервера> at <имя хоста>.....	25
18) 300 Reactivate Tests And Tasks	25
19) 301 Program For Testing	25
20) 400 Forbidden.....	25
21) 401 Method Not Allowed	25
22) 402 Team Disqualified.....	25
23) 403 Length Required.....	25
24) 404 Bad Request	26
25) 405 Code-Team Disparity.....	26
26) 406 Comparison Standard Disparity	26
27) 407 Olympiad Currently Running.....	26
28) 408 Olympiad Is Not In Database.....	26
29) 500 Internal Server Error	26

30)	<i>501 Method Not Implemented</i>	26
31)	<i>502 OLYMP Version Not Supported</i>	26
6.	Структура базы данных	27

1. Каналы связи

Общение между клиентом и сервером происходит через виртуальный канал связи. Канал связи между клиентом и сервером устанавливается сразу после открытия сетевого соединения по инициативе клиента. Если канал связи не был открыт, то сетевое соединение закрывается сервером по тайм-ауту.

Каналы связи бывают следующих типов:

1. виртуальный канал для общения клиента-участника и сервера – КУ-канал
2. виртуальный канал для общения тестера и сервера – Т-канал
3. виртуальный канал для общения администратора и сервера – А-канал
4. виртуальный канал для общения рейтинг-сервера и сервера – Р-канал

Основная идея заключается в том, что каждому каналу соответствует свой круг клиентских запросов, которые могут быть приняты и обработаны сервером, только если они были посланы через соответствующий канал. Например, запустить олимпиаду можно только по А-каналу связи, а запросить тесты и решения – по Т-каналу. Даже административный канал связи не обладает всей полнотой запросов, так как это привело бы к излишней загрузке администратора ненужной функциональностью.

Однако запросы могут пересекаться, как например, запросить список компиляторов может как клиент-участник, так и тестер. Свободных запросов – то есть таких, которые можно отправлять вне какого-либо канала – не существует. Вернее есть единственный подобный запрос – запрос LOGIN, который как раз и служит для открытия канала передачи данных.

Канал связи устанавливается клиентом с помощью запроса серверу LOGIN. Параметр запроса указывает на тип создаваемого канала. Основными условиями открытия виртуального канала связи служат:

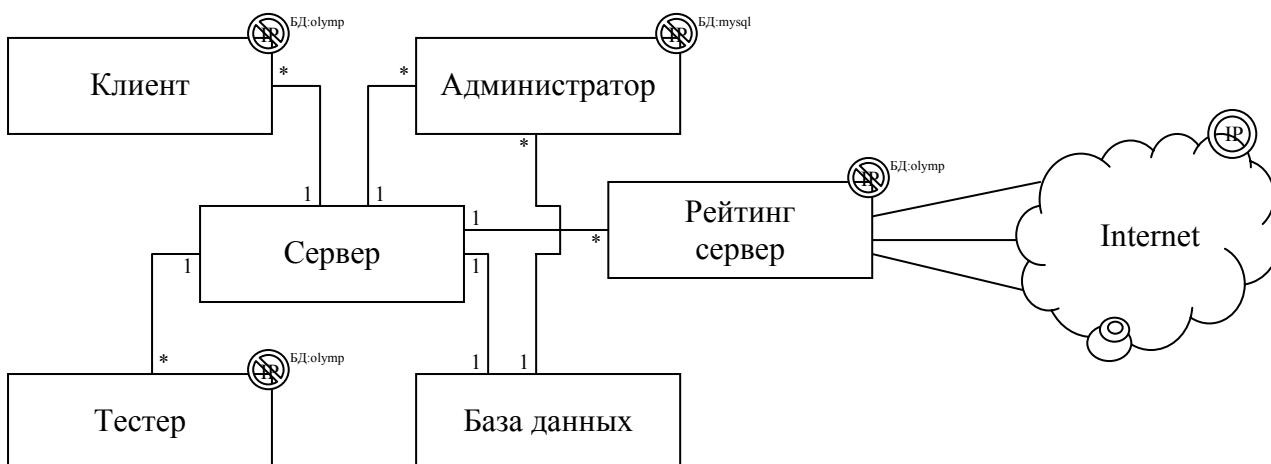
1. для КУ-каналов – клиент должен находиться на машине с IP адресом, зарегистрированным за одной из команд. IP адреса хранятся в базе данных и присваиваются каждой команде индивидуально. Помимо этого, заголовок Code запроса должен содержать кодовое слово, принадлежащее данной команде. В случае, если команда дисквалифицирована, сервер не открывает канал.
2. для Т-каналов – клиент должен находиться на машине с IP адресом, прописанном в базе данных, как IP адрес тестирующей машины.
3. для А-каналов - клиент должен находиться на машине с IP адресом, прописанном в базе данных, как IP адрес администраторской машины.

В случае, если клиент попытается сделать запрос, запрещенный в данном канале связи, сервер вернет ошибку несоответствия запроса каналу связи.

Канал связи может быть закрыт клиентом либо посредством запроса LOGOUT, либо при закрытии сетевого соединения.

2. Состояния сервера

Общую схему клиент-серверного взаимодействия можно описать следующей диаграммой:



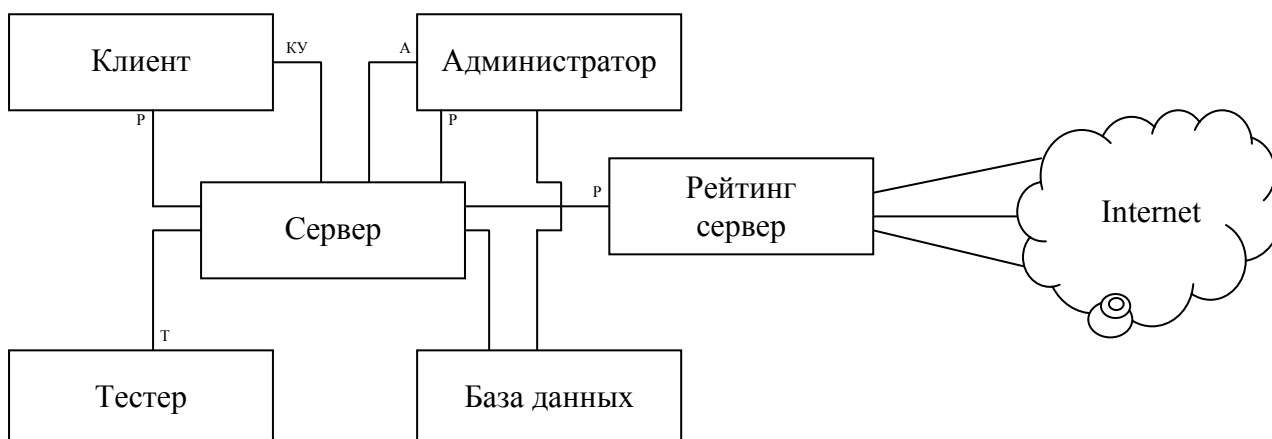
где:

1. – доступ разрешен с любого IP адреса
2. – доступ разрешен только с тех IP адресов, что прописаны в базе данных olymp
3. – доступ разрешен только с тех IP адресов, что прописаны в базе данных mysql

