Система проведения соревнований ejudge
Руководство участника турнира

Версия ejudge — 2.3.3.

Версия документации — 20070901.

Москва — 2007

Оглавление

Глава 1

Введение

Данный документ описывает работу с системой ejudge обычного (непривилегированного) пользователя. Обычный пользователь — это обычный участник турнира, олимпиады или экзамена, который выполняет предложенные ему задания в соответствии с правилами турнира.

Описание работы с системой ejudge привилегированных пользователей (судей, администраторов и пр.) находится в отдельном документе.

Система ejudge допускает широкий диапазон настроек, в зависимости от которых интерфейс пользователя может несколько отличатся. В настоящем документе описываются все возможности, предоставляемые пользователю. В зависимости от настроек системы часть возможностей может быть недоступной.

В зависимости от специфики мероприятия, оно может называться «соревнование», «турнир», «олимпиада», «конкурс», «экзамен» и т. п. В дальнейшем всегда будет употребляться название «турнир» для всех типов соревнований. Лицо, принимающее участие в турнире, будет называться «участник» турнира.

1.1 Типы турниров

В зависимости от системы оценивания результатов и времени участия турниры делятся на несколько типов.

- **Турнир по системе ACM.** Правила такого турнира соответствуют правилам соревнований ACM ICPC. Проверка решений ведется во время турнира. Решение оценивается как «принято»/«не принято». Во время турнира участникам доступна таблица текущих результатов других команд. Как правило, за решение задач начисляются штрафные баллы в зависимости от количества неудачных попыток и времени, прошедшего от начала турнира. Как правило, турнир имеет ограниченную продолжительность.

- **Виртуальный турнир по системе ACM.** Отличие виртуального турнира от обычного в том, что в виртуальном турнире у каждого участника ведется индивидуальный отсчет времени. Виртуальный турнир может быть начат участником в произвольный момент времени. Таблица текущих результатов для каждого участника отображается в соответствии с виртуальным временем участника. Виртуальный турнир ACM всегда имеет ограниченную продолжительность.

- **Турнир по системе Kirov.** В таком турнира задача оценивается в некоторое количество баллов в зависимости от числа пройденных тестов. Как и в турнире по системе ACM

решения проверяются непосредственно во время турнира. Время, прошедшее от начала турнира, не учитывается. За решение могут назначаться дополнительные баллы, например, если это — первое решение в турнире. Турнир может иметь как ограниченную, так и неограниченную продолжительность.

- **Турнир по системе Moscow.** Правила такого турнира являются комбинацией правил турнира ACM и Kirov. Турнир может иметь только ограниченную продолжительность.

- **Турнир по системе Olympiad.** Главное отличие такого турнира — в отложенной проверке решений. Во время турнира решения участников только принимаются на проверку. При этом возможен их запуск на предварительных неоцениваемых тестах. Полная проверка решений проводится после окончания турнира, при этом проверяется только последнее принятое на проверку решение каждой задачи каждого участника. Непосредственно во время турнира участникам таблица текущих результатов недоступна.

- **Виртуальный турнир по системе Olympiad.** Как в виртуальном турнире ACM, отличие этого виртуального турнира в том, что отсчет времени для каждого участника ведется независимо от других.

В зависимости от правил регистрации турниры делятся на несколько типов.

- **Турнир с открытой регистрацией.** Такой турнир характеризуется свободной регистрацией. Любой желающий может зарегистрироваться на турнир, указав при регистрации необходимую информацию. Регистрация на турнир не требует вмешательства модератора. Другими словами, для регистрации на открытый турнир участнику только требуется заполнить регистрационную анкету.

- **Турнир с модерируемой регистрацией.** Для регистрации на модерируемый турнир участнику необходимо заполнить регистрационную анкету. Регистрация на турнир требует подтверждения модератора.

- **Турнир с закрытой регистрацией.**

По составу участников турниры делятся на личные и командные.

- **Личный турнир.** Для каждого участника могут быть заданы собственно данные об участнике, а кроме того, могут быть указаны данные тренеров и руководителей.

- **Командный турнир.** Участником командного турнира является команда, состоящая из нескольких основных участников (contestants). Кроме основных участников могут быть указаны запасные участники, тренера и руководители команды. Ограничения на максимальное количество лиц каждого класса задаются администратором турнира.

Глава 2

Вход в турнир

В данной главе рассматривается процедура входа в турнир.

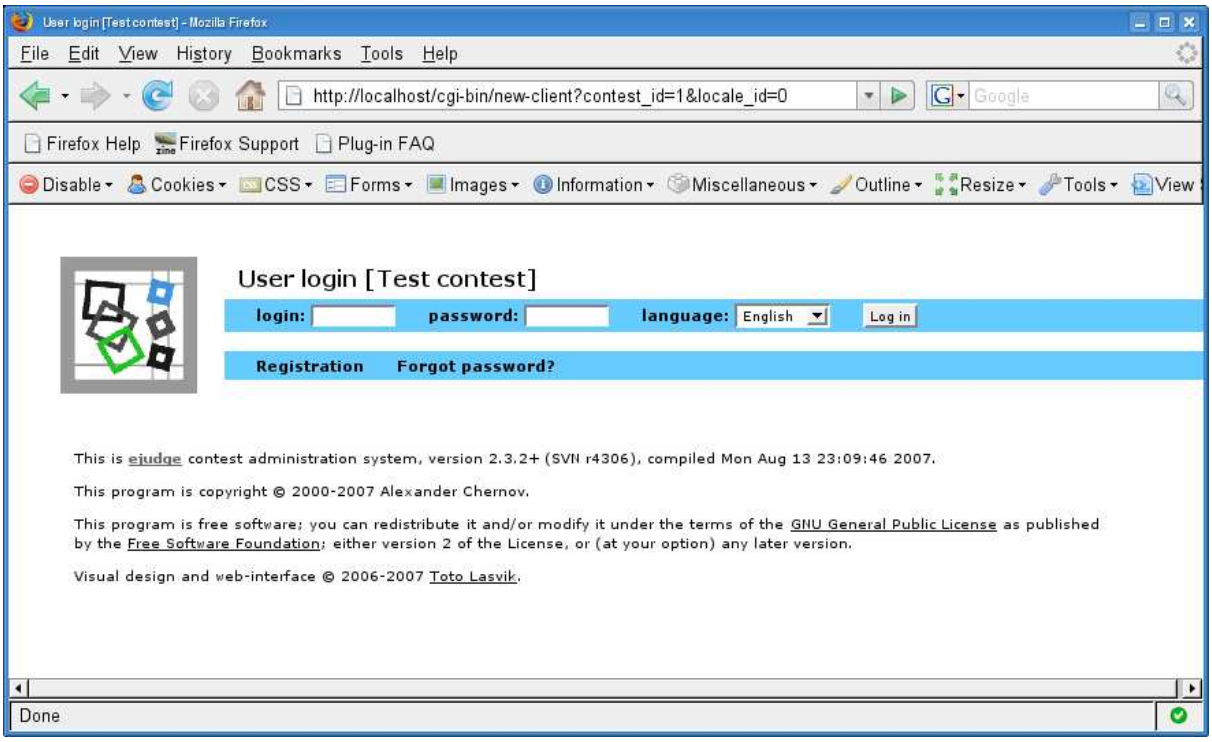


Рис. 2.1: Страница входа в систему (английская версия)

На рис. 2.1 приведён вид страницы входа в систему для английской версии программы, а на рис. 2.2 — для русской версии программы.

Страница входа в систему содержит следующие элементы:

- Поля ввода **Login** и **Password** (**Пароль**). Эти поля предназначены для аутентификации пользователей, уже зарегистрированных на турнир. Для начала работы в турнире необходимо заполнить эти поля и нажать на кнопку **Log in** (**Войти**).

- Выпадающее меню **Language** (**Язык**), которое позволяет выбрать язык отображения страниц. В текущей версии системы ejudge поддерживается два языка: English (Английский) и Russian (Русский). Желаемый язык должен быть выбран до нажатия кнопки

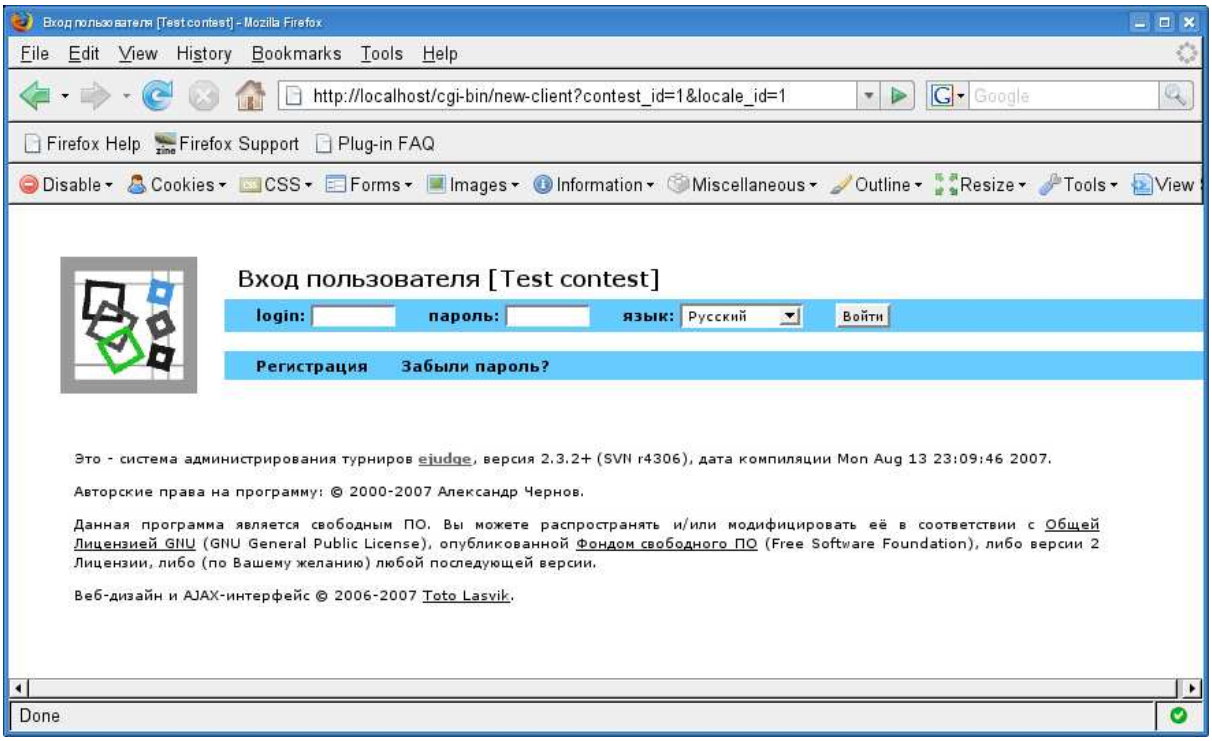


Рис. 2.2: Страница входа в систему (русская версия)

Log in. Основной язык (язык по умолчанию) турнира устанавливается администратором турнира. Если основной язык не установлен, им считается английский язык. Выпадающее меню **Language** может отсутствовать, если администратором турнира запрещено изменение языка отображения страниц.

- Ссылка **Registration** (**Регистрация**), которая используется для регистрации новых пользователей на турнир или для изменения данных о пользователе. Ссылка **Registration** может отсутствовать в турнирах с закрытой регистрацией.

- Ссылка **Forgot password?** (**Забыли пароль?**), которая используется для восстановления утраченного пароля. Ссылка может отсутствовать, если восстановление паролей запрещено администратором турнира.

