

DRAFT
Version 0.3.1 [12.04.2004]

ПРОТОКОЛ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ
“OLYMPIA”

Протокол: Илья Солдатенко

База данных: Евгений Шургалин

Март 2004

Центр Новых Информационных Технологий
Тверской государственной университет

(0822) 36-57-43

Оглавление

Оглавление	2
1. Каналы связи	4
2. Состояния сервера	6
3. Теория кодирования откликов сервера	8
4. Сводные таблицы запросов и ответов	9
5. Запросы	10
5.1. Синтаксис запроса	10
5.2. Описание команд	10
LOGIN	10
LOGOUT	12
GET-TASKS.....	12
GET-COMPILERS	12
TASK.....	13
GTT.....	14
READY.....	15
DONE.....	16
START.....	17
STATUS-CHANGE.....	17
DSQ.....	18
REACTIVATE.....	18
INIT.....	19
RATING-UPDATE.....	20
DB.....	20
RATING.....	21
RATING-PART	21
6. Ответы	23
6.1. Синтаксис ответа	23
6.2. Описание ответов	23
1) 100 Wait For Beginning	23
2) 101 Program Accepted For Testing.....	23
3) 102 Free Tester Registered.....	24
4) 200 Logged In.....	24
5) 201 Logged Out	24
6) 202 Result Of Testing.....	24
7) 203 Tests And Tasks.....	24
8) 204 Result Accepted.....	26
9) 205 OK.....	26
10) 206 Full Rating	26
11) 207 Part Of Rating.....	27
12) 208 Rating Not Changed.....	27
13) 209 Olympiad Started	27
14) 210 Already Solved	28
15) 211 Tasks	28
16) 212 Compilers.....	28
17) 213 Database Credentials.....	28
18) 220 <имя сервера> at <имя хоста>.....	28
19) 300 Reactivate Tests And Tasks	29
20) 301 Program For Testing	29
21) 400 Forbidden.....	29
22) 401 Method Not Allowed	29
23) 402 Team Disqualified	29
24) 403 Length Required.....	29
25) 404 Bad Request	29
26) 405 Code-Team Disparity.....	30
27) 406 Comparison Standard Disparity	30

28)	<i>407 Olympiad Currently Running</i>	30
29)	<i>408 Olympiad Is Not In Database</i>	30
30)	<i>500 Internal Server Error</i>	30
31)	<i>502 OLYMPIA Version Not Supported</i>	30
6. Структура базы данных		31

1. Каналы связи

Общение между клиентом и сервером происходит через виртуальный канал связи. Канал связи между клиентом и сервером устанавливается после открытия сетевого соединения по инициативе клиента. Сетевое соединение открывается только в том случае, если IP адрес клиента содержится хотя бы в одной из таблиц в базе данных (mysql:user, mysql:host, olymp:team, olymp:host). Таким образом, подключение к серверу с посторонних IP адресов невозможно. После установки сетевого соединения, клиент может послать запрос LOGIN на открытие одного из каналов связи.

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

Существует еще виртуальный канал передачи данных, по которому могут обрабатываться только запросы на выдачу рейтинга – Р-канал. Данный канал открывается автоматически после установления сетевого соединения и не требует отсылки запроса LOGIN.

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, при этом все запросы для каждого типа канала (кроме запроса LOGOUT) не пересекаются (но сами каналы могут быть открыты параллельно – т.е. с использованием одного сетевого соединения, а именно, Р-канал может быть открыт параллельно с любым другим). 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 загруженности администратора ненужной функциональностью.

Свободных запросов – то есть таких, которые можно отсылать вне какого-либо канала – не существует. Вернее есть единственный подобный запрос – запрос LOGIN.

Как было сказано ранее, канал связи устанавливается клиентом с помощью запроса серверу LOGIN. Параметр запроса указывает на тип создаваемого канала. Основными условиями открытия виртуального канала связи служат:

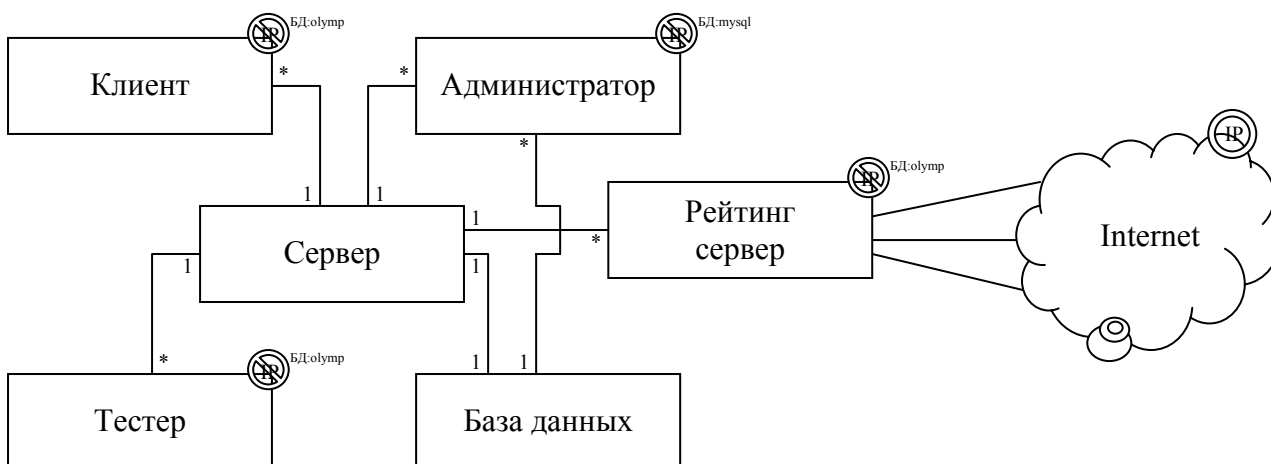
1. для КУ-каналов – клиент должен находиться на машине с IP адресом, зарегистрированным за одной из команд. IP адреса хранятся в базе данных и присваиваются каждой команде индивидуально. Помимо этого, заголовок Password запроса должен содержать кодовое слово, принадлежащее данной команде. В случае, если 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. Вместе с запросом LOGIN тестер отправляет серверу список поддерживаемых им компиляторов. Если данный список расходится с тем, что имеется у сервера, то сервер возвращает ошибку 400 и не открывает канал.
3. для А-каналов - клиент должен находиться на машине с IP адресом, прописанном в системной базе данных (mysql), как IP адрес администраторской машины и заголовок Password запроса должен содержать администраторский пароль, совпадающий с паролем, хранящимся в той же системной базе данных (mysql).
4. для Р-каналов - клиент должен находиться на машине с IP адресом, прописанном в базе данных, как IP адрес одной из команд участниц, тестирующей машины, администраторской машины, рейтинг-сервера. Как было упомянуто ранее, канал связи открывается автоматически после открытия сетевого соединения и не требует дополнительного запроса LOGIN.

В случае если клиент попытается сделать запрос, запрещенный в данном канале связи, сервер вернет ошибку несоответствия запроса каналу связи. Если же клиент попытается установить канал связи, не соответствующий его (клиента) IP адресу (или: кодовое слово или пароль не соответствуют действительным – для А- и КУ-каналов, или списки компиляторов сервера и тестера расходятся – для Т-каналов), то сервер возвращает ответ 400 с текстовым описанием ошибки в поле Message.

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:



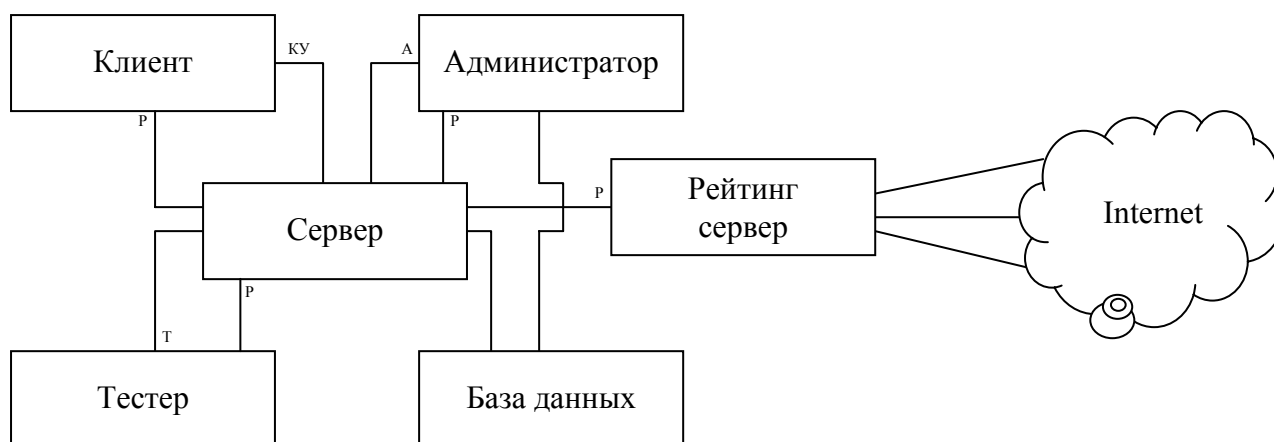
где:

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

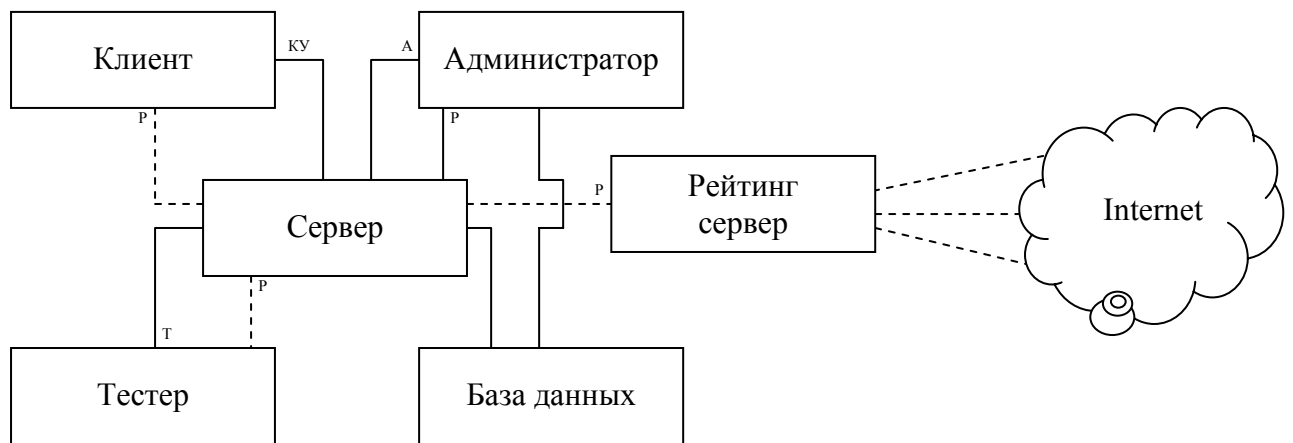
Таким образом, напрямую с базой данных могут общаться только сервер и администратор. Причем IP адрес и пароли администраторов прописан в системной базе данных mysql, что предотвращает несанкционированный доступ в БД администраторов с любой машины.