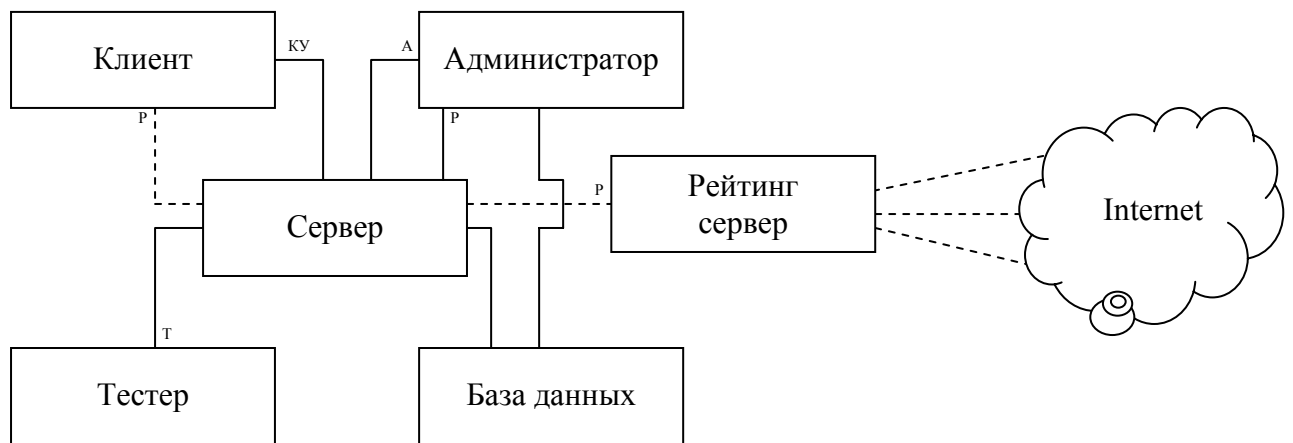
Таким образом, напрямую с базой данных могут общаться только сервер и администратор. Причем IP адрес администраторов прописан в системной базе данных mysql, что предотвращает несанкционированный доступ в БД администраторов с любой машины.

Сервер может быть в одном из следующих состояний:

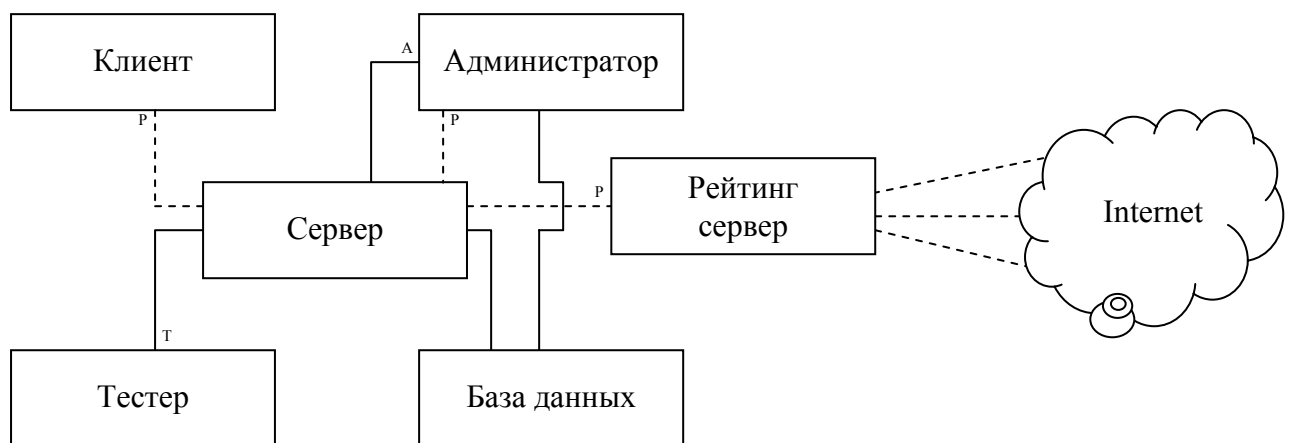
- 1) олимпиада идет (КУ, А, Т, Р – соответствующие каналы передачи данных)



2) олимпиада идет, рейтинг заморожен



3) олимпиада остановлена



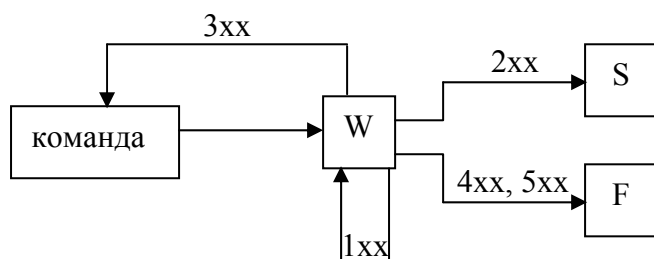
3. Теория кодирования откликов сервера

Код отклика сервера является трехзначным числом, в котором первая цифра идентифицирует тип отклика, а остальные две образуют порядковый номер отклика.

Существует пять значений для первой цифры кода отклика:

- 1yz – позитивный предварительный отклик:
Сюда относятся отклики двух типов:
 - запрос был принят, но его исполнение временно приостановлено в ожидании каких-либо действий от клиента. Клиент должен послать еще один запрос, указывающий, необходимо ли продолжать выполнение первоначальной команды или прервать его.
 - запрос принят и успешно выполнен, но прежде чем отсылать новый запрос, клиент должен дождаться еще одного отклика от сервера.
- 2yx – позитивный заключительный отклик
Запрос был успешно выполнен. Может быть инициирован новый запрос.
- 3yz – позитивный промежуточный отклик
Запрос был принят, но его исполнение временно приостановлено в ожидании дополнительной информации от клиента. Клиент должен послать следующий запрос с необходимой серверу информацией. Данный тип откликов используется в составных запросах, когда одна команда выполняется посредством некоторой цепочки запросов и ответов.
- 4yz – временно-отрицательный заключительный отклик
Запрос был отвергнут, и запрашиваемая команда, соответственно, не была выполнена. Тем не менее, ошибка, из-за которой запрос не был принят на исполнение, не является фатальной, и клиент может повторить запрос, изменив, быть может, некоторые его параметры или подождяв некоторое время. Точное значение “временности” зависит от конкретного запроса.
- 5yz – перманентно-отрицательный заключительный отклик
Запрос был отвергнут, и запрашиваемая команда, соответственно, не была выполнена. Клиент не должен посылать тот же самый запрос, так как его обработка невозможна из-за неустранимой ошибки (такой ошибки, которая не может быть устранена сервером самостоятельно, и сервер не имеет никакой дополнительной информации о том, когда и как она будет устранена).