В строке ввода пароля Вы должны ввести правильный пароль *участния в турнире*. В системе ejudge поддерживается два вида пароля: пароль для редактирования регистрационных данных (так называемый регистрационный пароль) и пароль участия в турнире. При создании нового пользователя системой генерируется именно регистрационный пароль, который потом может использоваться для редактирования информации о пользователе. Пароль участия в турнире может генерироваться администратором турнира уже после окончания регистрации всех пользователей и перед началом турнира.

В большинстве случаев (как правило, во всех открытых турнирах) пароль участия в турнире автоматически совпадает с регистрационным паролем. Однако в особенности в очных турнирах пароль участия в турнире может быть отличным от регистрационного пароля. Другими словами, если участникам непосредственно перед началом турнира были розданы пароли для участия, вводиться должны именно они, а не регистрационные пароли, которые, возможно, использовались участником при регистрации.

Обратите внимание, что для появления страницы входа в турнир в адресной строке браузера необходимо указание адреса, содержащего так называемый *идентификатор турнира* (contest_id). Идентификатор турнира назначается администратором турнира при его создании. Как правило, вход в турнир происходит с другой веб-страницы, в которой в ссылке, ведущей на страницу входа в турнир, уже указан идентификатор турнира. Если же при попытке обращения к системе выдано сообщение об ошибке, показанное на рис. 2.3, это значит, что идентификатор турнира не был указан или был указан неправильно. В таком случае обратитесь к администратору турнира и уточните у него адрес веб-страницы для входа в турнир.

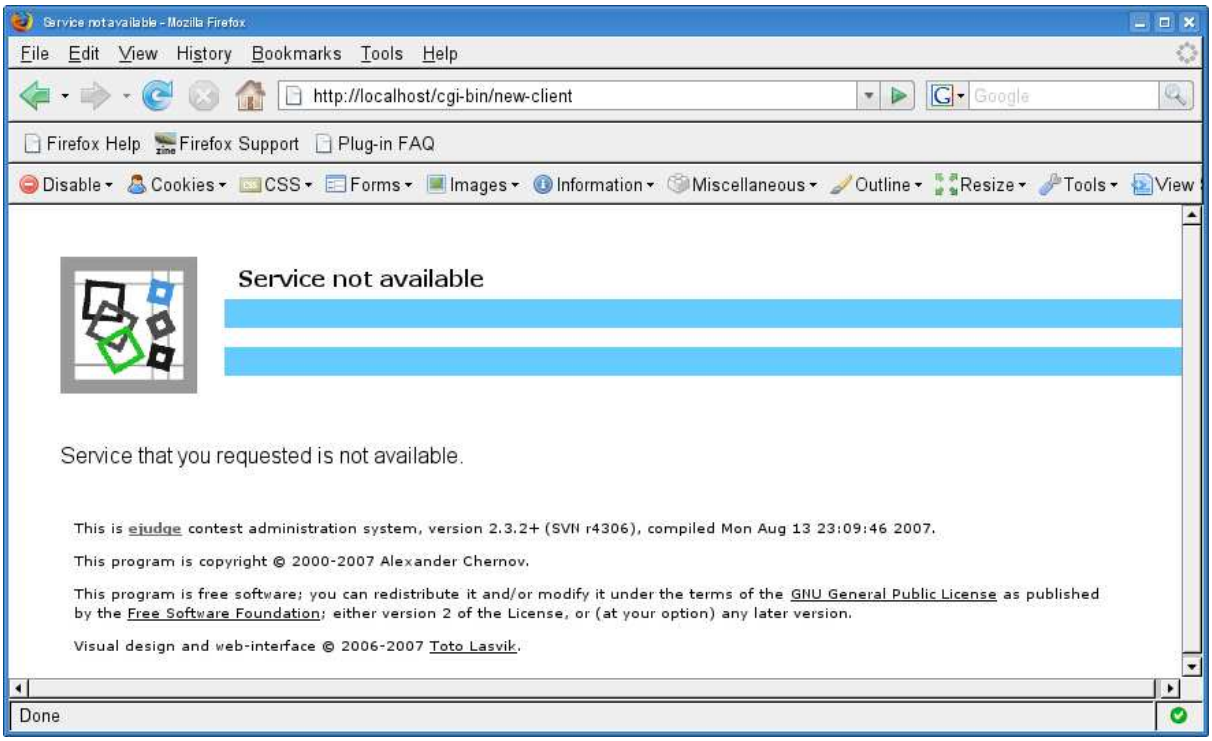


Рис. 2.3: Вид страницы, отображаемой при неправильном указании идентификатора турнира

Если login и пароль были указаны правильно и пользователь имеет полномочия участвовать в турнире, будет отображена страница участника турнира, которая подробно рассматривается в дальнейших главах. Если же вход участника в турнир невозможен, отображается страница, показанная на рис. 2.4.

Доступ участника к турниру может быть закрыт по многим причинам, которые перечислены на выводимой странице. Наиболее частая причина, конечно же, это неправильно набранный пароль. Точная причина, по которой доступ пользователя к турниру не возможен, не сообщается чтобы затруднить несанкционированное проникновение в систему. Поэтому, если при попытке входа в турнир возникла ошибка, сначала попробуйте повторить набор регистрационного имени и пароля, а если это не помогает, обратитесь к администратору турнира.

- **Неправильно введённое регистрационное имя (login)**. Необходимо проверить, что при вводе регистрационного имени не были перепутаны заглавные и строчные буквы, так как регистрационные имена чувствительны к регистру букв. Для облегчения проверки введённое регистрационное имя отображается в скобках.

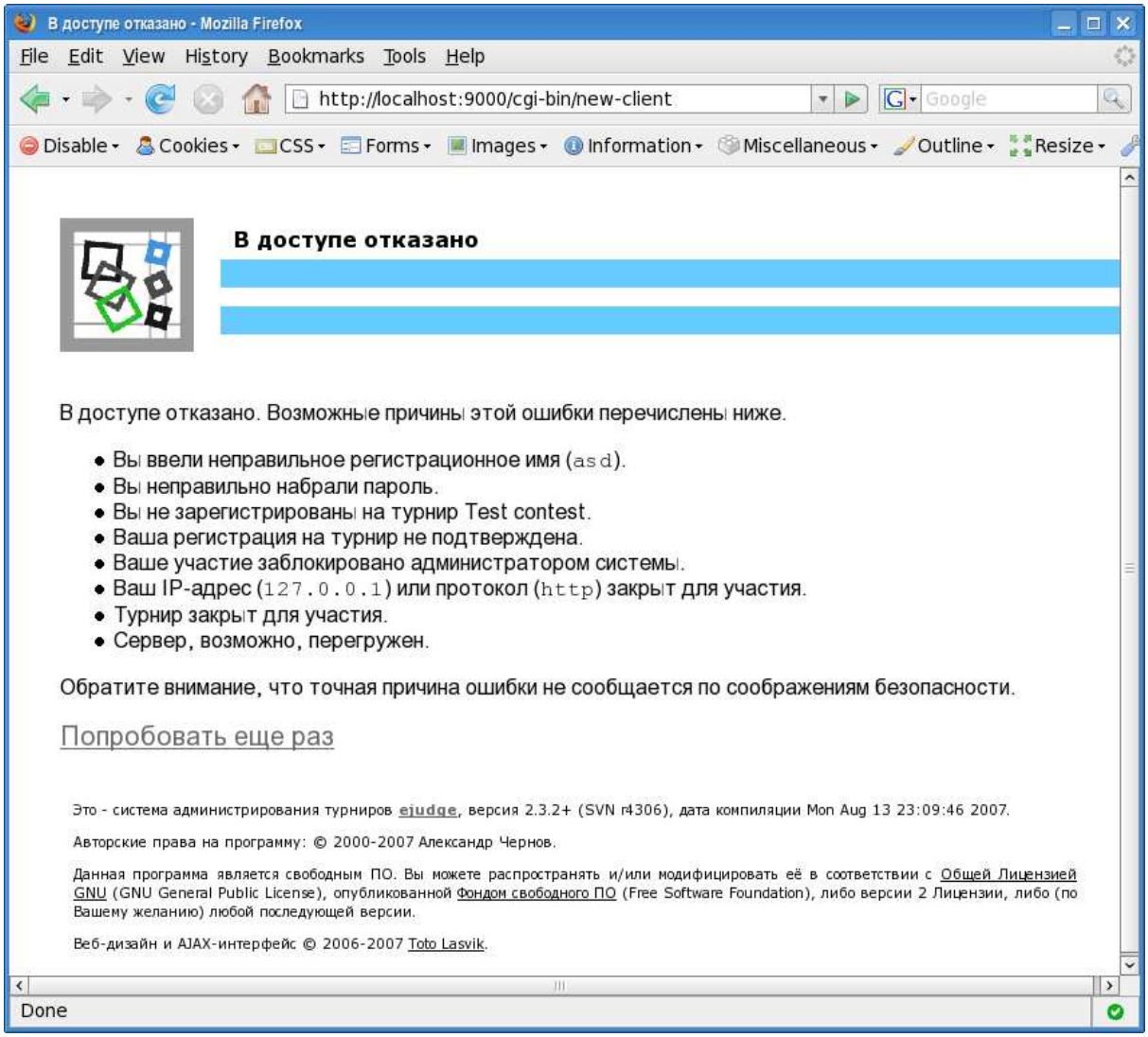


Рис. 2.4: Вид страницы, отображаемой при неуспешном входе в турнир

- **Неправильно введённый пароль**. Проверьте правильность ввода пароля. В отличие от регистрационного имени введённый пароль не отображается. Проверьте, что вводится действительно пароль участия в турнире, если это требуется.

- **Отсутствует регистрация на указанный турнир**. Регистрационное имя и пароль указаны правильно, но данный пользователь не зарегистрирован на турнир. Если турнир открытый или модерируемый, то вернитесь назад на страницу входа в турнир и проследуйте по ссылке **Регистрация**. Если турнир закрытый, обратитесь к администратору турнира.

- **Неподтверждённая регистрация на турнир**. Регистрационное имя и пароль указаны правильно, пользователь зарегистрировался на турнир, но этот турнир — модерируемый, и администратор турнира не подтвердил участие в нем. Чтобы проверить статус регистрации на модерируемый турнир вернитесь назад на страницу входа в турнир и проследуйте по ссылке **Регистрация**.

- **Участие заблокировано**. Несмотря на то, что регистрационное имя и пароль указаны правильно, пользователь зарегистрирован на турнир и его регистрация подтверждена, но администратор заблокировал доступ данному пользователю («забанил» его) за какие-либо недопустимые с точки зрения администратора проступки.

- **Неправильный IP-адрес или протокол**. IP-адрес пользователя закрыт для доступа к турниру администратором. Кроме того, возможно, что требуется доступ по протоколу https. IP-адрес клиента и имя протокола (http, https) выводятся для облегчения диагностики. Следует заметить, что данная причина, хотя и возможная, но маловероятная.

- **Закрытый турнир**. Турнир уже закончился и поэтому закрыт для участия администратором. Следует заметить, что завершение турнира само по себе не означает, что он становится закрыт для входа участников. Участники могут смотреть таблицу результатов, протоколы тестирования, сданные ими решения, подавать апелляции и т. п. Лишь после того, как администратор сочтёт, что дальнейший вход пользователей в этот турнир нецелесообразен, он может его закрыть.

Глава 3

Участие в турнире

В данной главе рассматривается интерфейс участника турнира, доступный при проведении турнира.

3.1 Страница информации о турнире

Предположим, что участник успешно авторизовался как описано в предыдущей главе. После авторизации участник попадает на страницу информации о турнире. Если турнир еще не начался, страница информации о турнире может иметь вид, показанный на рис. 3.1.

На странице отображается имя участника (на данном рисунке — это ejudge), название турнира (Test contest). Кроме того, в строке состояния (строка зеленого цвета) отображается текущее время на компьютере участника и состояние турнира (НЕ НАЧАЛСЯ). Дополнительно отображается статистика об участниках турнира.