Сервер может быть в одном из следующих состояний:

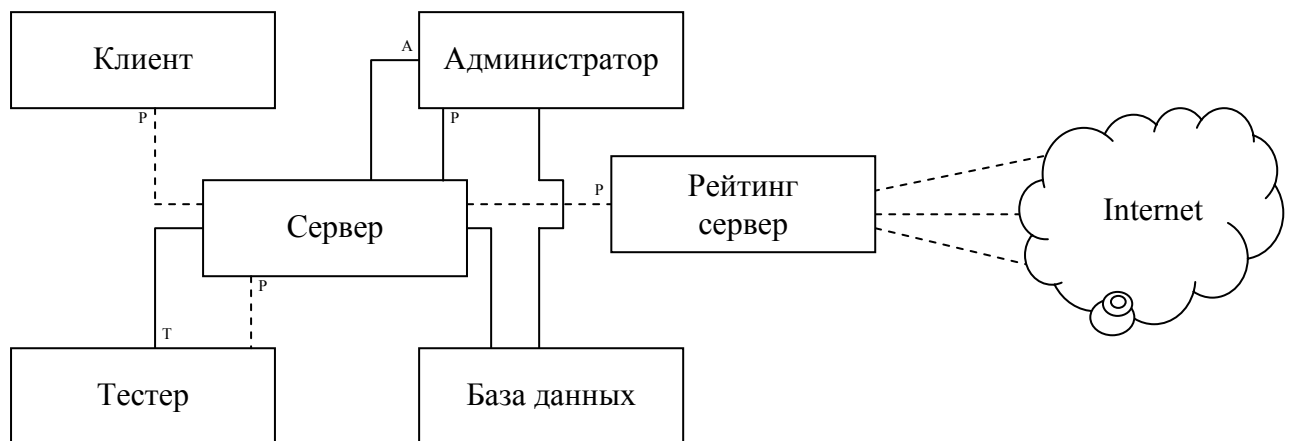
- 1) олимпиада идет (КУ, А, Т, Р – соответствующие каналы передачи данных, линии без подписей – прямой доступ)



2) олимпиада идет, рейтинг заморожен



3) олимпиада остановлена



Линии пунктиром обозначают канал, по которому передается замороженный рейтинг.

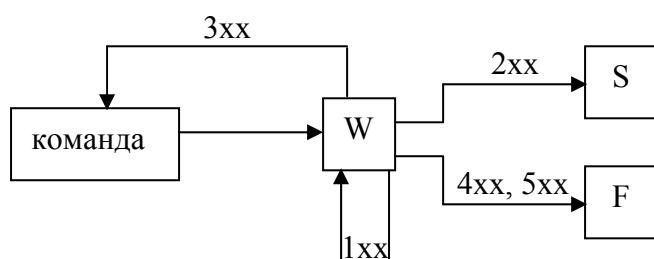
3. Принципы кодирования откликов сервера

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. Сводные таблицы запросов и ответов

Запросы	*	KY	A	T	P
LOGIN	✓				
LOGOUT		✓	✓	✓	
GET-TASKS		✓			
GET-COMPILERS		✓			
TASK		✓			
GTT				✓	
READY				✓	
DONE				✓	
START			✓		
STATUS-CHANGE			✓		
DSQ			✓		
REACTIVATE			✓		
INIT			✓		
RATING-UPDATE			✓		
DB			✓		
RATING					✓
RATING-PART					✓

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

Ответы
100 Wait For Beginning
101 Program Accepted For Testing
102 Free Tester Registered
200 Logged In
201 Logged Out
202 Result Of Testing
203 Tests And Tasks
204 Result Accepted
205 OK
206 Full Rating
207 Part Of Rating
208 Rating Not Changed
209 Olympiad Started
210 Already Solved
211 Tasks
212 Compilers
213 Database Credentials
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
409 Olympiad Not Started
500 Internal Server Error
501OLYMPIA Version Not Supported