Взаимоотношения откликов сервера можно изобразить на следующей диаграмме:



Здесь:

- команда – команда, посылаемая клиентом серверу на выполнение
- W – состояние ожидание отклика сервера
- S – успешное выполнение команды
- F – неуспешное выполнение команды

4. Сводная таблица запросов и ответов

Запросы	*	KY	A	T	P	Ответы
LOGIN	✓					100 Wait For Beginning
LOGOUT		✓	✓	✓	✓	101 Program Accepted For Testing
GET-TASKS		✓		✓		102 Free Tester Registered
GET-COMPILERS		✓		✓		200 Logged In
TASK		✓				201 Logged Out
GTT				✓		202 Result Of Testing
READY				✓		203 Tests And Tasks
DONE				✓		204 Result Accepted
START			✓			205 OK
STATUS-CHANGE			✓			206 Full Rating
DSQ			✓			207 Part Of Rating
REACTIVATE			✓			208 Rating Not Changed
INIT			✓			209 Olympiad Started
RATING-UPDATE			✓			210 Already Solved
RATING		✓	✓		✓	211 Tasks
RATING-PART		✓	✓		✓	212 Compilers
						220 <имя сервера> at <имя хоста>
						300 Reactivate Tests And Tasks
						301 Program For Testing
						400 Forbidden
						401 Method Not Allowed
						402 Team Disqualified
						403 Length Required
						404 Bad Request
						405 Code-Team Disparity
						406 Comparison Standard Disparity
						407 Olympiad Currently Running
						408 Olympiad Is Not In Database
						500 Internal Server Error
						501 Method Not Implemented
						502 OLYMP Version Not Supported

Расшифровка обозначений:

- * - запрос возможен вне какого-либо канала
- KY – запрос возможен в KY-канале
- A – запрос возможен в A-канале
- T – запрос возможен в T-канале
- P – запрос возможен в P-канале

Если запрос не соответствует каналу связи, сервер возвращает сообщение об ошибке 401 (*Method Not Allowed*). Если же запрос (кроме LOGIN) используется вне какого-либо канала, то сервер возвращает отклик 400 (*Forbidden*).

5. Запросы

5.1. Синтаксис запроса

Запрос имеет следующий синтаксис в БНФ-нотации:

```
запрос := команда [заголовки] [тело]
команда := <команда> [<SP> <параметр>] <SP> <протокол>/<версия>
<команда> := <КУ-запрос> | <Т-запрос> | <А-запрос> | <Ост. запросы>
<КУ-запрос> := TASK
<Т-запрос> := GTT | READY | DONE
<А-запрос> := START | STATUS-CHANGE | DSQ | REACTIVATE | INIT | RATING-
UPDATE
<Ост. запросы> := LOGIN | LOGOUT | GET-TASKS | RATING | RATING-PART
<параметр> := {для каждой команды могут быть свои}
<протокол> := {название протокола}
<версия> := {версия протокола}
заголовки := <CRLF> заголовок | заголовки
заголовок := <Имя>: <значение>
тело := <CRLF><CRLF> {текст, предусмотренный запросом}
```

Регистр команд, параметров, заголовков, а также названия протокола не важен. Тело запроса отделяется от заголовков двумя переносами строк. Команда представляет собой последовательность из латинских букв и дефиса. Порядок заголовков не имеет значения, имя заголовка отделяется от его значения двоеточием.

5.2. Описание команд

Во всех диаграммах состояния, приводимых для запросов, используются следующие обозначения:

- В – начало команды
- W – состояние ожидания клиентом ответа от сервера
- S – успешное завершение команды
- E – неуспешное завершение команды
- C – состояние ожидания сервером запроса от клиента. Встречается в изображении диаграмм составных команд

LOGIN

Назначение: открытие защищенного канала передачи данных.

Параметры:

- team** – открытие КУ-канала, канала между сервером и клиентом-участником олимпиады
- tester** – открытие Т-канала, канала между сервером и тестером
- admin** – открытие А-канала, канала между сервером и администратором
- rating** – открытие Р-канала, канала между сервером и рейтинг-сервером

Заголовки:

- Code – кодовое слово команды. Заголовок посылается клиентом только в том случае, если задан параметр team.

Допустимые каналы: вне какого-либо канала

Описание:

Общение с сервером может происходить только через защищенный канал передачи данных. Каналов связи бывает четыре – КУ-канал, Т-канал, А-канал и Р-канал (подробнее см. в разделе “Каналы связи”). Для установления канала связи служит запрос LOGIN, причем его параметр указывает на то, какой именно канал клиент хочет установить.

В случае если запрос производится с машины, IP адрес которой не прописан в базе данных, то сервер возвращает отклик 400 (*Forbidden*).

Если производится открытие КУ-канала с разрешенного IP адреса, но кодовое слово команды не соответствует кодовому слову той команды, что в действительности должна сидеть за данной машиной, то сервер возвращает ответ 405 (*Code-Team Disparity*).

Если открытие канала происходит успешно, то ответ сервера зависит от типа открываемого канала:

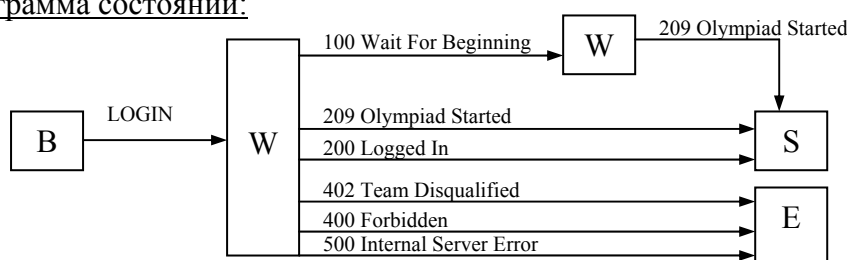
- если открыт КУ-канал, то:
 - если олимпиада еще не началась, сервер посылает клиенту ответ 100 (*Wait For Beginning*). Затем, когда олимпиада начинается, сервер посылает клиенту сообщение 209 (*Olympiad Started*)
 - если олимпиада началась, сервер посылает 209 (*Olympiad Started*)
- если открыт Т-канал, А-канал или Р-канал, то сервер посылает 200 (*Logged In*).

Следует заметить, что КУ-канал может быть успешно открыт только в том случае, если олимпиада уже идет, поэтому ответ сервера, завершающий команду открытия КУ-канала (209) отличается от ответа, завершающего открытие А-, Т- и Р-каналов (200). Если же клиент попытается отослать задачу при помощи запроса TASK еще до начала олимпиады, то сервер снова возвратит ответ 100 (*Wait For Beginning*).

Сетевое соединение без открытого виртуального канала не поддерживается сервером постоянно и по истечению некоторого времени обрывается.

Если делается попытка установить КУ-канал, а команда-участница является дисквалифицированной, то сервер возвращает соответствующий код ошибки (402).

Диаграмма состояний:



LOGOUT

Назначение: закрытие защищенного канала связи

Параметры: нет

Заголовки: нет

Допустимые каналы: любой канал

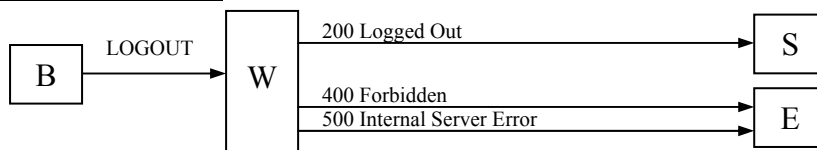
Описание:

Клиент может закрыть канал с сервером по своему желанию, послав ему запрос LOGOUT. Данный запрос не имеет ни параметров, ни заголовков. В случае успешного закрытия канала, сервер посылает ответ 201 (*Logged Out*) и закрывает сетевое соединение.

В случае, если клиент посылает данную команду при закрытом канале, сервер возвращает сообщение о несоответствии запроса каналу связи 400 (*Forbidden*).

Тем не менее, использование команды LOGOUT для закрытия канала передачи данных не является обязательным.