Обратите внимание, что все даты и времена указываются по часам сервера в часовом поясе, в котором находится сервер. Именно по ним будет вестись отчет времени от начала и до конца турнира, исчисление штрафных баллов и т. п. Если время сервера и время по Вашим часам различаются, учитывайте это различие в дальнейшем. Тем не менее, если сервер подключён к сети Интернет, его время, как правило, будет синхронизировано с мировым астрономическим временем с точностью до долей секунды.

До начала турнира участник может выполнять следующие операции (меню операций занимает две голубых строки вверху страницы).

- **Настройки** — изменение настроек: языка отображения страницы и пароля участника.
- **Выйти из системы** — завершить сеанс работы с системой и выйти из турнира. Участник может снова войти в турнир в любой момент.
- **Инфо** — просмотр информации о турнире. Именно с этой страницы начинается работа пользователя в турнире.
- **Сообщения** — просмотр сообщений от судей турнира. До начала турнира участник может только просматривать сообщения от судей. После начала турнира участник может получить возможность посылать сообщения судьям. Возможность как посылки сообщений судьям, так и просмотра сообщений может быть отключена администратором турнира. Тогда пункт меню «Сообщения» не будет отображаться.

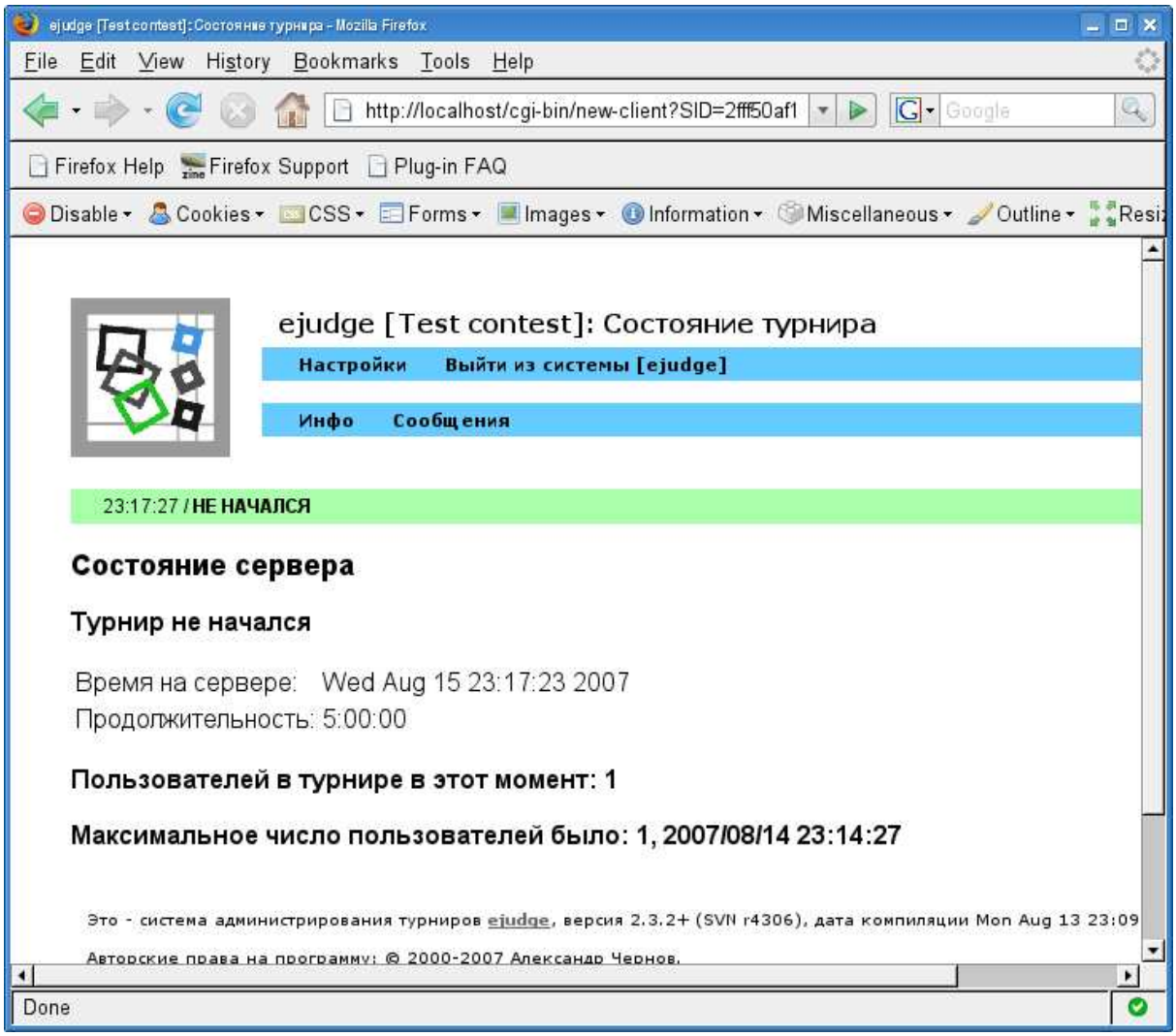


Рис. 3.1: Вид страницы, отображаемой после успешного входа в турнир

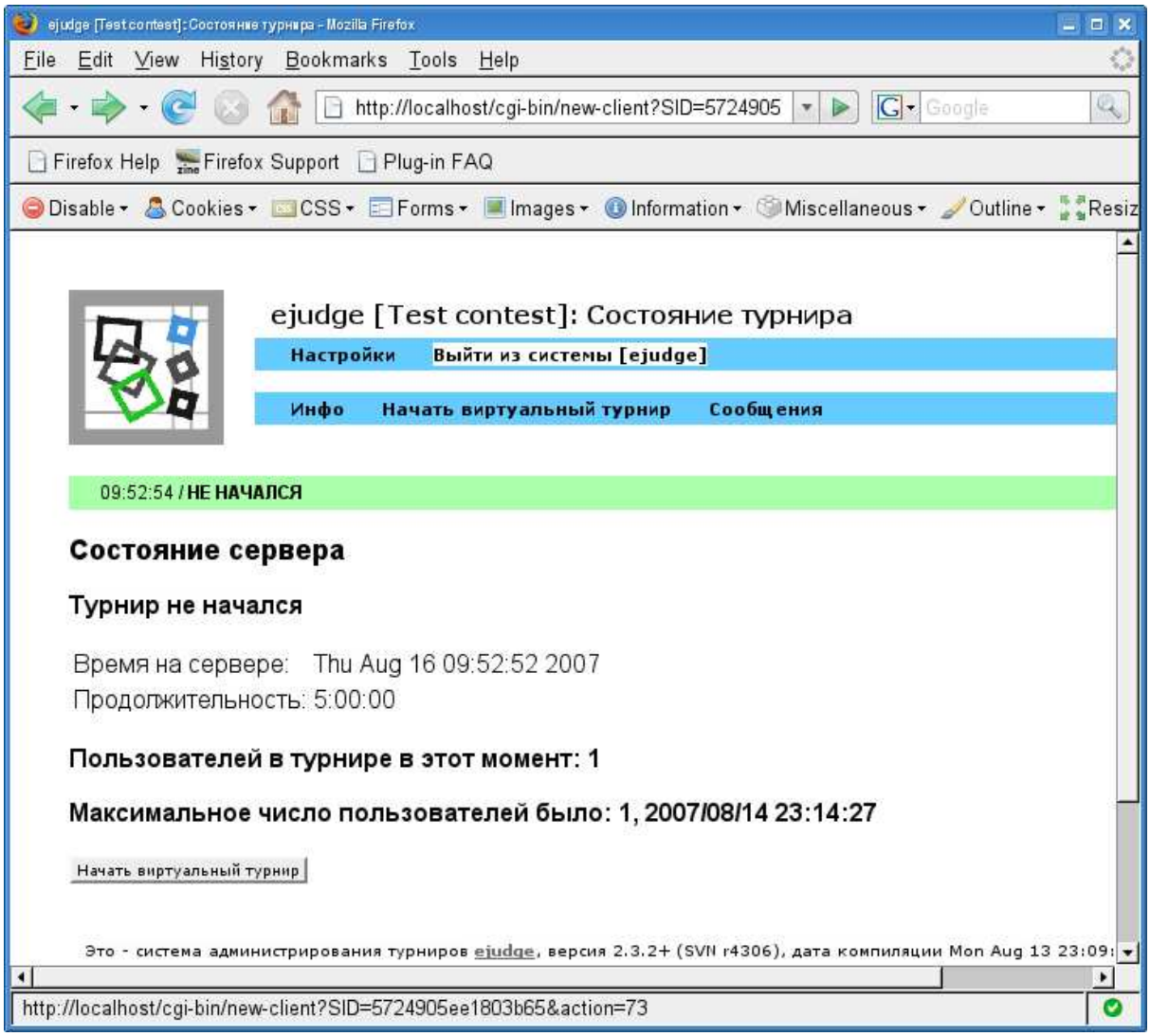


Рис. 3.2: Вид страницы, отображаемой после успешного входа в виртуальный турнир

Если турнир виртуальный, то страница информации о турнире выглядит, как показано на рис. 3.2.

В меню операций появился пункт **Начать виртуальный турнир**, кроме того, кнопка **Начать виртуальный турнир** продублирована и на странице информации о турнире.

На рис. 3.1 и 3.2 показан базовый вид страницы информации о турнире после входа в турнир, если турнир не начался. В зависимости от настроек турнира страница информации о турнире может значительно отличатся от приведённых выше. Например, на рис. 3.3 показан вид страницы информации о турнире для Демонстрационного варианта ЕГЭ 2007 (http://www.ejudge.ru/ege). Здесь пункт меню «Инфо» называется «Инструкции», «Начать виртуальный турнир» — «Начать экзамен», а остальные пункты меню недоступны.