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 <CRLF> [заголовки] <CRLF> [тело]
команда := <команда> [<SP> <параметр>] <SP> <протокол>/<версия>
<команда> := TASK | GTT | READY | DONE | START | STATUS-CHANGE |
           DSQ | REACTIVATE | INIT | RATING-UPDATE | LOGIN | DB |
           LOGOUT | GET-TASKS | RATING | RATING-PART | GET-COMPILERS
<параметр> := team | tester | admin | force | freeze | melt | stop | with-last-timestamp
<протокол> := OLYMPIA
<версия> := [0-9]* . [0-9]*
заголовки := заголовок <CRLF> | заголовок <CRLF> заголовки
заголовок := <Имя заголовка>: <значение>
<Имя заголовка> := Password | Task-Id | Task | Compiler | Content-Length | Tasks |
                 Result | Message | Test-Number | IP | OlympId | OlympType | From |
                 Content-Length
тело := {текст, предусмотренный запросом. Длина тела указана в Content-Length}
<SP> := _ | \t
<CRLF> := \n
```

Длина строки команды и заголовков не может превосходить 1024 символов. Количество заголовков также не может превосходить 1024. Если запрос подразумевает наличие тела, то среди заголовков обязательно должен присутствовать заголовок Content-Length с длиной тела в байтах.

Регистр команд, параметров, заголовков, а также названия протокола не важен. Концом заголовка запроса (включающего строку команды и опциональный блок заголовков) является пустая строка. Команда представляет собой последовательность из латинских букв и дефиса. Порядок заголовков не имеет значения, имя заголовка отделяется от его значения двоеточием. Заголовок, а также строка команды не должны содержать начальных пробелов.

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

Заголовки:

- Password: кодовое слово команды или пароль администратора, в зависимости от того, какой канал открывается. Заголовок посылается клиентом только в том случае, если заданы параметры team или admin.
- Compilers: список компиляторов, разделенных точкой с запятой, которые поддерживаются тестером. Заголовок отсылается только в том случае, если задан параметр tester.

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. Р-канал устанавливается автоматически при открытии сетевого соединения и поэтому не требует запроса LOGIN.

Сервер возвращает отклик 400 (*Forbidden*) в следующих случаях:

- если запрос производится с машины, IP адрес которой не прописан в базе данных
- если производится открытие канала, не соответствующего IP адресу
- если делается запрос на открытие А-канала, но либо отсутствует заголовок Password, либо пароль не совпадает с реальным паролем администратора
- если делает запрос на открытие Т-канала, но либо отсутствует заголовок Compilers, либо список компиляторов, поддерживаемых сервером расходится с тем, что поддерживает тестер

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

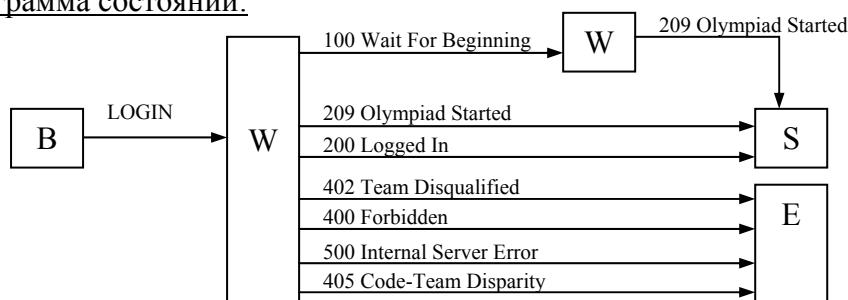
Если открытие канала происходит успешно, то ответ сервера зависит от типа открываемого канала:

- 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 или А-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- и Т- каналов (200).

Если же клиент попытается отослать задачу при помощи запроса TASK еще до начала олимпиады, то сервер возвратит ответ 409 (*Olympiad Not Started*).

Диаграмма состояний:



LOGOUT

Назначение: закрытие защищенного канала связи

Параметры: нет

Заголовки: нет

Допустимые каналы: любой канал

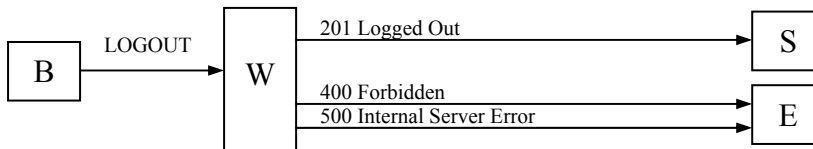
Описание:

Клиент может закрыть канал с сервером по своему желанию, пошлав 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 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

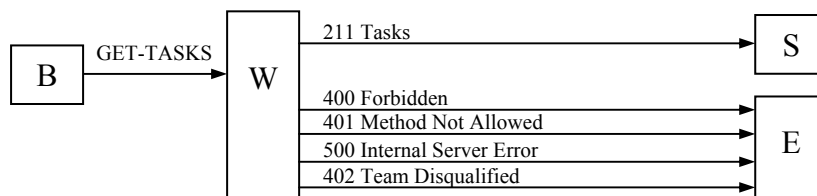