Диаграмма состояний:



GET-TASKS

Назначение: запрашивает у сервера номера и названия задач

Параметры: нет

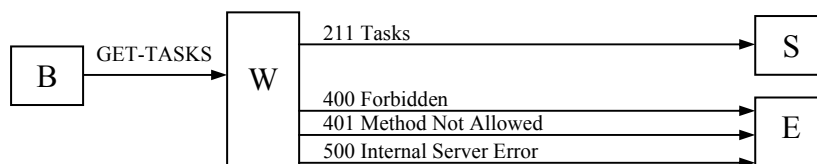
Заголовки: нет

Допустимые каналы: КУ, Т

Описание:

При помощи данной команды клиент может запросить у сервера номера и названия задач текущей олимпиады. В случае успешного выполнения сервер возвращает ответ 211 (*Tasks*), содержащий запрашиваемую информацию.

Диаграмма состояний:



GET-COMPILERS

Назначение: запрашивает у сервера допустимые в данной олимпиаде компиляторы

Параметры: нет

Заголовки: нет

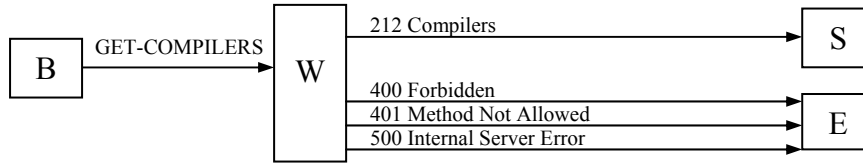
Допустимые каналы: КУ, Т

Описание:

При помощи данной команды клиент может запросить у сервера список компиляторов, допустимых в данной олимпиаде. Названия компиляторов передаются клиенту вместе с их унифицированными идентификаторами, необходимыми в запросе TASK и отклике сервера 301 Program For Testing. Каждый раз, когда тестер подключается к серверу, он должен отсылать этот запрос для согласования списков компиляторов, имеющихся у него и необходимых в данной олимпиаде. Если списки не согласуются, то тестер должен отключиться от сервера и просигнализировать об ошибке.

В случае успешного выполнения, сервер возвращает ответ 212 (*Compilers*), содержащий запрашиваемую информацию.

Диаграмма состояний:



TASK

Назначение: отсылает задачу серверу на проверку

Параметры:

- **force** – если программа, отсылаемая клиентом, соответствует задаче, которая уже была успешно решена командой ранее, то сервер возвращает ответ 210 (*Already Solved*) и не ставит ее в очередь на проверку, чтобы не нагружать тестеров. Если же клиент все же хочет проверить корректность данной программы, то он может указать в запросе параметр *force*, который заставит сервер протестировать данную задачу и вернуть результат. Но, несмотря на это, тестирование данной задачи никак не отразится на рейтинге команды (даже в том случае, если результат тестирования окажется отрицательным).

Заголовки:

- Task-Id: идентификатор задачи, являющийся уникальным для всех задач, отсылаемых клиентом. Идентификатор необходим клиенту для определения того, к какой задаче относится результат, присланный сервером. После проверки задачи, сервер присылает отклик 202 (*Result Of Testing*), одним из заголовков которого является идентичное поле Task-Id. Данное поле будет иметь такое же значение, если результат является решением соответствующей задачи
- Task: порядковый номер задачи
- Compiler: унифицированный идентификатор компилятора, для которого писалась программа. Все возможные идентификаторы прописаны в базе данных.
- Content-Length: длина текста программы в байтах

Допустимые каналы: КУ

Описание:

При помощи данного запроса клиент может отослать задачу серверу на проверку.

В теле запроса идет сам текст программы. Необходимо, чтобы реальный размер программы совпадал с тем, что указан в заголовке Content-Length, так как никаких явных меток, указывающих на конец программы, нет. Если клиент не указал заголовок Content-Length, сервер должен ответить сообщением об ошибке 403 (*Length Required*).

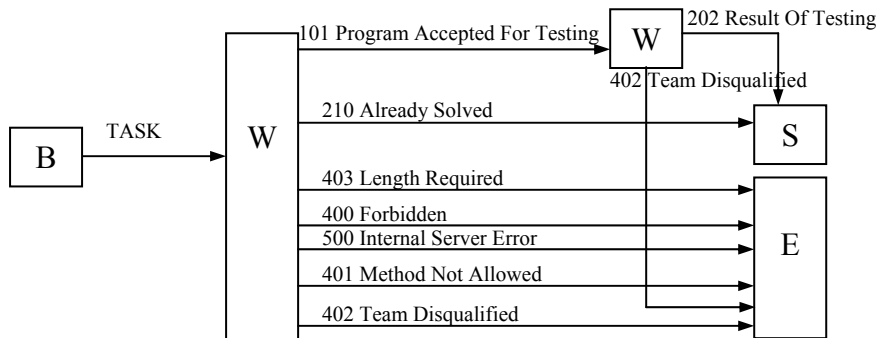
Если задача уже была ранее успешно решена командой, то сервер сразу возвращает ответ 210 (*Already Solved*). Предполагается, что сервер не будет отсылать на тестирование данную задачу, так как это приведет к лишней нагрузке тестера, тем более что данная задача никак не может повлиять на текущий рейтинг. Тем не менее данная задача регистрируется сервером в базе данных

В случае, если сервер успешно принял задачу и поставил ее в очередь на проверку, он должен вернуть клиенту отклик 101 (*Program Accepted For Testing*). После того как программа протестирована, сервер возвращает отклик 202 (*Result Of Testing*), содержащий результаты тестирования данной задачи. Спецификация не накладывает никаких ограничений на то, может ли клиент сразу после отсылки первой задачи, отсылать следующую. Так как запрос 101 принадлежит к классу 1xx запросов (См. теорию кодирования откликов сервера), то подразумевается, что выполнение команды еще не

завершено и клиент находится в состоянии ожидания ответа от сервера. Все зависит от того, может ли клиент инициировать выполнение параллельной команды и корректно отслеживать выполнение всех выполняющихся в данный момент команд. Предполагается, что сервер умеет обрабатывать параллельные команды, к тому же протокол позволяет это делать (заголовок Task-Id команды TASK предоставляет возможность идентифицировать различные команды).

Если между принятием от клиента решения и его проверкой тестером команда дисквалифицируется, то сервер вместо результатов тестирования посылает код ошибки 402 (*Team Disqualified*) и закрывает соединение с клиентом.

Диаграмма состояний:



GTT

Назначение: запрашивает у сервера номера и временные лимиты задач, а также все тесты и решения к ним

Параметры: нет

Заголовки:

- Tasks: необязательный заголовок, содержащий номера задач, тесты и решения к которым необходимо выслать. Разделены точкой с запятой. Если заголовок не указан, сервер высылает все задачи с тестами и решениями.

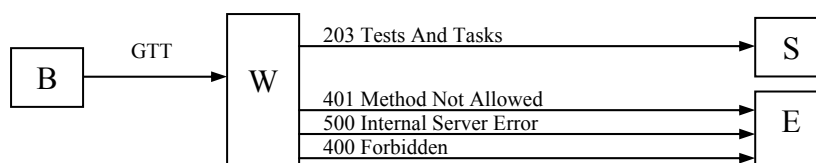
Допустимые каналы: T

Описание:

При помощи данного запроса тестеры в момент начала своей работы получают у сервера все задачи с лимитом времени, а также все тесты и решения к ним. Тестер может запросить отдельные задачи, используя заголовок Tasks (это может понадобиться в момент реактивации задач, когда необходимо обновить лишь некоторые задачи – подробнее см. команду REACTIVATE).

В случае успешного выполнения сервер возвращает ответ 203 (*Tests And Tasks*) с запрашиваемой информацией.