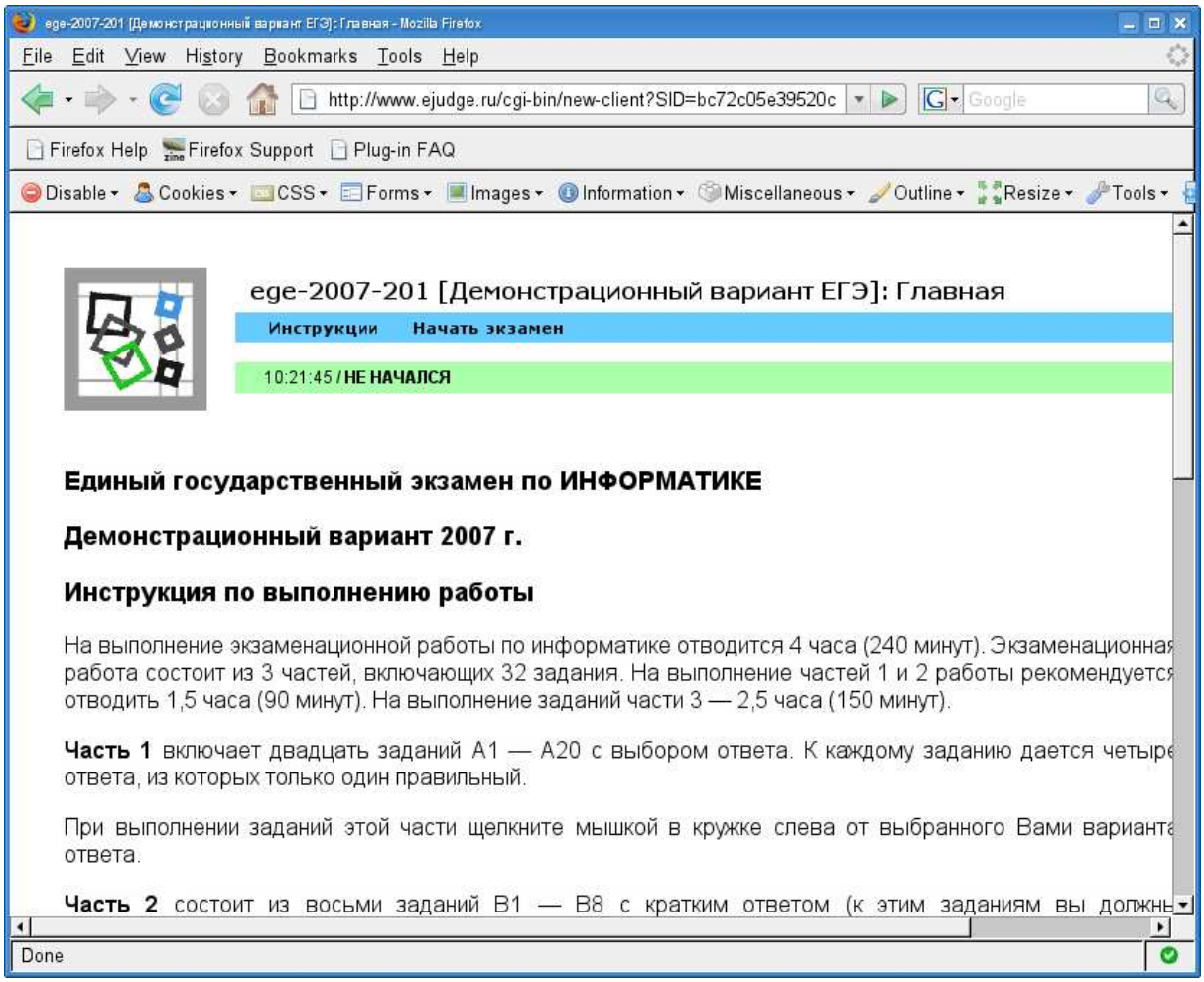


Рис. 3.3: Заглавная страница демонстрационного варианта ЕГЭ 2007 г.

3.2 Начало турнира

Виртуальный турнир начинается каждым участником самостоятельно с помощью нажатия кнопки «Начать виртуальный турнир».

Невиртуальный турнир стартует либо в момент астрономического времени, указанный администратором заранее, либо непосредственно по команде администратора турнира. Ес-

ли старт турнира запланирован на определённое время, время старта указывается в строке состояния турнира как показано на рис. 3.4.

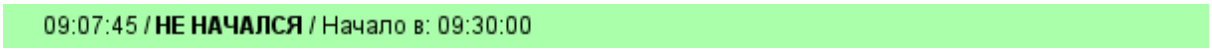


Рис. 3.4: Строка состояния турнира, показывающая время его начала

Когда настанет время начала турнира, показанное в строке состояния турнира, или когда будет объявлено, что турнир начался, участник турнира должен нажать кнопку «Refresh» веб-браузера. После этого страница информации о турнире будет перезагружена и примет вид, показанный на рис. 3.5.

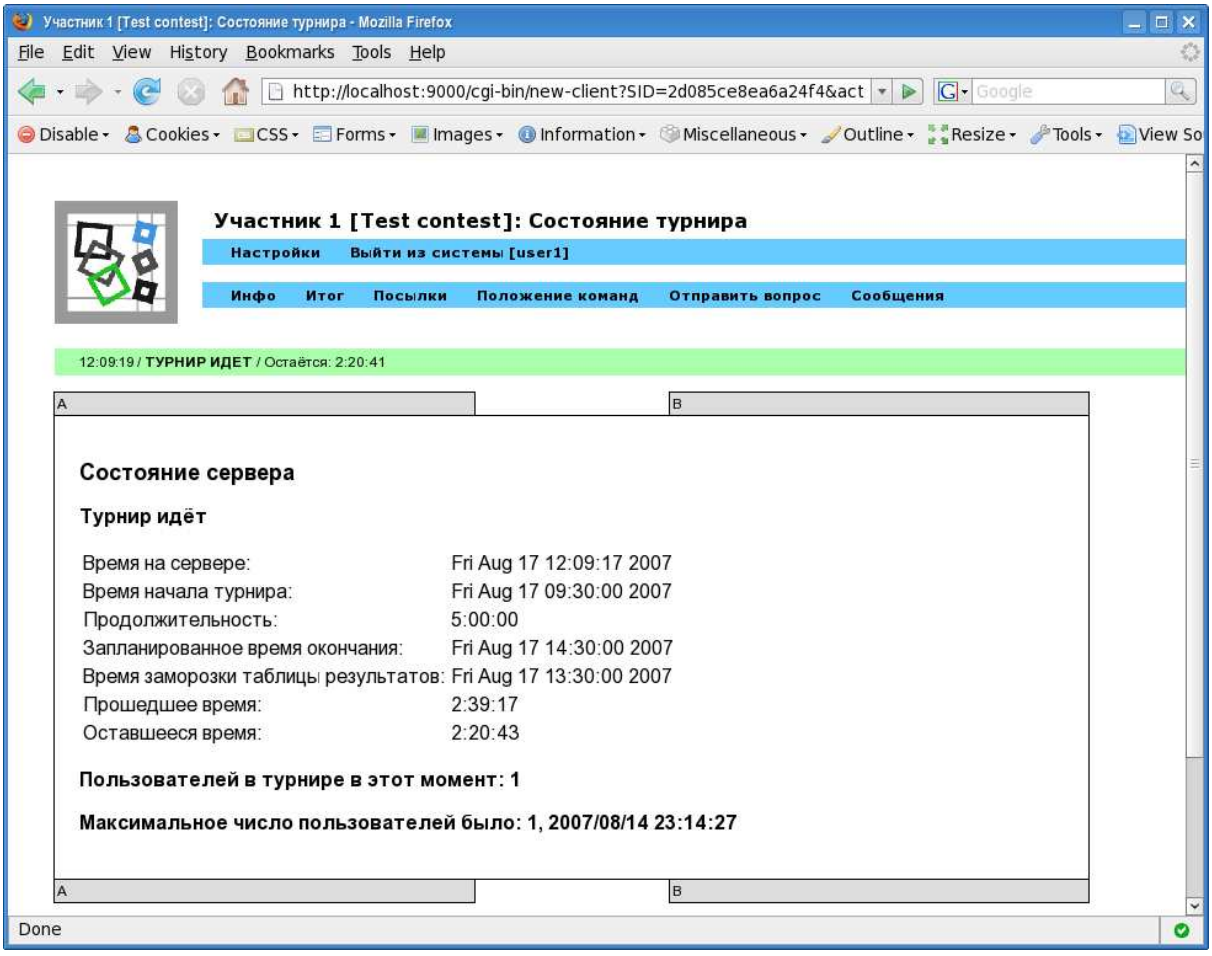


Рис. 3.5: Страница информации о турнире после его начала

В дополнение к уже рассмотренным пунктам меню «Настройки», «Выход из системы», «Инфо» появились пункты меню «Итог», «Посылки», «Положение команд», «Отправить вопрос», «Сообщения». Все пункты меню будут подробно рассмотрены в последующих разделах. Обратите внимание, что в зависимости от настроек турнира часть пунктов меню может отсутствовать или называться по-другому.

Строка состояния турнира теперь содержит текущее время сервера, которое обновляется каждую секунду, сообщение «ТУРНИР ИДЕТ» и время, оставшееся до конца турнира

(если турнир имеет ограниченную продолжительность), которое также обновляется каждую секунду.

На рабочем поле страницы появились закладки, соответствующие названиям задач («A» и «B»). Изменилась информация о турнире? которая теперь содержит следующую информацию.

- Состояние турнира («Турнир идёт»). Кроме того, здесь отображается дополнительная информация о состоянии турнира, например, информация о приостановке проверки решений, приостановке печати и т. п.
- Астрономическое время на сервере.
- Астрономическое время начала турнира, продолжительность турнира и ожидаемое время окончания турнира. Продолжительность турнира и, соответственно, ожидаемое время окончания турнира могут быть изменены администратором в ходе турнира. Для турниров неограниченной продолжительности эта информация не отображается, если только администратором не было установлено время окончания турнира.
- Время заморозки таблицы результатов. В правилах проведения турнира может быть предусмотрена «заморозка» таблицы текущих результатов, доступной участникам турнира и зрителям. Это значит, что за определённое время до конца турнира (обычно за 1 час) таблица результатов перестаёт обновляться, то есть участники и зрители видят положение участников на момент начала заморозки.

  В ходе турнира может возникнуть ситуация, когда решения участников будут перетестированы из-за изменения ограничений на решение, изменения тестов и т. п. Хотя, конечно же, эта ситуация нежелательна, на турнирах она встречается, к сожалению, достаточно часто. В этом случае замороженная таблица будет обновлена, чтобы отражать положение участников на момент начала заморозки после перетестирования.

  Для турниров неограниченной продолжительность, заморозка не предусмотрена никогда.
- Время, прошедшее от начала турнира, и время, остающееся до конца турнира.
- Статистика об посещаемости турнира.

3.3 Действия во время турнира

В данном разделе рассматриваются действия, доступные участнику во время турнира.

На рис. 3.5 показано, что каждой задаче, предложенной на турнире, соответствует закладка с именем задачи. Цвет закладки в ходе турнира может изменяться и зависит от того, решена ли данная задача данным участником или нет. Возможные цвета закладки приведены ниже в таблице 3.1

Если хотя бы одна из закладок отмечена жёлтым цветом, то есть какое-либо решение участника находится в процессе проверки, страница в браузере будет автоматически перезагружаться каждые 5 секунд, пока все задачи участника не будут проверены.

белый	текущая задача (независимо от того, решена она или нет)
серый	не было попыток решения задачи
красный	задача не решена или решена не полностью
зелёный	задача решена
жёлтый	какое-либо из решений задачи в данный момент проверяется

Таблица 3.1: Цвета закладок

3.3.1 «Итог» — итог по задачам

При выборе пукта меню «Итог» отображается страница результатов проверки каждой задачи. Пример страницы результатов проверки показан на рис. 3.6.