Заголовки: нет

Допустимые каналы: КУ

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

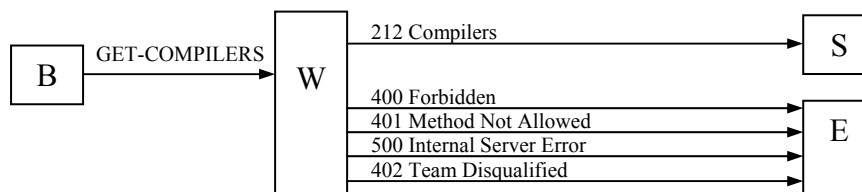
Допустимые каналы: КУ

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*). Данный запрос нужен КУ-клиенту для отображения списка имеющихся компиляторов команде.

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.
- Task: порядковый номер задачи.
- Compiler: унифицированный идентификатор компилятора, для которого писалась программа. Все возможные идентификаторы прописаны в базе данных.
- Content-Length: длина тела сообщения в байтах

Допустимые каналы: КУ

Описание:

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

В теле запроса идет сам текст программы, закодированный при помощи кодировки Base64. Это обеспечивает безопасность прохождения текста программы через сеть. 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.

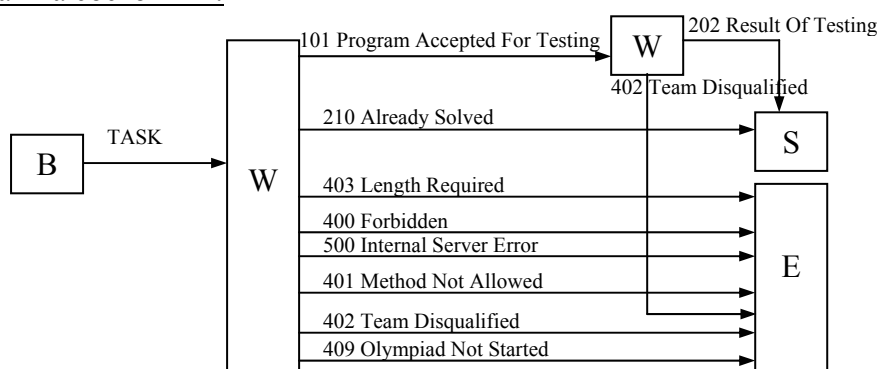
В случае если сервер успешно принял задачу и поставил ее в очередь на проверку, он должен вернуть клиенту отклик 101 (*Program Accepted For Testing*). После того как программа протестирована, сервер возвращает отклик 202 (*Result Of Testing*), содержащий результаты тестирования данной задачи. Спецификация 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.

Сервер должен гарантировать, что если клиент отправляет сразу две задачи подряд, то независимо от распределения их по очередям на тестирование, задачи попадут в базу данных и повлияют на формирование рейтинга именно в той временной последовательности, в которой они были отосланы. При этом, если первая задача была решена успешно, а во второй содержится ошибка (кроме ошибки security violation), то результат проверки второй задачи регистрируется, но уже не влияет на формирование рейтинга.

При попытке отправить задачу на проверку еще до начала олимпиады или после ее завершения, сервер возвращает ошибку 409 (*Olympiad Not Started*).

Диаграмма состояний:



GTT

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

Параметры: нет

Заголовки:

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

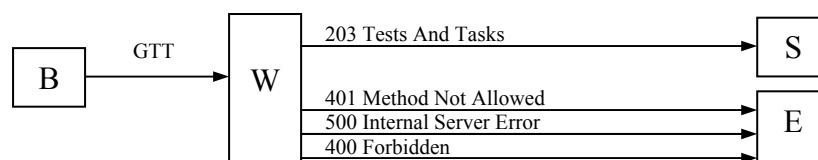
Допустимые каналы: T

Описание:

При помощи данного запроса тестеры в момент начала своей работы получают у сервера все задачи с лимитом времени, а также все тесты, решения к ним и фильтры, если таковые имеются. 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, 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.

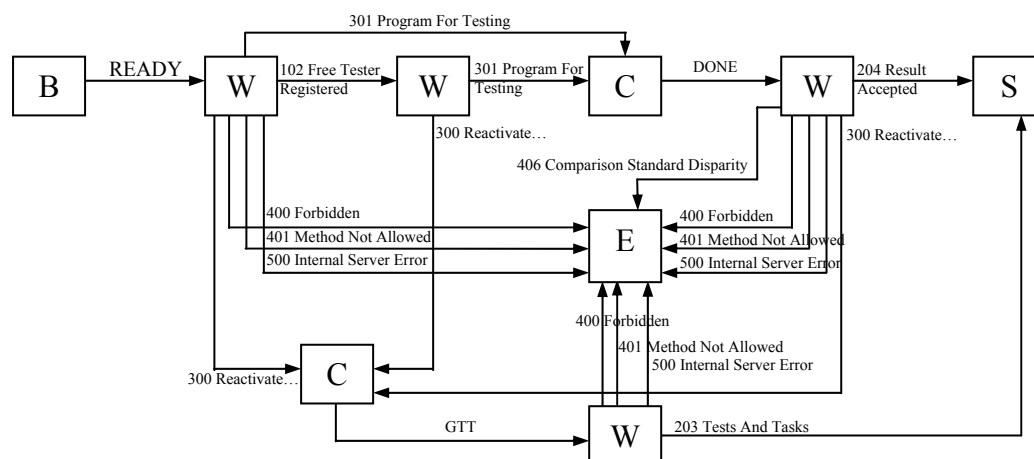
Общая схема выглядит следующим образом. Когда тестер готов принять задачу на проверку, он посылает серверу запрос READY. Если с сервером все в порядке, то он либо сразу же отправляет ему задачу на проверку с помощью ответа 301 (*Program For Testing*), либо, если таковых нет, отмечает тестера как свободного 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.

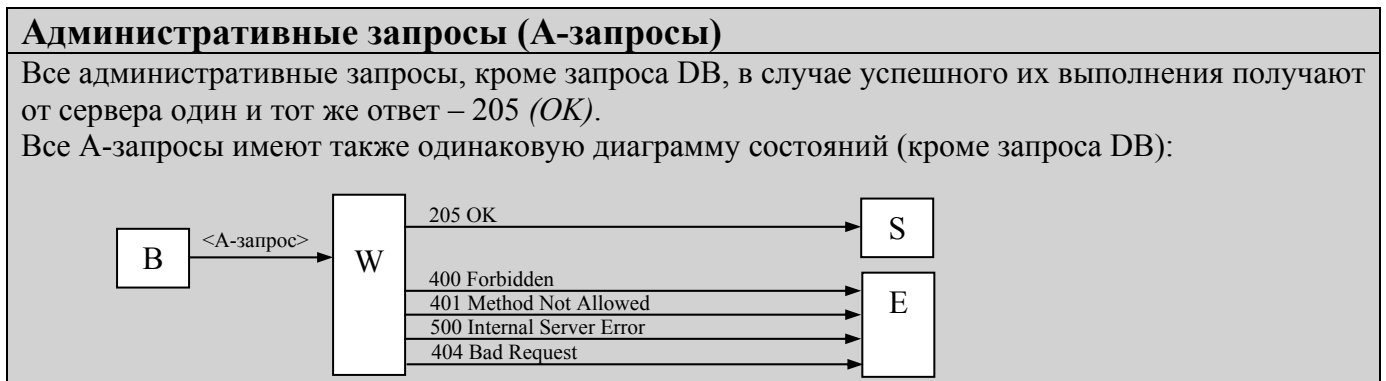
В случае если сервер в ответ на запрос READY возвращает некорректные данные (например, не указывает номер задачи, к которой относится данная программа, или не указывает компилятор), то тестер отвечает командой DONE с кодом ошибки -1 и сообщением об ошибке в заголовке Message. Ошибки типа -1 (ошибки сервера) 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.



START

Назначение: запускает олимпиаду

Параметры: нет

Заголовки: нет

Допустимые каналы: А

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. Если запрос отсылается в тот момент, когда олимпиада уже идет, сервер возвращает 404 (*Bad Request*). В случае успеха сервер возвращает ответ 205 (*OK*).

STATUS-CHANGE

Назначение: меняет статус олимпиады