Диаграмма состояний:



READY

Назначение: сообщает серверу о готовности сервера принять очередную задачу на проверку

Параметры: нет

Заголовки: нет

Допустимые каналы: T

Описание:

Открытие T-канала еще не позволяет серверу отправлять тестеру задачи на проверку. Во-первых, из-за того, что возможно тестеру еще требуется запросить у сервера задачи с тестами и список компиляторов и, следовательно, он еще не готов что-либо проверять. Во-вторых, сервер сам не может ничего запрашивать, так как единственный способ его общения с клиентами – это выполнение команд по схеме запрос-ответ.

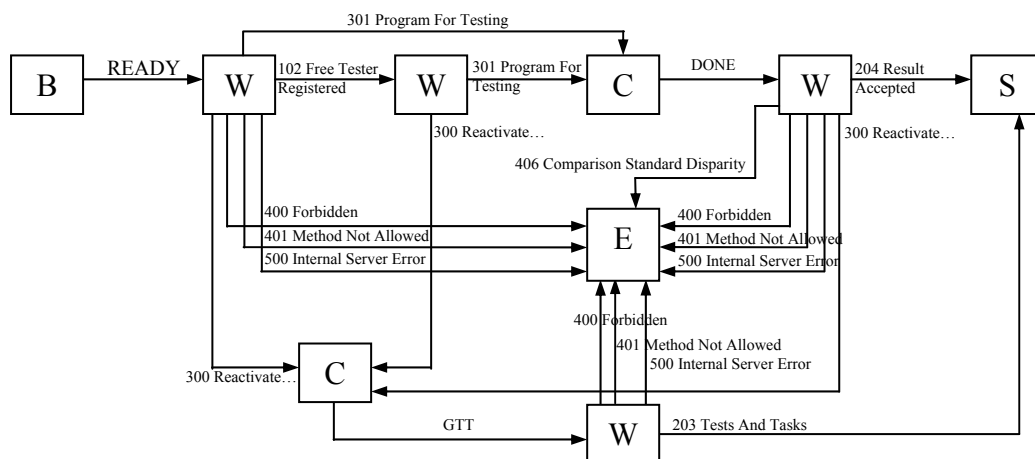
Именно для решения этих проблем и служит запрос READY, отсылаемый тестером тогда, когда он готов принимать задачи – к тому же теперь сервер имеет возможность прислать тестеру задачу на проверку вместе с ответом на данный запрос. Запрос READY является первым запросом в составной команде READY-DONE.

Общая схема выглядит следующим образом. Когда тестер готов принять задачу на проверку, он посылает серверу запрос READY. Если с сервером все в порядке, то он либо сразу же отправляет ему задачу на проверку с помощью ответа 301 (*Program For Testing*), либо, если таковых нет, отмечает тестера как свободного и посылает ему ответ 102 (*Free Tester Registered*). Во втором случае, когда появляется задача и сервер решает отправить ее на проверку, он делает это также с помощью ответа 301 (*Program For Testing*) и вдобавок к этому помечает тестера как занятого, таким образом, предотвращая отсылку одновременно двух задач на одну тестирующую машину. Тестер, получив ответ 301, проверяет задачу и возвращает результат проверки серверу с помощью запроса DONE. Сервер (опять же – если в его работе не произошло никаких фатальных ошибок) отвечает на этот запрос либо сообщением 204 (*Result Accepted*), либо если посланная тестеру задача была эталонной (см. Поведение сервера) посылает или 204 (*Result Accepted*), или 406 (*Comparison Standard Disparity*), и на этом общение сервера и тестера заканчивается, так как составная команда READY-DONE выполнена полностью. (Во втором случае после отсылки сообщения об ошибке 406, сервер удаляет тестера и закрывает с ним соединение). Теперь, для того чтобы получить новую задачу на проверку, тестер должен снова отправить серверу запрос READY.

Существует лимит ожидания сервером запроса от тестера DONE, по истечению которого сервер закрывает соединение с тестером и переадресует задачу другому тестеру.

Вместо ответов 102, 301 и 204 сервер может послать ответ 300 (*Reactivate Tests And Tasks*). В этом случае тестер должен, перед тем, как послать запрос READY, заново запросить задачи и тесты у сервера посредством запроса GTT (см. запрос REACTIVATE).

Диаграмма состояний:



DONE

Назначение: возвращает результат тестирования задачи серверу

Параметры: нет

Заголовки:

- Result: код результата
 - -2 – задача на проверялась из-за ошибки тестера
 - -1 – задача не проверялась из-за ошибки сервера
 - 0 – задача прошла все тесты успешно
 - 1 – ошибка компиляции (compilation error)
 - 2 – превышение предела времени (time limit exceeded)*
 - 3 – нарушение правил соревнований (security violation)*
 - 4 – ошибка исполнения (runtime error)*
 - 5 – нарушение формата вывода (presentation error)*
 - 6 – неверный ответ (wrong answer)*
- Message: опциональный заголовок для вывода текстового сообщения об ошибке. Тестер может по своему усмотрению уточнять тип ошибки.
- Test-Number: для всех результатов, помеченных выше звездочкой, тестер также возвращает порядковый номер теста, на котором возникла ошибка

Допустимые каналы: T

Описание:

Результат проверки задачи возвращается тестером посредством запроса DONE. Данный запрос служит только для возврата результата проверки, он никоим образом не сообщает серверу, что тестер свободен и готов принять новую задачу на проверку. Тестер должен инициировать новую команду READY-DONE, послав запрос READY.

В случае если сервер в ответ на запрос READY возвращает некорректные данные (например, не указывает номер задачи, к которой относится данная программа, или не указывает компилятор), то тестер отвечает командой DONE с кодом ошибки -1 и сообщением об ошибке в заголовке Message. Ошибки типа -1 (ошибки сервера) должны устраняться в момент разработки сервера. Спецификация не предусматривает специального поведения сервера, в случае получения им запроса DONE с кодом результата -1.

Если же тестер не может продолжать нормальную работу из-за какой-то внутренней ошибки, то он отвечает командой DONE с кодом ошибки -2 и сообщением об ошибке в заголовке Message. Получив от тестера код результата, равный -2, сервер должен послать ту же самую задачу другому тестеру и закрыть сетевое соединение с прежним тестером, если тот этого еще не сделал.

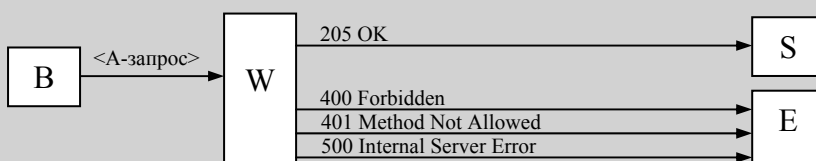
Диаграмма состояний:

См. диаграмму состояний для запроса READY.

Администраторские запросы (А-запросы)

Все администраторские запросы в случае успешного их выполнения получают от сервера один и тот же ответ – 205 (OK).

Все А-запросы также имеют одинаковую диаграмму состояний:



START

Назначение: запускает олимпиаду

Параметры: нет

Заголовки: нет

Допустимые каналы: A

Описание:

Администраторский запрос, запускающий олимпиаду. Получив данный запрос, сервер переходит в активное состояние (см. Состояния сервера), посылает всем уже подключенным КУ-клиентам ответ 209 (*Olympiad Started*) и начинает прием запросов TASK и их обработку. В случае успеха сервер возвращает ответ 205 (*OK*).