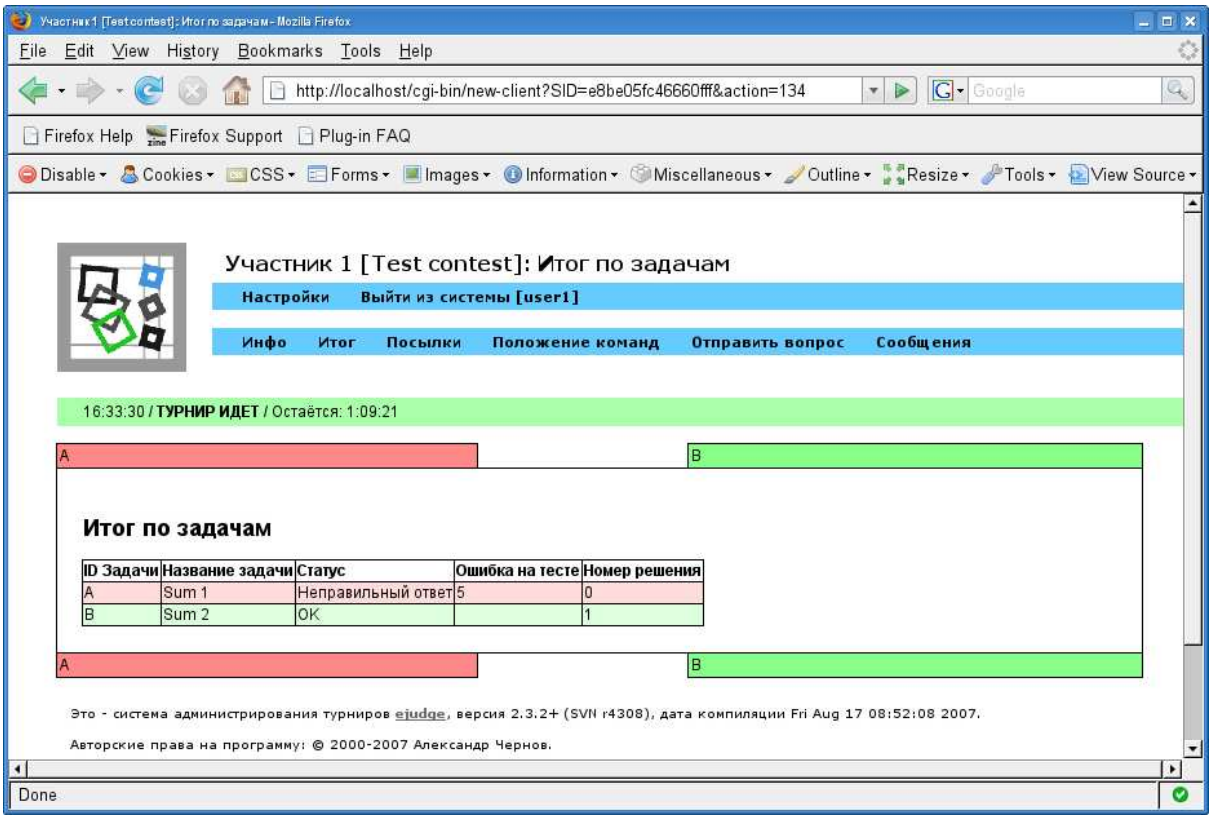


Рис. 3.6: Страница суммарной информации по задачам

Обратите внимание, что закладка задачи «A» имеет красный цвет, поскольку задача A не решена, а закладка задачи «B» — зелёная.

В зависимости от типа турнира итоговая таблица может иметь немного другой вид. Например, в турнира типа Kirov, Olympiad и Moscow дополнительно отображается балл, полученный за каждую задачу.

В таблице для каждой задачи отображается посылка, которая которая будет давать вклад по данной задаче в окончательный результат участника. Правила выбора такой посылки зависят от типа турнира.

- В турнирах типа ACM, Kirov и Moscow отображается последняя посылка по задаче, кроме посылок, отправленных после того, как задача была принята.

- В турнирах типа Olympiad отображается последняя посылка по задаче, которая прошла предварительные тесты.

3.3.2 «Посылки» — просмотр журнала посылок

При выборе пункта меню «Посылки» отображается журнал посылок данного участника. Пример страницы журнала посылок показан на рис. 3.7.

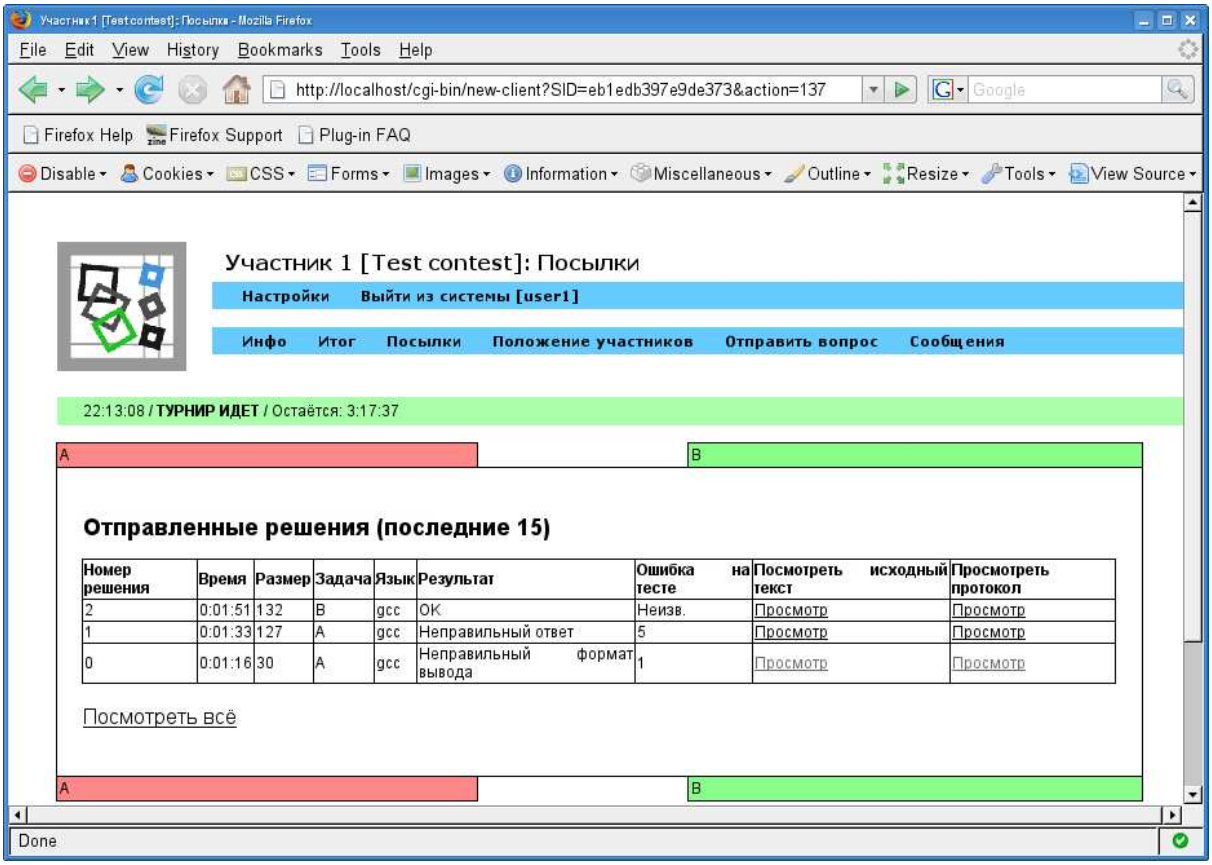


Рис. 3.7: Страница журнала посылок

По умолчанию показываются последние 15 посылок. Если было сделано более чем 15 посылок, просмотреть список всех посылок можно нажав на ссылку «Просмотреть все».

Строки в таблице посылок располагаются в порядке уменьшения времени посылки (то есть более поздняя посылка отображается выше).

Столбец «Номер решения» — это уникальный номер решения в турнире. Все посылки всех участников турнира нумеруются последовательно, начиная с 0. Номер посылки является её уникальным идентификатором в рамках одного турнира. Чем больше номер посылки, тем позже в турнире она была сделана.

Столбец «Время» — это время посылки от начала турнира. Для турниров большой продолжительности или неограниченных по времени администратор турнира может установить режим астрономического времени, тогда в этом столбце будет выводиться астрономическое время.

Столбце «Размер» — это размер посылки в байтах.

Столбец «Задача» — это короткое имя (идентификатор) задачи. Кототкое имя задачи обычно представляет собой короткую строку из латинских букв и цифр.

Столбец «Язык» — это идентификатор языка программирования, который был использован при решении задачи. Для задач, решение которых не требует указания языка, данный столбец будет пустым.

Столбец «Результат» — это результат проверки посылки. Более подробно о возможных результатах можно прочитать в соответствующем разделе.

Столбец «Ошибка на тесте» содержит номер теста, на котором программа выдала неправильный результат. Для турниров типа Kirov, Olympiad данный столбец называется «Пройдено тестов» и содержит количество тестов, на которых тестируемая программа выдала правильный результат. Кроме того, турниры Kirov и Olympiad содержат дополнительный столбец с количеством набранных баллов.

Нажав на ссылку «Просмотр» в столбце «Посмотреть исходный текст» можно посмотреть на исходный текст данной посылки. Нажав на ссылку «Просмотр» в столбце «Просмотреть протокол» можно открыть страницу просмотра протокола тестирования данной посылки. Примерный вид протокола тестирования показан на рис. 3.8. Администратор турнира может отключить возможность просмотра как исходного текста посылки, так и протокола тестирования. В этом случае соответствующие столбцы таблицы будут отсутствовать.

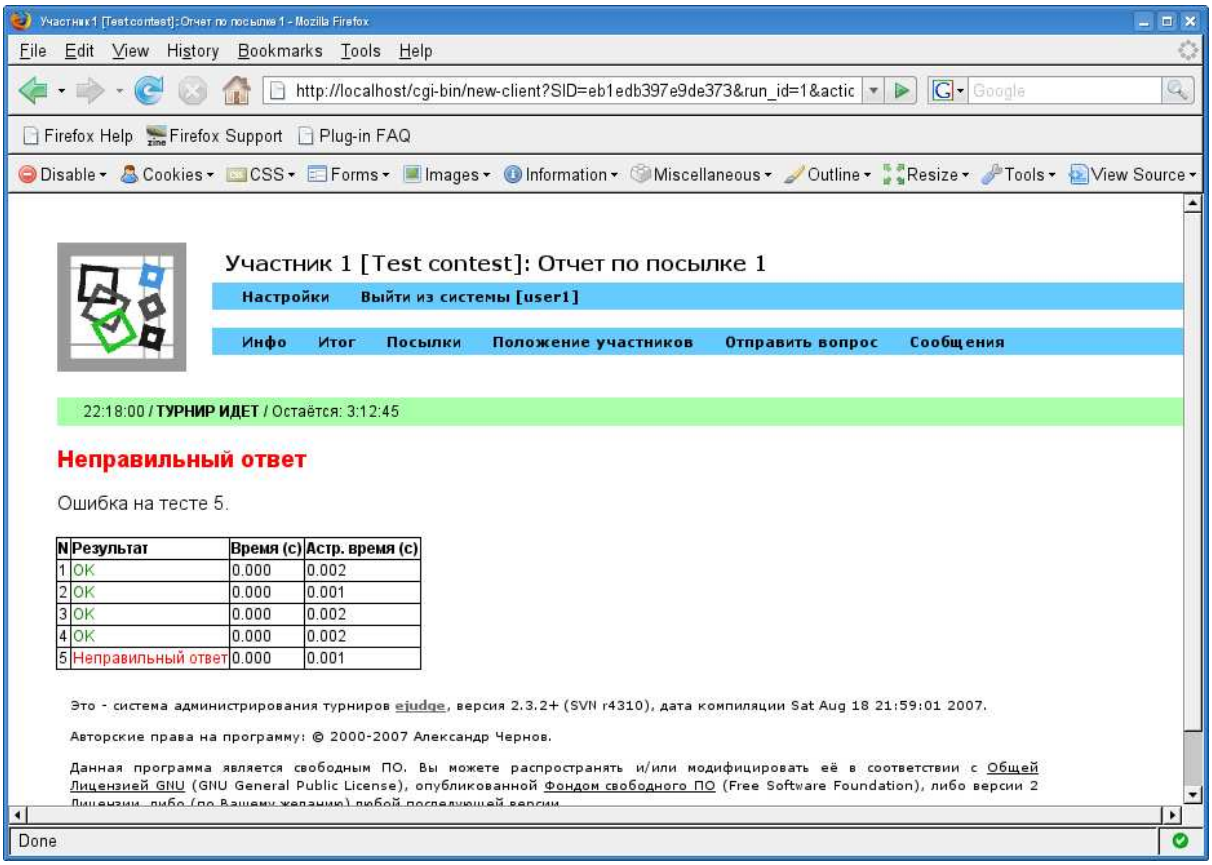


Рис. 3.8: Просмотр протокола тестирования

Протокол тестирования содержит информацию о прохождении тестируемой программы на тестах, на которых она запускалась. В турнирах по правилам ACM и Moscow тестирование ведётся до первого непрошедшего теста, в турнирах по правилам Olympiad в режиме приёма

решений на проверку тестирование ведётся до первого непрошедшего теста из пробных тестов. В турнирах по правилам Kirov и в турнирах по правилам Olympiad в режиме проверки программа запускается на всех тестах из набора тестов.