Параметры:

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

Заголовки: нет

Допустимые каналы: А

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 $\langle \text{время окончания олимпиады} \rangle - \langle \text{текущее время олимпиады} \rangle \leq t$, 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 $\langle \text{время окончания олимпиады} \rangle - \langle \text{текущее время олимпиады} \rangle$. Таким образом, рейтинг становится замороженным. (Пржнее 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: A

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: A

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 и как тестеры, еще не прошедшие стадию тестирования.

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: A

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 администраторов, 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.

Перед загрузкой новой олимпиады, сервер должен проверить ее целостность (наличие хотя бы одного тестера, хотя бы одной команды, задачи, теста и т.д.). Если целостность олимпиады нарушена, сервер ее не загружает и возвращает ответ 500 (*Internal Server Error*).

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 A-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 A-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*).

DB

Назначение: запрашивает имя хоста и номер порта базы данных MySQL, а также имя базы данных, в которой хранится информация по олимпиаде

Параметры: нет

Заголовки: нет

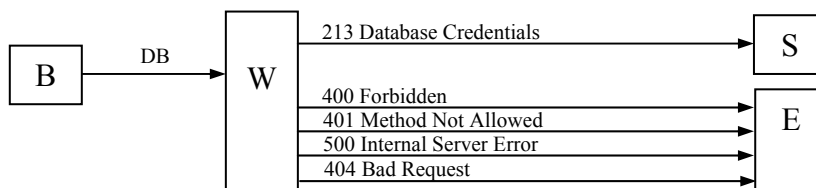
Допустимые каналы: A

Описание:

Администратор не хранит в явном виде имя хоста и номер порта для базы данных MySQL, с которой он работает. Эти данные он запрашивает у сервера в момент запуска после открытия А-канала. Имя пользователя и пароль используются те же, что и в запросе LOGIN.

В случае успеха сервер возвращает ответ 213 (*Database Credentials*) с запрашиваемой информацией.

Диаграмма состояний:



RATING

Назначение: запрашивает всю таблицу рейтинга

Параметры:

- **with-last-timestamp:** если указать, то сервер помимо таблицы всего рейтинга вернет и время добавления последней записи в таблицу Лог решений.

Заголовки: нет

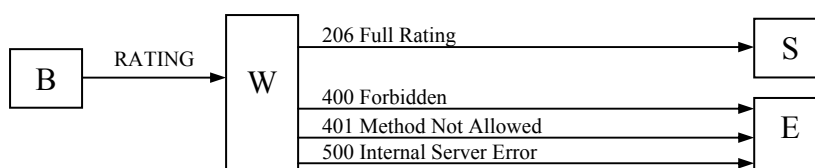
Допустимые каналы: КУ, А, Р

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, которые были добавлены в таблицу после времени, указанного 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.

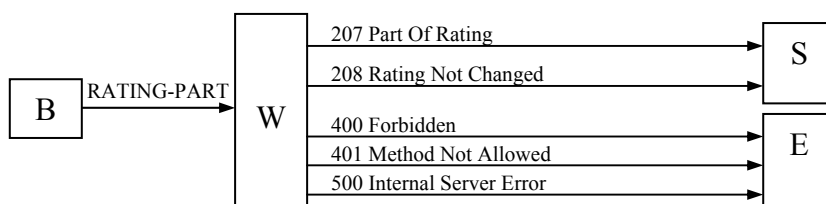
На запрос RATING-PART сервера возвращает не таблицу с рейтингом, а последние записи из таблицы Лог решений, по которым клиент сам восстанавливает картину текущего рейтинга. Если значение, указанное в заголовке From, равно идентификатору последней записи в базе данных (то есть с момента последнего запроса 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> <CRLF> [заголовки] <CRLF> [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 | 212 | 220
3xx := 300 | 301
4xx := 400 | 401 | 402 | 403 | 404 | 405 | 406 | 407 | 408
5xx := 500 | 501
<текст ответа> := {для каждого кода свой}
<протокол> := OLYMPIA
<версия> := [0-9]* . [0-9]*
заголовки := заголовок <CRLF> | заголовок <CRLF> заголовки
заголовок := <Имя заголовка>: <значение>
<Имя заголовка> := Task-Id | Result | Message | Test-Number | Tasks-Number |
                  Time-Limits | Total-Tests-Number | Boundaries | Teams-Number |
                  From | Records-Number | Task | Compilers-Number | Tasks |
                  Compiler | Content-Length | Message | Block-Number
тело := {текст, предусмотренный ответом, длина которого указана в Content-Length}
<SP> := _ | \t
<CRLF> := \n
```