5. STATUS-CHANGE

Назначение: меняет статус олимпиады

Параметры:

- **freeze** – заморозить рейтинг
- **melt** – разморозить рейтинг
- **stop** – остановить олимпиаду

Заголовки: нет

Допустимые каналы: A

Описание:

Позволяет вручную изменить статус олимпиады. В нормальной ситуации единственным ручным вмешательством в проведение олимпиады является ее запуск посредством запроса START. Сервер имеет параметр конфигурации – t (считываемый из базы данных), определяющий интервал времени до конца олимпиады, когда замораживается рейтинг.

В момент, когда <время окончания олимпиады> - <текущее время олимпиады> $\leq t$, рейтинг замораживается автоматически. Когда время олимпиады выходит, она также останавливается автоматически.

Если же требуется ручное вмешательство в текущий ход олимпиады, то используется команда STATUS-CHANGE с одним из указанных параметров, но при этом происходят следующие необратимые вещи:

- в случае заморозки рейтинга (**freeze**), параметру t присваивается текущая продолжительность олимпиады, т.е. время <время окончания олимпиады> - <текущее время олимпиады>. Таким образом, рейтинг становится замороженным. (Прежнее значение параметра t при этом сохраняется сервером).
- в случае разморозки рейтинга (**melt**), параметру t присваивается значение 0. Таким образом, рейтинг размораживается до конца проведения олимпиады. Таким же способом можно разморозить рейтинг тогда, когда он уже был заморожен сервером.

Восстановить исходное значение параметра t можно при помощи запроса INIT. В случае успеха сервер возвращает ответ 205 (*OK*).

DSQ

Назначение: дисквалифицирует команду

Параметры: нет

Заголовки:

- IP: IP адрес машины, за которой сидит дисквалифицируемая команда.

Допустимые каналы: А

Описание:

Позволяет дисквалифицировать команду в момент проведения олимпиады. Помечает команду как дисквалифицированную, так что на любой запрос от данной команды, сервер будет возвращать 402 (*Team Disqualified*). Сервер также должен сам внести изменения в базу данных, касающиеся дисквалифицированной команды.

В случае успеха сервер возвращает ответ 205 (*OK*).

REACTIVATE

Назначение: заставляет сервер послать всем активным тестерам сообщение о реактивации

Параметры: нет

Заголовки:

- Tasks: номера задач, которые необходимо реактивировать. Разделяются точкой с запятой. Если заголовок не указан, то реактивируются все задачи. Между числом и точкой с запятой не должно идти никаких пробелов.

Допустимые каналы: А

Описание:

В случае если вносятся какие-либо изменения в задачи, тесты или ответы к ним, то необходимо, чтобы тестеры узнали об этих изменениях. Для этого администратор посылает серверу данный запрос. После этого сервер посылает всем активным тестерам сообщение о том, что им следует заново загрузить указанные (или все) задачи. Соответственно все тестеры помечаются как занятые.

В случае успеха сервер возвращает ответ 205 (*OK*).

INIT

Назначение: заставляет сервер перезагрузить все свои динамические настройки из базы данных

Параметры: нет

Заголовки:

- OlympId: идентификатор олимпиады, настройки которой необходимо загрузить (не указывается, если перезагружаются настройки текущей олимпиады)
- OlympType: тип олимпиады, настройки которой необходимо загрузить. Олимпиада в базе данных идентифицируется двойным ключом OlympId,OlympType, поэтому в заголовках указываются оба эти параметра (не указывается, если перезагружаются настройки текущей олимпиады)

Допустимые каналы: А

Описание:

Все настройки сервера можно разделить на две группы: динамические – те, которые загружаются из базы данных, и статические – которые загружаются из конфигурационного файла в момент запуска сервера.

Самым важным параметром динамической конфигурации является идентификатор текущей олимпиады, состоящий из двух параметров. Если эти параметры указаны, то сервер загружает указанную олимпиаду, если же заголовки не указаны, то сервер перезагружает текущую олимпиаду. Если в текущий момент времени идет олимпиада и серверу делается запрос INIT с идентификатором другой олимпиады, то сервер возвращает ошибку 407 (*Olympiad Currently Running*). Если делается попытка загрузить выгруженную олимпиаду, то сервер возвращает сообщение об ошибке 408 (*Olympiad Is Not In Database*).

К числу наиболее важных динамических настроек относятся следующие вещи:

- продолжительность
- пробный тур
- время до остановки олимпиады, когда замораживается рейтинг
- список используемых в данной олимпиаде компиляторов
- список всех команд – их код, статус, IP адрес, кодовое слово
- IP-адреса тестеров и рейтинг-серверов

В момент, когда загружается сервер, он считывает динамические настройки из базы данных. Далее могут возникнуть моменты, когда необходимо их перезагрузить. Для этого можно либо перезагрузить сам сервер, если надо еще обновить и статические параметры, либо использовать запрос INIT. Например, для того чтобы создать новую олимпиаду, нужно занести в базу данных всю информацию о ней, а затем послать серверу запрос INIT, после чего сервер начнет работать с вновь созданной олимпиадой.

Подразумевается, что запрос не приводит к уничтожению и созданию всех внутренних объектов сервера, так как данный запрос должен без какого-либо ущерба работать и тогда, когда олимпиада уже идет (например, если машина команды-участницы выйдет из строя и необходимо будет выдать ей новую машину, а, следовательно, и новый IP адрес).

Можно заметить, что запросы START, STATUS-CHANGE, DSQ являются своего рода “точечными” запросами, изменяющими лишь отдельные параметры динамической конфигурации сервера. Все, что делают данные запросы, можно сделать также с помощью одного запроса INIT – для этого необходимо вручную изменить соответствующие записи в базе данных и послать серверу запрос INIT.

В случае успеха сервер возвращает ответ 205 (*OK*).

RATING-UPDATE

Назначение: заставляет сервер актуализировать рейтинг

Параметры: нет

Заголовки: нет

Допустимые каналы: A

Описание:

Для того чтобы получить рейтинг, существуют два запроса: RATING и RATING-PART. Если рейтинг не заморожен, то данные для формирования ответов на эти запросы берутся непосредственно из соответствующих таблиц базы данных. В момент, когда рейтинг замораживается, сервер делает “снимок” (т.е. текст рейтинга в том же самом формате, в котором сервер отправляет его клиенту) текущего рейтинга и размещает его в памяти. При получении всех дальнейших запросов на выдачу рейтинга, сервер не обращается к базе данных, а отправляет этот самый “снимок”. Исключение составляют лишь рейтинговые

запросы, отправленные по А-каналу – в этом случае данные все также берутся из базы данных.

Рейтинг является активным только в том случае, если олимпиада идет и время заморозки еще не настало (исключение составляет А-канал – в нем рейтинг всегда активен). Следовательно, когда олимпиада остановлена (в том числе непосредственно после запуска сервера), рейтинг заморожен и существует некоторый “снимок” рейтинга.

Запрос RATING-UPDATE заставляет сервер построить новый “снимок”, исходя из данных, находящихся в базе данных сейчас. Это полезно после окончания олимпиады и оглашения результатов – когда результаты официально оглашены желательно, чтобы рейтинг, возвращаемый сервером, их показывал.

В случае успеха сервер возвращает ответ 205 (OK).