Столбец «Время» содержит процессорное время, затраченное при работе программы. Процессорное время измеряется с точностью, не выше чем значение кванта времени центрального процессора. В зависимости от настроек ОС значение кванта времени может изменяться от 1мс до 10 мс. Таким образом, если, например, значение кванта времен ЦП равно 10 мс, и программа затратила 2 мс процессорного времени, процессорное время, отображаемое в таблице, может быть равно как 0.000 с, так и 0.001 с.

Столбец «Астр. время» содержит астрономическое время, в течение которого работала программа. Значение в этом столбце получается разностью астрономического времени в моменты начала и завершения работы. Астрономическое время измеряется с микросекундной точностью. Астрономическое время должно быть примерно равно процессорному. Они могут незначительно различаться в обе стороны. Далее в соответствующем разделе обсуждаются детали измерения времени при тестировании программ.

3.3.3 «Положение участников»

При выборе пункта меню «Положение участников» отображается текущая таблица результатов турнира. Если в ходе турнира наступил момент заморозки результатов, показывается таблица результатов на момент заморозки. На рис. 3.9 показан вид страницы текущих результатов.

3.3.4 «Отправить вопрос»

При выборе пункта меню «Отправить вопрос» отображается страница посылки вопроса жюри, показанная на рис. 3.10. Администратор турнира может отключить возможность отправки вопросов жюри, тогда данный пункт меню будет недоступен.

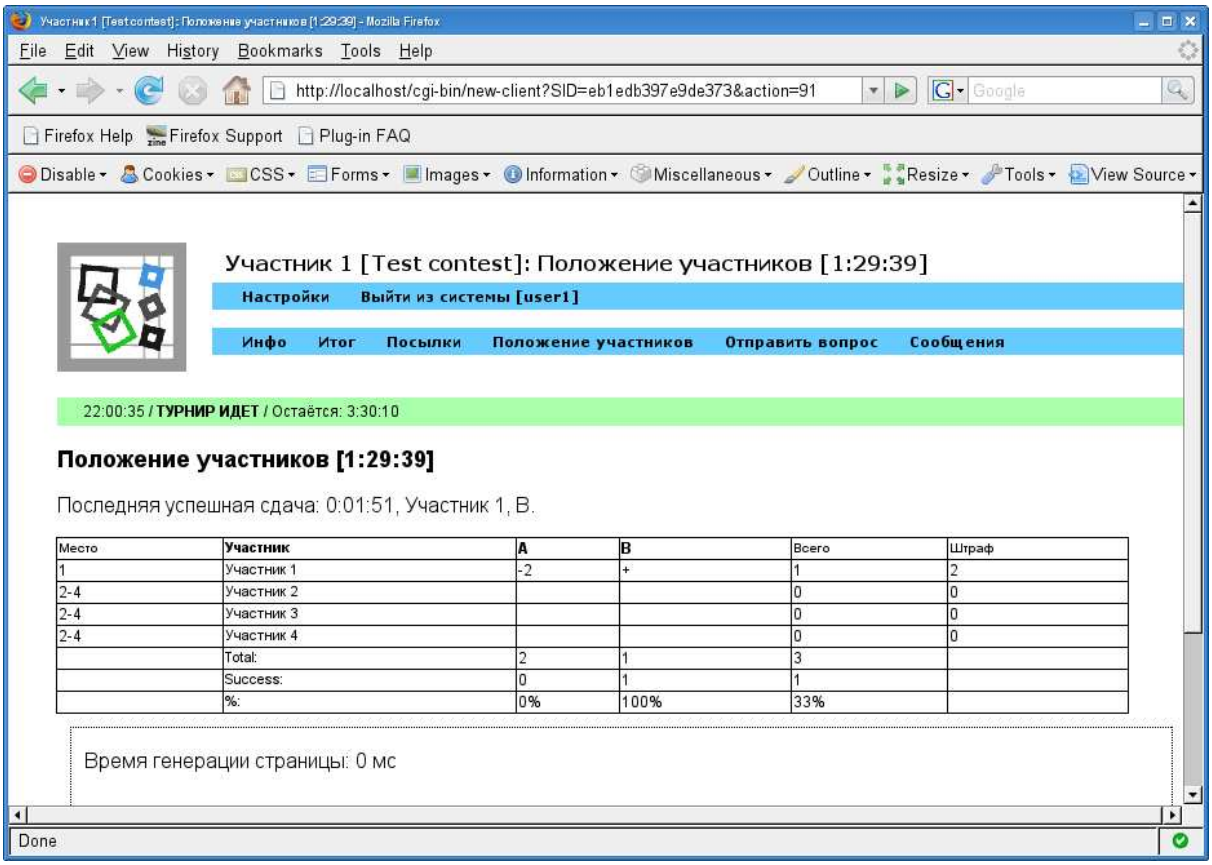
В выпадающем меню «Задача» можно указать задачу, по которой задаётся вопрос, а в поле «Тема» сформулировать тему вопроса. Затем в поле ввода набирается собственно вопрос. Общая длина вопроса ограничена и равна по умолчанию 1024 байта, но администратор турнира может изменить это ограничение. Для отсылки вопроса необходимо нажать на кнопку «Отправить».

3.3.5 «Сообщения»

Пункт меню «Сообщения» позволяет просматривать сообщения, отправленные жюри, и ответы, полученные от жюри. Если у участника есть не просмотренные им сообщения от жюри, строка состояния турнира имеет жёлтый цвет, и в ней появляется надпись *n* **непрочитанных сообщений** (как показано на рис. 3.11. Участник может нажать на ссылку «Сообщения» и перейти на страницу просмотра сообщений, показанную на рис. 3.11.

В таблице выводятся последние 15 сообщений, отправленных участником жюри и полученных им от жюри. Чтобы посмотреть все сообщения необходимо нажать на ссылку «Посмотреть все».

В таблице сообщений содержится следующая информация. **Номер сообщения** — это порядковый номер сообщения в турнире. Все сообщения всех пользователей нумеруются от 0, таким образом номер сообщения позволяет однозначно идентифицировать сообщение в турнире.



Место	Участник	A	B	Всего	Штраф
1	Участник 1	-2	+	1	2
2-4	Участник 2			0	0
2-4	Участник 3			0	0
2-4	Участник 4			0	0
	Total:	2	1	3	
	Success:	0	1	1	
	%:	0%	100%	33%	

Рис. 3.9: Страница текущих результатов

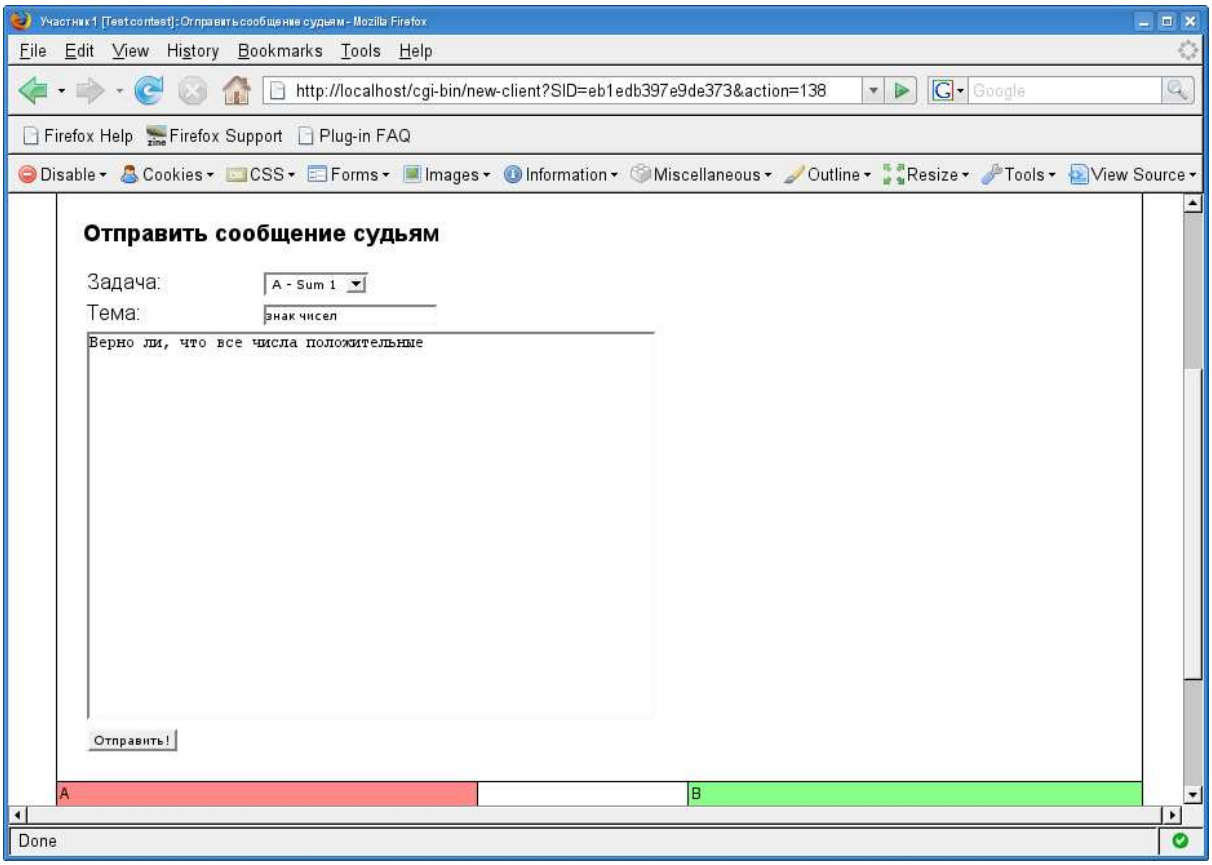


Рис. 3.10: Отправка вопроса жюри

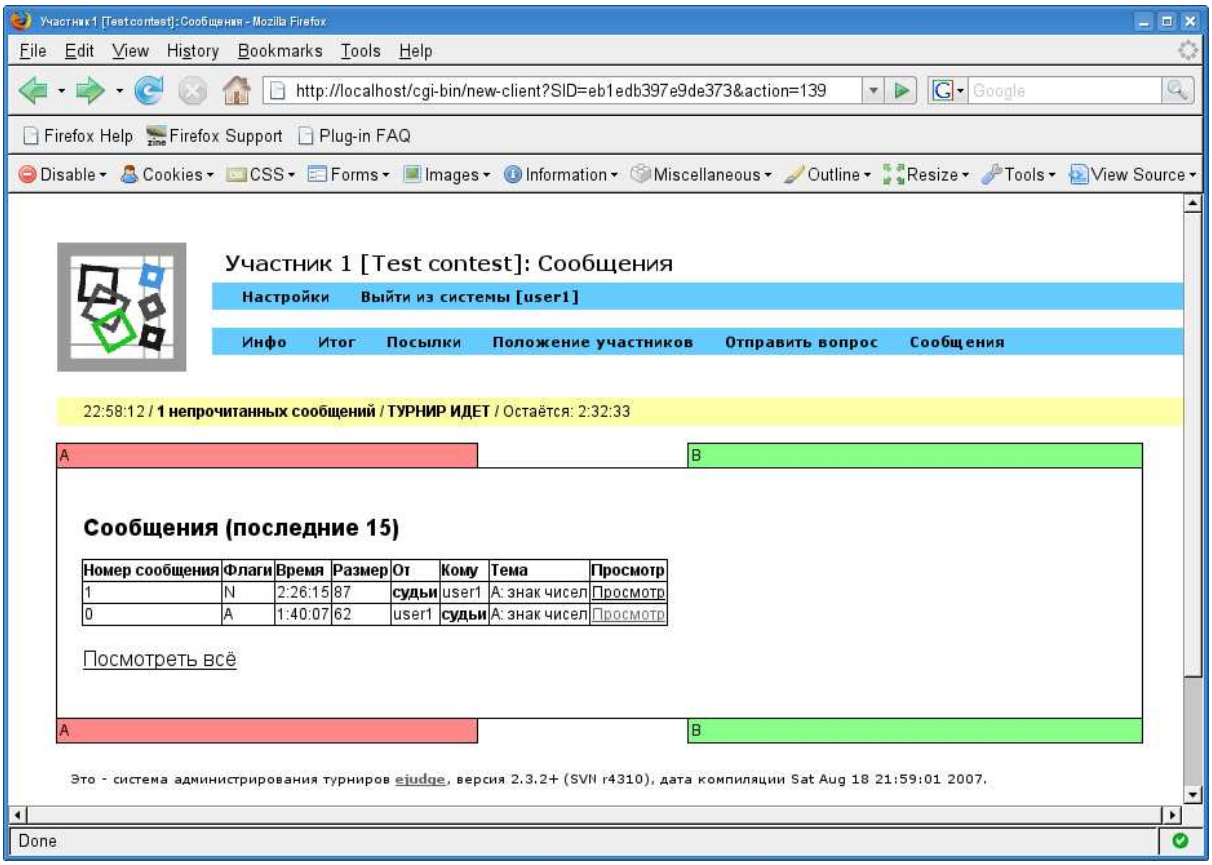


Рис. 3.11: Просмотр сообщений

Столбец **Флаги** может принимать следующие значения:

U	Вопрос еще не был прочитан (для вопросов участника жюри)
R	Вопрос был прочитан, но ответ не ещё не дан (для вопросов участника жюри)
A	На вопрос был дан ответ (для вопросов участника жюри)
N	Новое сообщение участнику от жюри

Таблица 3.2: Возможные значения столбца «Флаги»

Столбец «Время» содержит время отправки сообщения. Время измеряется от начала турнира, но для турниров большой или неограниченной продолжительности администратор турнира может включить режим отображения календарного времени.

«Размер» — это размер сообщения в байтах. Как было сказано выше, размер сообщения от участника жюри ограничен по умолчанию 1024 байтами. Это ограничение может изменяться администратором турнира. Размер сообщения от жюри участнику не ограничен.

Столбец «От» содержит строку **судьи**, если сообщение отправлено жюри участнику, или регистрационное имя участника. Показываются только сообщения либо отправленные данным участником, либо адресованные данному участнику, либо адресованные всем участникам. Столбец «Кому» содержит строку **судьи**, если сообщение отправлено участником жюри, регистрационное имя участника, если сообщение отправлено жюри этому участнику, или строку **все**, если сообщение отправлено жюри всем участникам.

Столбец «Тема» содержит тему сообщения. Нажав на ссылку «Просмотр» можно просмотреть сообщение. Вид страницы просмотра сообщений показан на рис. 3.12.

Как только новое сообщение, адресованное участнику, будет просмотрено, строка состояния турнира снова примет зелёный цвет.

3.3.6 Сдача задач

Нажатием на закладку задачи можно перейти на страницу сдачи решения. Вид страницы сдачи решения зависит от типа задачи. Для стандартных задач, требующих сдачи программ, страница сдачи задач выглядит как показано на рис. 3.13.

На странице сдачи задач может отображаться условие задачи или ссылка на условие задачи. С помощью выпадающего меню «Язык» выбирается язык программирования.

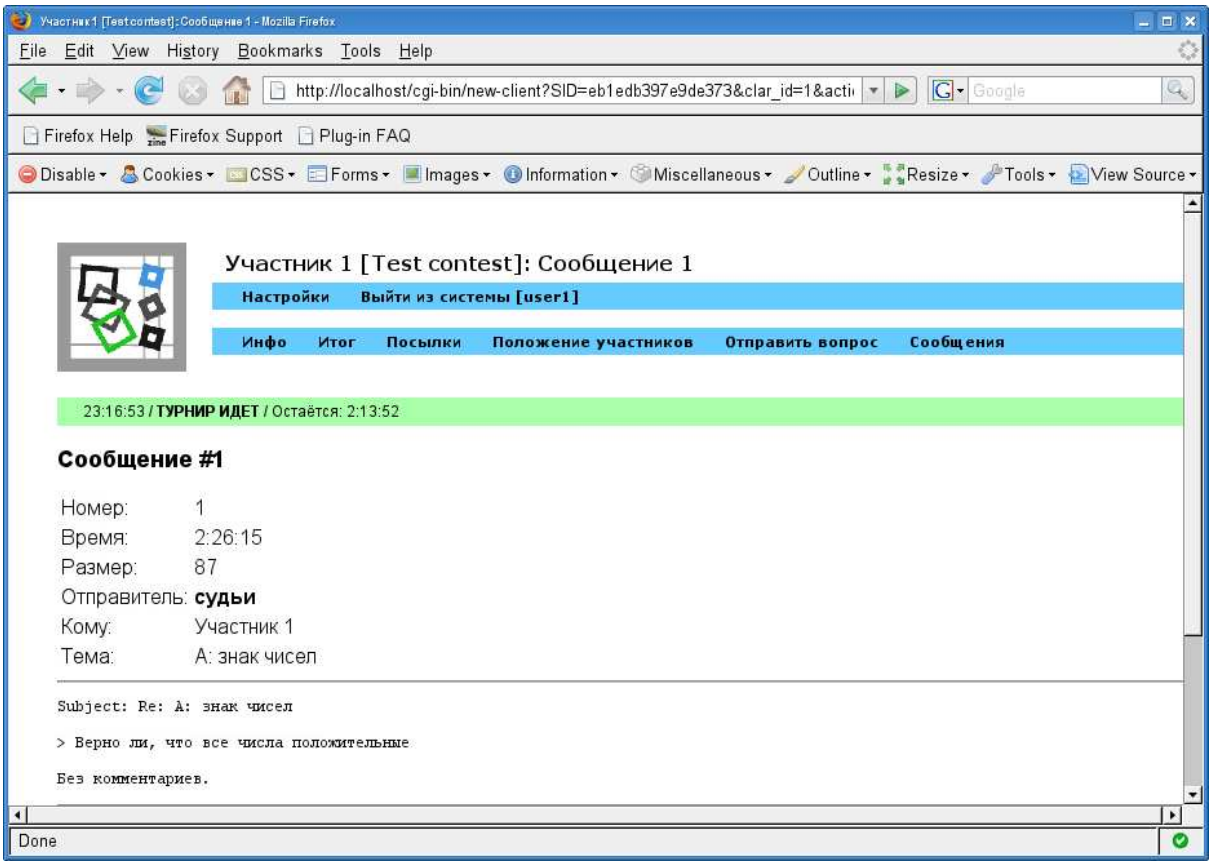


Рис. 3.12: Просмотр сообщения

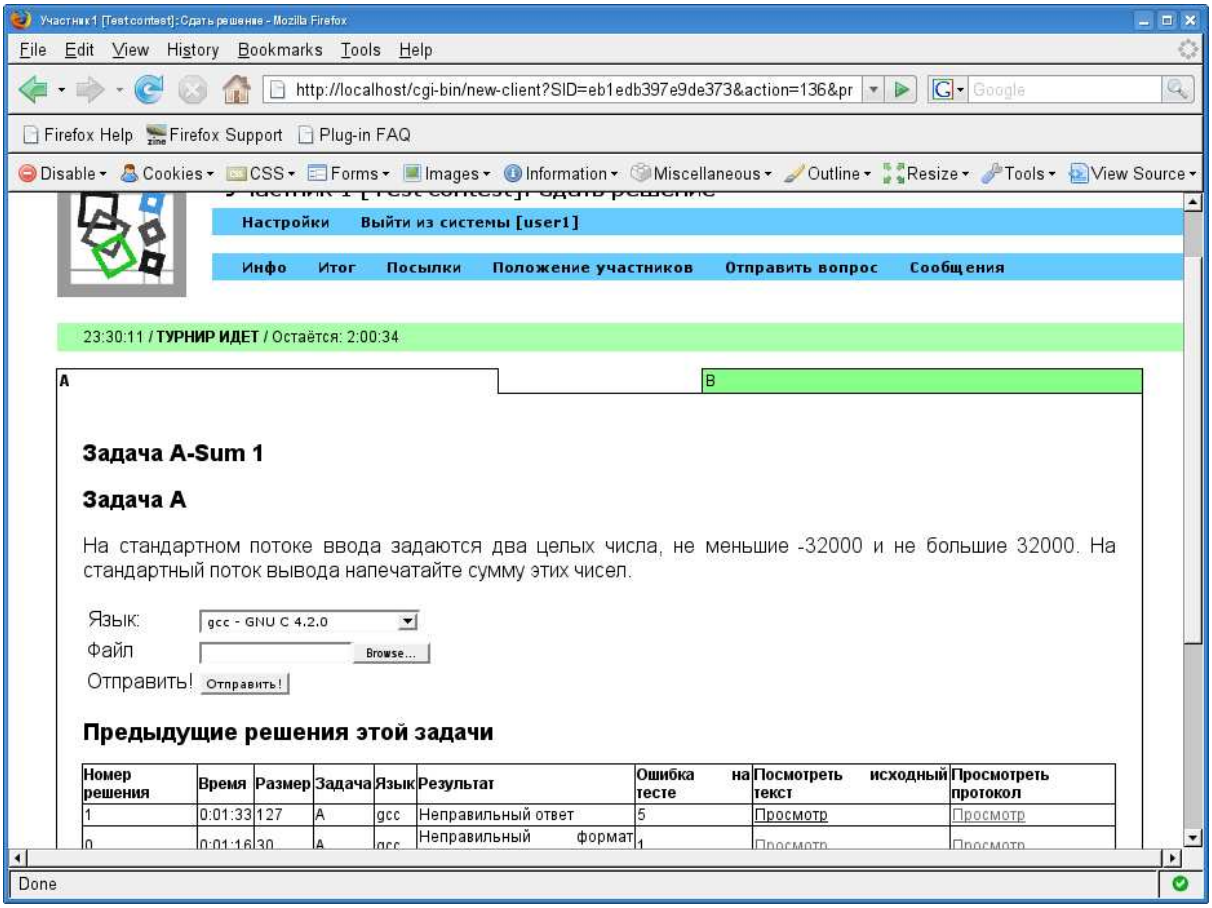


Рис. 3.13: Страница сдачи задач

Глава 4

Оформление решений

При оформлении решений всегда руководствуйтесь требованиям к решению, сформулированным в условии задачи, памятке участника или правилах проведения турнира. Для задач, не предполагающих написание и автоматическое тестирование программ, такие требования, как правило, не вызывают дополнительных трудностей.

В настоящей главе описываются как общие требования к решениям, так и особенности тестирования решений под управлением системы ejudge на ОС Linux. Данная глава будет особенна полезна участникам, программирующим на операционных системах семейства Windows.

4.1 Общие требования

Решение, сдаваемое на проверку, должно представлять собой файл с исходным текстом программы. Файл с исходным текстом должен быть текстовым файлом в 8-битовой кодировке (cp1251 — windows, cp866 — dos, koi8-r, utf8) и не должен содержать байта 0. Это значит, что не могут быть сданы файлы в 16-битной кодировке (например, windows unicode).

Если сдаётся бинарный файл или файл, содержащий нулевой байт, файл не будет принят на проверку, и будет выдано соответствующее сообщение об ошибке.

Решение должно состоять из одного файла с исходным текстом. Проекты, состоящие более чем из одного исходного файла не поддерживаются.