Длина строки команды и заголовков не может превосходить 1024 символов. Количество заголовков также не может превосходить 1024. Если ответ подразумевает наличие тела, то среди заголовков обязательно должен присутствовать заголовок Content-Length с длиной тела в байтах.

Регистр команд, параметров, заголовков, а также названия протокола не важен. Концом заголовка ответа (включающего строку команды и опциональный блок заголовков) является пустая строка. <код> представляет собой последовательность из цифр. Порядок заголовков не имеет значения, имя заголовка отделяется от его значения двоеточием. Заголовок, а также строка команды не должны содержать начальных пробелов.

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

Заголовки:

- Task-Id: идентификатор задачи, поставленной в очереди – тот же, что и в соответствующем заголовке запроса TASK.

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, а также синтаксические фильтры, если таковые имеются

Заголовки:

- Tasks-Number: количество задач. Заголовок возвращается в том случае, если соответствующий запрос GTT не имел заголовка Tasks, то есть в теле ответа идет информация по всем имеющимся задачам. Заголовки Tasks-Number и Tasks являются взаимоисключающими – должен присутствовать обязательно только один из них
- Tasks: в заголовке через точку запятую перечисляются номера тех задач, информация по которым идет в теле ответа. Заголовок возвращается в том случае, если соответствующий запрос GTT имел заголовок Tasks. Заголовки Tasks-Number и Tasks являются взаимоисключающими – должен присутствовать обязательно только один из них
- Time-Limits: лимиты времени для каждой задачи в следующем формате: номер_задачи-лимит;номер_задачи-лимит... (Например: 1-20;2-30;4-34;5-23;3-12). Лимит времени указывается в секундах.

- Space-Limits: лимиты памяти для каждой задачи в следующем формате: номер_задачи-лимит;номер_задачи-лимит... (Например: 1-300;2-300;3-300). Лимиты памяти указываются в килобайтах
- Tasks-Names: названия задач в следующем формате: номер_задачи-название_задачи;номер_задачи-название_задачи... (Например: 1-Spreadsheet;2-Малый бизнес)
- Blocks-Number: общее количество всех блоков в теле ответа
- Content-Length: длина тела ответа в байтах

Описание:

Тело ответа состоит из последовательности подряд идущих блоков. Блоки не упорядочены, так как каждый блок содержит необходимое количество информации. Количество блоков задается в заголовке Blocks-Number.

Всего есть пять типов блоков: блок с тестом, блок с ответом к тесту, блок с грамматикой к задаче, блок с фильтром к задаче, блок с программой, решающей задачу (solution). Но в теле ответа могут встретиться не все типы блоков.

Каждый блок состоит из заголовков и тела блока. Заголовки отделены от тела одной пустой строкой. Тело блока закодировано с помощью алгоритма Base64, и его длина записана в заголовке Content-Length данного блока. Сами блоки отделены друг от друга как минимум одной пустой строкой. Каждый блок начинается со своего ключевого слова (см. описание каждого блока ниже). В блоке не может не быть заголовков или тела, поэтому в каждом блоке обязательно должен присутствовать заголовок Content-Length.

Блок с тестом имеет следующий формат:

__TEST__

Task-Number: номер задачи, к которой относится этот тест

Test-Number: номер теста

Content-Length: длина закодированного текста теста

{закодированный текст теста}

Блок с ответом к тесту имеет следующий формат:

__OUTPUT__

Task-Number номер задачи, к которой относится тест, соответствующий этому ответу

Test-Number: номер теста, соответствующий данному ответу

Content-Length: длина закодированного текста ответа

{закодированный текст ответа}

Блок с грамматикой к задаче имеет следующий формат:

__GRAMMAR__

Task-Number: номер задачи, к которой относится данная грамматика

Content-Length: длина закодированного текста грамматики

{закодированный текст грамматики}

Блок с фильтром имеет следующий формат:

__FILTER__

Task-Number: номер задачи, к которому относится данный фильтр

Content-Length: длина закодированного текста фильтра

Compiler: унифицированный идентификатор компилятора, с помощью которого необходимо компилировать данный фильтр

{закодированный текст фильтра}

Блок с решением задачи имеет следующий формат:

__SOLUTION__

Task-Number: номер задачи, к которому относится данное решение

Content-Length: длина закодированного текста решения
Compiler: унифицированный идентификатор компилятора, с помощью которого необходимо компилировать данное решение
{закодированный текст решения}

__TEST__, __OUTPUT__, __GRAMMAR__, __FILTER__ и __SOLUTION__ - идентификаторы, обозначающие начало блока соответствующего типа. Они должны идти в верхнем регистре и на отдельной строке. Между идентификатором и заголовками могут идти лишние переносы строк.