RATING

Назначение: запрашивает всю таблицу рейтинга

Параметры:

- **with-last-id:** если указать, то сервер помимо таблицы всего рейтинга вернет и идентификатор последней записи в таблице Лог олимпиады

Заголовки: нет

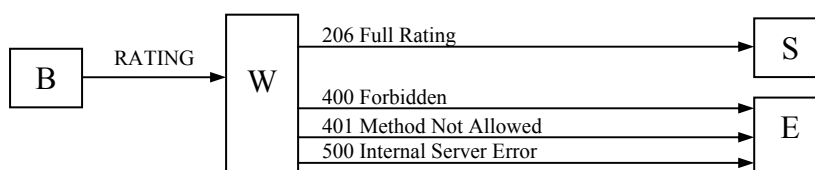
Допустимые каналы: КУ, А, Р

Описание:

Запрашивает всю таблицу рейтинга. При помощи данного запроса реализуется статически обновляемый рейтинг. Клиент не должен рассчитывать на то, что сервер вернет таблицу с уже отсортированными записями.

В случае успеха сервер возвращает ответ 206 (*Full Rating*) с запрашиваемой информацией.

Диаграмма состояний:



RATING-PART

Назначение: запрашивает последние записи из таблицы Лог олимпиады

Параметры: нет

Заголовки:

- From: запрашивает все записи из таблицы Лог решений, идентификатор которых больше указываемого в данном заголовке

Допустимые каналы: КУ, А, Р

Описание:

Клиент может реализовывать вывод рейтинга двумя способами: статическим и динамическим. При статическом способе клиент предоставляет пользователю самому решать, когда необходимо загрузить рейтинг. При динамическом – клиент загружает рейтинг автоматически через определенный интервал времени, создавая впечатление вывода рейтинга в режиме реального времени.

Для динамического вывода, клиент использует следующий механизм. Сначала он запрашивает всю таблицу рейтинга, используя запрос RATING с параметром with-last-id. После этого он через фиксированные промежутки времени посылает запрос RATING-PART с имеющимся идентификатором. Запрос RATING-PART помимо данных возвращает также новое значение идентификатора, которое клиент пошлет в следующий раз.

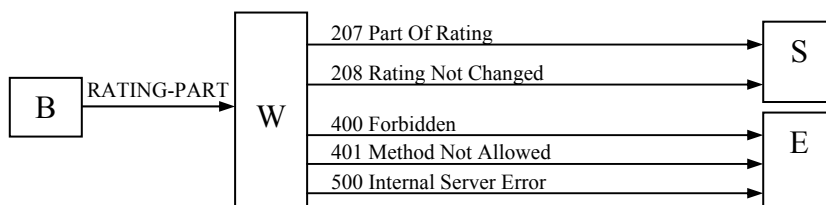
На запрос RATING-PART сервера возвращает не таблицу с рейтингом, а последние записи из таблицы Лог решений, по которым клиент сам восстанавливает картину текущего рейтинга. Если значение, указанное в заголовке From, равно идентификатору последней записи в базе данных (то есть с момента последнего запроса никаких новых данных в базу не было добавлено), то сервер возвращает ответ 208 (*Rating Not Changed*).

Когда рейтинг заморожен, при получении запроса RATING-PART, сервер ведет себя следующим образом:

- если идентификатор в заголовке From меньше или равен идентификатору (n) последней записи, положенной сервером в таблицу перед заморозкой рейтинга, то сервер возвращает записи, начиная с From + 1 и заканчивая n
- в противном случае, сервер возвращает ответ 208 (*Rating Not Changed*).

В случае успеха сервер возвращает ответ 207 (*Part Of Rating*) с запрашиваемой информацией.

Диаграмма состояний:



6. Ответы

6.1. Синтаксис ответа

Ответ имеет следующий синтаксис в БНФ-нотации:

```
ответ := <ответ> [заголовки] [тело]
ответ := <протокол>/<версия> <SP> <код> <SP> <текст ответа>
<код> := 1xx | 2xx | 3xx | 4xx | 5xx
1xx := 100 | 101 | 102
2xx := 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | 211 | 220
3xx := 300 | 301
4xx := 400 | 401 | 402 | 403 | 404 | 405
5xx := 500 | 501 | 502
<текст ответа> := {для каждого кода свой}
<протокол> := {название протокола}
<версия> := {версия протокола}
заголовки := <CRLF> заголовок | заголовки
заголовок := <Имя>: <значение>
тело := <CRLF><CRLF> {текст, предусмотренный ответом}
```

Регистр текста ответа, параметров, заголовков, а также названия протокола не важен. Тело ответа отделяется от заголовков двумя переносами строк. Порядок заголовков не имеет значения, имя заголовка отделяется от его значения двоеточием.

6.2. Описание ответов

1) *100 Wait For Beginning*

Назначение: посылается сервером клиенту при установлении КУ-канала, когда олимпиада еще не началась

2) *101 Program Accepted For Testing*

Назначение: возвращается клиенту на запрос TASK. Указывает на то, что задача успешно поставлена в очередь на проверку

3) *102 Free Tester Registered*

Назначение: возвращается тестеру на запрос READY. Указывает на то, что тестер успешно перемещен в пул свободных тестеров

4) *200 Logged In*

Назначение: сообщает об успешном открытии виртуального канала передачи данных

5) *201 Logged Out*

Назначение: сообщает об успешном закрытии виртуального канала передачи данных

6) *202 Result Of Testing*

Назначение: возвращает клиенту результат тестирования программы

Заголовки:

- Task-Id: идентификатор задачи, являющийся уникальным для всех задач, отсылаемых клиентом. Идентификатор необходим клиенту для определения того, к какой задаче относится результат, присланный сервером
- Result: код результата
 - -1 – задача не проверялась
 - 0 – задача прошла все тесты успешно
 - 1 – ошибка компиляции (compilation error)
 - 2 – превышение предела времени (time limit exceeded)*
 - 3 – нарушение правил соревнований (security violation)*
 - 4 – ошибка исполнения (runtime error)*
 - 5 – нарушение формата вывода (presentation error)*
 - 6 – неверный ответ (wrong answer)*
- Message: опциональный заголовок для вывода текстового сообщения об ошибке.
- Test-Number: для всех результатов, помеченных выше звездочкой, тестер также возвращает порядковый номер теста, на котором возникла ошибка

7) 203 Tests And Tasks

Назначение: возвращает тестеру номера задач, лимиты по времени для каждой задачи, тесты и ответы к ним

Заголовки:

- Tasks-Number: количество задач
- Time-Limits: ограничения по времени ко всем задачам в секундах; идут в порядке возрастания порядковых номеров соответствующих задач. Разделяются точкой с запятой
- Total-Tests: общее количество тестов по всем задачам
- Boundaries: через точку с запятой определяются два параметра
 - 1) test-boundary="<разделитель1>": разделитель1 – строка, ограничивающая текст теста снизу
 - 2) output-boundary="<разделитель2>": разделитель2 – строка, ограничивающая текст ответа к тесту снизу

Описание:

Тело ответа состоит из подряд идущих блоков тестов и ответов к ним:

```
<блок с тестом1>
разделитель1
<блок с ответом к тесту1>
разделитель2
...
<блок с тестомN>
разделитель1
<блок с ответом к тестуN>
разделитель2
```

<Блок с тестом> выглядит следующим образом:

```
Task: номер задачи, для которой определен этот тест
Test: номер теста
{сам текст теста}
```

<Блок с ответом к тесту> выглядит следующим образом:

Task: номер задачи, для которой определен тест
Test: номер теста, соответствующего данному ответу
{сам текст ответа}

8) *204 Result Accepted*

Назначение: возвращается тестеру в ответ на запрос DONE. Сообщает о том, что результат проверки программы принят

9) *205 OK*

Назначение: возвращается администратору при успешном выполнении A-команды

Заголовки:

- Message: текстовое сообщение о той команде, что была выполнена

10) *206 Full Rating*

Назначение: возвращает всю таблицу рейтинга

Заголовки:

- Teams-Number: количество команд, участвующих в олимпиаде (представленных в рейтинге)
- Tasks-Number: количество задач в олимпиаде

Описание:

Тело ответа содержит Teams-Number строк следующего формата:

<код команды> \t <название команды> \t <рез>₁ \t ... \t <рез>_{Tasks-Number} \t <КРЗ> \t <В>

где:

- \t – символ табуляции
- <код команды> - код команды, как он есть в базе данных
- <рез>_i – запись вида +[N] или –[N], причем знак плюс означает, что задача решена, знак минус – что задача нерешена, N – количество неудачных попыток решить задачу. Всего этих записей столько же, сколько и задач.
- <КРЗ> - количество решенных командой задач
- <В> - время, потраченное на решение задач

11) *207 Part Of Rating*

Назначение: возвращает последние записи из таблицы Лог решений

Заголовки:

- From: идентификатор последней записи в таблице Лог решений (если рейтинг заморожен, то идентификатор последней незамороженной записи)
- Records: количество записей

Описание:

Тело ответа содержит Records строк в следующем формате:

<код команды> \t <номер задачи> \t <тип ошибки> \t <время решения>

где:

- \t – символ табуляции

- <код команды> - код команды, как он есть в базе данных
- <номер задачи> - порядковый номер задачи
- <тип ошибки> - количество решенных командой задач
 - -1 – задача не проверялась
 - 0 – задача прошла все тесты успешно
 - 1 – ошибка компиляции (compilation error)
 - 2 – превышение предела времени (time limit exceeded)*
 - 3 – нарушение правил соревнований (security violation)*
 - 4 – ошибка исполнения (runtime error)*
 - 5 – нарушение формата вывода (presentation error)*
 - 6 – неверный ответ (wrong answer)*
- <время решения> - время, потраченное на решение задачи

12) 208 Rating Not Changed

Назначение: возвращает сообщение о том, что с момента последнего запроса рейтинг не изменялся

Заголовки:

- From: содержит тот же идентификатор, что и соответствующий рейтинговый запрос

13) 209 Olympiad Started

Назначение: возвращается клиенту, открывшему КУ-канал во время проведения олимпиады. Также возвращается клиенту в момент начала олимпиады, если канал уже был открыт ранее

14) 210 Already Solved

Назначение: возвращает сообщение о том, что задача уже была решена командой ранее

Заголовки:

- Task-Number: номер задачи, о которой идет речь

15) 211 Tasks

Назначение: возвращает названия задач

Заголовки:

- Tasks-Number: количество задач

Описание:

Тело ответа содержит Tasks-Number строк, каждая из которых содержит название соответствующей задачи. Все строки упорядочены по номеру соответствующей задачи.

16) 212 Compilers

Назначение: возвращает список компиляторов, допустимых в данной олимпиаде

Заголовки:

- Compilers-Number: количество компиляторов

Описание:

Тело ответа содержит Compilers-Number строк в следующем формате:

<compiler-id> \t <compiler-name>

где <compiler-id> - это унифицированный идентификатор компилятора, <compiler-name> - название компилятора. \t – это символ табуляции.

17) 220 <имя сервера> at <имя хоста>

Назначение: строка приветствия сервера. Возвращается любому клиенту в момент установления сетевого соединения

18) 300 *Reactivate Tests And Tasks*

Назначение: Сообщает тестеру о том, что необходимо заново загрузить все тесты и ответы к ним

Заголовки:

- Tasks: опциональный заголовок, указывающий, какие задачи тестеру следует загрузить заново. Номер задач идут через точку с запятой. Если заголовок опущен, то заново загрузить необходимо все задачи

19) 301 *Program For Testing*

Назначение: Возвращает тестеру программу для тестирования

Заголовки:

- Task: порядковый номер задачи, решение к которой находится в теле ответа
- Compiler: унифицированный идентификатор компилятора, для которого писалась программа. Все существующие идентификаторы описаны в базе данных
- Content-Length: длина тела ответа в байтах

Описание:

В теле ответа идет сама программа. Необходимо, чтобы ее размер соответствовал указанному в заголовке.

20) 400 *Forbidden*

Назначение: сообщает о том, что запрос не может быть выполнен из-за отсутствия прав доступа

21) 401 *Method Not Allowed*

Назначение: возвращается клиенту при попытке послать защищенный запрос, не соответствующий каналу связи

22) 402 *Team Disqualified*

Назначение: посылается клиенту в ответ на запросы LOGIN и TASK, если соответствующая команда дисквалифицирована

23) 403 *Length Required*

Назначение: возвращается клиенту в ответ на запрос TASK, если в данном запросе опущен заголовок Content-Length

24) *404 Bad Request*

Назначение: возвращается в ответ на ошибочный запрос. Ошибочным считается запрос, у которого синтаксис команды корректен, а в заголовках или теле запроса содержатся ошибки. Также возвращается в ответ на команду LOGOUT, если она послана вне какого-либо канал передачи данных

25) *405 Code-Team Disparity*

Назначение: возвращается при попытке открыть КУ-канал, если указанное в заголовке Code кодовое слово не соответствует кодовому слову команды, находящемуся в базе данных.

26) *406 Comparison Standard Disparity*

Назначение: Когда тестер подключается к серверу, открывает канал связи и после необходимых приготовлений делает первый запрос READY, сервер поочередное посылает ему определенный набор эталонных задач, с целью проверить адекватность работы тестера. Если тестер проходит эту проверку, то сервер начинает посылать ему реальные задачи для проверки, но если же на какой-либо задаче тестер ошибся, сервер возвращает ему данный отклик и закрывает с ним соединение. Для тестера подобная проверка является прозрачной, то есть после подключения к серверу тестер как и обычно начинает принимать задачи на проверку и отправлять результаты решения, но если тестер получает данный отклик, вместо ожидаемого 204 Result Accepted, то он должен просигнализировать об ошибке и более не пытаться подключиться к серверу, пока данная ошибка не будет устранена.

27) *407 Olympiad Currently Running*

Назначение: Если олимпиада уже идет, а клиент посылает запрос INIT с идентификатором другой олимпиады, сервер возвращает данный отклик.

28) *408 Olympiad Is Not In Database*

Назначение: Возвращается клиенту при попытке загрузить олимпиаду, статус которой “выгружена”. “Выгруженная” олимпиада – та олимпиада, у которой данные из некоторых таблиц выгружены во внешние файлы и не находятся в текущий момент времени в базе данных.

29) *500 Internal Server Error*

Назначение: возвращается при внутренней ошибке сервера, как например: нет ни одного зарегистрированного тестера, нет подключения к базе данных и т.д.

Заголовки:

- Message: текстовое описание ошибки

30) *501 Method Not Implemented*

Назначение: возвращается сервером, если формат строки запроса корректен, но сама команда запроса не описывается данной спецификацией

31) *502 OLYMP Version Not Supported*

Назначение: возвращается клиенту, если в запросе указан протокол более поздней версии.