В зависимости от присутствия в теле ответа тех или иных блоков, тестер ведет себя по разному:

1. Если для какой-то задачи присутствует блок __SOLUTION__, то игнорируются все блоки __OUTPUT__ для соответствующей задачи. Ответы к тестам получаются при помощи данной программы.
2. Если для какой-то задачи присутствует блок __FILTER__, то присланные решения проверяются не побайтовым сравнением результатов, а с помощью запуска соответствующего фильтра.
3. Если для какой-то задачи присутствует блок __GRAMMAR__, то перед проверкой выходного файла, выданного присланной программой, этот файл сначала проверяется на соответствие данной грамматике.

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
- Last-Timestamp: время добавления последней записи в таблицу Лог решений. Заголовок присутствует, если в запросе был указан параметр with-last-timestamp.
- Content-Length: длина тела ответа в байтах

Описание:

Тело ответа содержит Teams-Number строк следующего формата:

```
<код команды> \t <название команды> \t <рез>_1 \t ... \t <рез>_Tasks-Number \t <КР3> \t <B>
```

где:

- \t – символ табуляции
- <код команды> - код команды, как он есть в базе данных
- <название команды> - название команды

- $\langle \text{рез} \rangle_i$ – 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.
- $\langle \text{КРЗ} \rangle$ - количество решенных командой задач
- $\langle \text{В} \rangle$ - время, потраченное на решение задач (сумма времени решенных задач + штрафное время: по 20 минут за каждую неудачную попытку решения задач, которые сейчас уже решены)

Порядок, в котором идут данные строки, не определен.

11) 207 Part Of Rating

Назначение: возвращает последние записи из таблицы Лог решений

Заголовки:

- From: время последней записи в таблице Лог решений (если рейтинг заморожен, то время последней незамороженной записи)
- Records-Number: количество записей
- Content-Length: длина тела ответа в байтах

Описание:

Тело ответа содержит Records строк в следующем формате:

$\langle \text{код команды} \rangle \backslash t \langle \text{номер задачи} \rangle \backslash t \langle \text{тип результата} \rangle \backslash t \langle \text{время решения} \rangle$

где:

- $\backslash t$ – символ табуляции
- $\langle \text{код команды} \rangle$ - код команды, как он есть в базе данных
- $\langle \text{номер задачи} \rangle$ - порядковый номер задачи
- $\langle \text{тип результата} \rangle$ - результат, которые вернул тестер для данной задачи
 - -1 – задача не проверялась (уже была решена командой ранее, на рейтинг не влияет)
 - 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)*
- $\langle \text{время решения} \rangle$ - 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 то же время, 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-Id: идентификатор задачи, который был послан клиентом вместе с этой задачей

15) *211 Tasks*

Назначение: возвращает названия задач

Заголовки:

- Tasks-Number: количество задач
- Content-Length: длина тела ответа в байтах

Описание:

Тело ответа содержит Tasks-Number строк, каждая из которых содержит номер и название соответствующей задачи в виде:

<номер задачи> \t <название задачи>

16) *212 Compilers*

Назначение: возвращает список компиляторов, допустимых в данной олимпиаде

Заголовки:

- Compilers-Number: количество компиляторов
- Content-Length: длина тела ответа в байтах

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) *213 Database Credentials*

Назначение: возвращает хоста и порт для подключения к базе данных

Заголовки:

- Content-Length: длина тела ответа в байтах

Описание:

Тело ответа содержит строку в формате: host:port/db, где host – имя хоста, port – номер порта, db – имя базы данных

18) *220 <имя сервера> at <имя хоста>*

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

19) *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. Номера задач идут через точку с запятой. Если заголовок опущен, то необходимо заново загрузить все задачи.

20) *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, закодированная с помощью алгоритма Base64.

21) *400 Forbidden*

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

22) *401 Method Not Allowed*

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

23) *402 Team Disqualified*

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

24) *403 Length Required*

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

25) *404 Bad Request*

Назначение: возвращается в ответ на ошибочный запрос. Ошибочным считается запрос, у которого синтаксис команды корректен, а в семантике содержатся ошибки. Например, возвращается в ответ на команду LOGOUT, если она послана вне какого-либо канала передачи данных, или в ответ на команду START, если олимпиада уже идет.

Заголовки:

- **Message:** текстовое описание ошибки

26) *405 Code-Team Disparity*

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

27) *406 Comparison Standard Disparity*

Назначение: Когда тестер подключается к серверу, открывает канал связи и после необходимых приготовлений делает первый запрос READY, сервер поочередно 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.

28) *407 Olympiad Currently Running*

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

29) *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.

30) *409 Olympiad Not Started*

Назначение: Возвращается клиенту при попытке послать запрос TASK в то время, когда олимпиада еще не началась.

31) *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

32) *502 OLYMPIA Version Not Supported*

Назначение: возвращается клиенту, если в запросе указан протокол более поздней версии, чем та, которую поддерживает сервер.

6. Структура базы данных