В зависимости от условия задачи программа должна считывать входные данные со стандартного потока ввода (часто в этом случае пишут «считывать с клавиатуры») или из заданного файла. В зависимости от условия задачи результат работы должен выводиться либо на стандартный поток вывода (часто называемый «экраном»), либо в заданный файл. Никакие другие файлы использовать нельзя. Программа не должна создавать диалоговые окна (как графические, так и текстовые), подгружать другие модули и библиотеки и т. п.

4.1.1 Работа со стандартными потоками ввода-вывода

Во всех поддерживаемых языках программирования предусмотрены функции вывода данных на стандартный поток вывода и ввода данных со стандартного потока ввода.

В качестве примера рассмотрим фрагмент программы, который считывает два целых числа в переменные `a` и `b`, производит над ними некоторые вычисления и выводит результат, полученный в переменной `res`.

В программах на языке Си для чтения данных следует использовать функции `scanf`, `getchar` и т. д., а для вывода результата — функции `printf`, `putchar` и т. д.

```
scanf("%d%d", &a, &b);
/* вычисления */
printf("%d\n", res);
```

В программах на языке Си++ можно использовать как функции ввода-вывода языка Си, так и операции чтения из `cin` и записи в `cout`.

```
cin >> a >> b;
// вычисления
cout << res << endl;
```

В программах на языке Паскаль для чтения следует использовать процедуру `read` без файлового параметра или с файлом `input`, а для записи — процедуру `write` без файлового параметра или с файлом `output`.

```
read(a, b);
{ вычисления }
writeln(res);
```

В программах на языке Джава для чтения следует использовать методы работы с потоком `System.in`, а результат выводить в поток `System.out`. Для чтения данных можно использовать класс `Scanner`, но рекомендуется создавать `StreamTokenizer`. Специфика языка Джава более подробно обсуждается ниже.

```
a = s.nextInt();
b = s.nextInt();
// вычисления
System.out.println(res);
```

В программах на языке Бейсик для чтения следует использовать оператор `INPUT`, а для вывода результата — оператор `PRINT`.

```
INPUT a
INPUT b
REM вычисления
PRINT res
```

Обратите внимание, что при чтении данных со стандартного потока ввода ни в коем случае нельзя выводить приглашений ко вводу данных, например, `Введите x:`. Этот текст будет выведен на стандартный поток вывода и «испортит» результат работы программы. Тестирование программы завершится с ошибкой «Неправильный формат вывода» или «Неправильный ответ».

4.1.2 Работа с файлами

Если в формулировке задачи требуется ввод данных из файла и (или) вывод результата в файл, нужно использовать средства работы с текстовыми файлами. Кроме того, в некоторых языках программирования существует возможность привязать стандартные потоки ввода и вывода к файлам.

Предположим, что по условию задачи требуется считывать данные из файла `input.txt` и записывать результат в файл `output.txt`. Обратите внимание, что имена файлов должны записываться в точности так же, как в условии, то есть должен соблюдаться регистр букв и не должны использоваться какие-либо абсолютные или относительные пути.

Таким образом, имя файла `Input.txt` неправильно, так как не совпадает регистр букв, а имя файла `c:\work\input.txt` неправильно, так как содержит элементы пути к файлу, кроме текущего каталога.

В языке Си для открытия файла файла можно использовать функцию `fopen`, а для последующей работы функции `fscanf` и `fprintf`. Вызовы функций `fclose`, в принципе, необязательны, так как при нормальном завершении все открытые файлы закрываются автоматически.

Предположим, что считываются значения двух целых переменных `a` и `b`, над ними проводятся вычисления, и печатается значение переменной `res`.

```
fin = fopen("input.txt", "r");
fout = fopen("output.txt", "w");
fscanf(fin, "%d%d", &a, &b);
/* вычисления */
fprintf(fout, "%d\n", res);
fclose(fin);
fclose(fout);
```

Альтернативный способ заключается в использовании функции `freopen` для привязки стандартных потоков к файлам.

```
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
scanf("%d%d", &a, &b);
/* вычисления */
printf("%d\n", res);
```

Оба способа равнозначны, но второй способ предпочтительнее, так как в первом способе легче ошибиться в тексте программы, перепутав вызовы функций `fprintf` и `printf` или имена файловых переменных.

В языке Си++ можно использовать все возможности языка Си или использовать файловые потоки языка Си++.

```
ifstream fin("input.txt");
ofstream fout("output.txt");
fin >> a >> b;
// вычисления
fout << res << endl;
```

В языке Паскаль для открытия файлов используются процедуры `assign`, `reset` и `rewrite`.

```
assign(fin, 'input.txt'); reset(fin);
assign(fout, 'output.txt'); rewrite(fout);
read(fin, a, b);
{ вычисления }
writeln(fout, res);
close(fin);
close(fout);
```

По аналогии с языком Си можно привязать стандартные файлы `input` и `output`.

```
assign(input, 'input.txt'); reset(input);
assign(output, 'output.txt'); rewrite(output);
read(a, b);
{ вычисления }
writeln(res);
close(input);
close(output);
```

В языке Бейсик файлы открываются следующим образом.

```
open "input.txt" for input as #1
open "output.txt" for output as #2
input #1,a,b
rem вычисления
print #2,res
close #1
close #2
```

4.1.3 Завершение программы

В конце работы программа должна закрыть все открытые файлы и завершиться с кодом возврата 0. Во всех языках, кроме Паскаля и Бейсика все открытые файлы при завершении программы закрываются автоматически. Если в программе на Паскале не закрыть выходной файл, то возможно, что часть выведенной информации окажется потеряна, и, таким образом, результат работы программы окажется неправильным.

Во всех языках программирования кроме Си и Си++ нулевой код завершения программы вырабатывается автоматически при её нормальном завершении. В языках Си и Си++ в конце функции `main` необходимо использовать оператор `return 0`.

4.1.4 Полные примеры программ

Рассмотрим завершенные программы для решения задачи о нахождении суммы двух целых чисел. Предположим, что в файле `input.txt` находятся два целых числа, а их сумму необходимо вывести в файл `output.txt`. Исходные числа по модулю не превосходят 10000.

Программа на языке Си.

```
#include <stdio.h>
int main(void)
{
  int a, b;
  freopen("input.txt", "r", stdin);
  freopen("output.txt", "w", stdout);
  scanf("%d%d", &a, &b);
  printf("%d\n", a + b);
  return 0;
}
```

Обратите внимание, что функция `main` объявлена как возвращающая целое значение. Программа, в которой функция `main` объявлена как не возвращающая значение (void), является неправильной с точки зрения стандарта языка Си. Компиляция такой программы компилятором GNU C завершится ошибкой. Обратите внимание также на оператор `return` в конце функции `main`, который необходим, чтобы программа завершалась с кодом возврата 0.

Программа на языке Си++.

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream fin("input.txt");
ofstream fout("output.txt");
int main(void)
{
  int a, b;
  fin >> a >> b;
  fout << a + b << endl;
  return 0;
}
```

Как и в языке Си, функция `main` должна возвращать целое значение. Для этого в конце функции находится оператор `return`.

Программа на языке Паскаль.

```
  var a,b: integer;
begin
  assign(input,'input.txt'); reset(input);
  assign(output,'output.txt'); rewrite(output);
  read(a,b);
  writeln(a+b);
  close(input);close(output);
end.
```

Программа на языке Бейсик.

```
open "input.txt" for input as #1
open "output.txt" for output as #2
```

```
input #1,a,b
print #2,a+b
close #1
close #2
```

4.2 Проверка программ

Сданное на проверку решение обрабатывается следующим образом.

Исходный текст программы компилируется указанным при посылке компилятором. Если при компиляции возникают ошибки компиляции, проверка решения на этом завершается со статусом «Ошибка компиляции» (“Compilation error”). Время компиляции ограничено, и если компиляция программы потребовала больше времени, чем установленное ограничение, или компилятор зациклился, так же диагностируется ошибка компиляции. Такая ситуация, в частности, может возникнуть при неправильном использовании шаблонов языка Си++.

Как правило, в случае ошибки компиляции участник может просмотреть полный вывод компилятора на стандартный поток вывода и стандартный поток ошибок чтобы установить и устранить причину ошибки. Некоторые особенности используемых в Linux компиляторов, которые могут приводить к ошибкам компиляции, описаны далее. Как правило, попытки сдачи, приведшие к ошибке компиляции, не учитываются при вычислении штрафных баллов участника. Тем не менее все попытки сдачи учитываются при проверке квот участника на количество и общий размер посылок.

Если компиляция программы завершилась успешно, исполняемый файл передаётся на тестирование. Если при компиляции программы компилятор выдал какие-либо диагностические предупреждения, то они теряются, и их текст участнику не доступен.

В зависимости от типа и режима турнира решение может запускаться на всех тестах, на части тестов или до первого непрошедшего теста.

- В турнирах типа ACM и MOSCOW решение запускается на всех тестах до первого теста, который оно не проходит.
- В турнирах типа KIROV решение запускается на всех тестах. После запуска на всех тестах может быть запущена специальная программа-оцениватель, которая по информации о прохождении тестов тестируемой программой выставит окончательный балл.
- В турнирах типа OLYMPIAD в режиме принятия решения на проверку оно запускается только на предварительных тестах до первого непрошедшего, а в режиме окончательного тестирования — на всех тестах. После запуска на всех тестах может быть запущена специальная программа-оцениватель, которая по информации о прохождении тестов тестируемой программой выставит окончательный балл.

Программа считается прошедшей какой-либо тест, если при запуске на этом тесте программа уложилась в ограничения по времени и по памяти, завершилась нормально (с кодом завершения 0) и выдала правильный результат. В остальных случаях программа считается не прошедшей тест.

При запуске программы на каждом тесте выполняются следующие действия. Тестовый файл копируется в рабочий каталог программы с именем, соответствующим условию задачи. Затем запускается тестируемая программа. Во время работы тестируемой программы контролируется ограничение процессорного времени, ограничение реального времени

и ограничение размера используемой памяти. Если тестируемая программа завершилась с кодом возврата 0 и уложилась в ограничения по времени и памяти, запускается проверяющая программа, которая сравнивает ответ, выданный тестируемой программой, с правильным.

Контроль ограничений по памяти. Для 32-битных и 64-битных приложений linux (языки gcc, g++, fpc, dcc и др.), а так же для программ на скриптовых языках, для выполнения которых запускается соответствующая программа-интерпретатор (perl, python, scheme и др.), контролируется максимальный размер всего адресного пространства процесса и размер стека. В адресное пространство процесса входит код программы, код и данные всех используемых программой динамических библиотек, данные программы и стек. Если в процессе работы эти ограничения превышаются, работает модифицированное ядро Linux, и в турнире включена поддержка ошибки ML (Memory limit exceeded), программа будет снята с выполнения с диагностикой ML. Если диагностика этой ошибки не поддерживается, то программа скорее всего завершится с ошибкой RT (Run-time error).

Для 16-битных приложений DOS доступно примерно 500 килобайт памяти, а стек не может превышать 64 килобайт. Эти ограничения обусловлены моделью памяти DOS и специально не контролируются. Диагностика ошибки ML для DOS-программ не поддерживается.

Для программ на языке Java при запуске виртуальной машины устанавливается ограничение на размер кучи (области данных) и размер стека. В дальнейшем контроль этих ограничений производится виртуальной машиной Java. В случае превышения ограничений выбрасывается исключение и программа снимается с выполнения с диагностикой RT.

Ограничения на максимальный объем памяти и максимальный размер стека определяются в условиях задач или в правилах проведения соревнований. По умолчанию максимальный размер стека равен 8 мегабайтам.

Контроль ограничений по времени. Время работы программы на каждом тесте ограничено. Ограничения на время работы программы также определяются либо в условиях задач, либо в правилах проведения соревнований. При выполнении программы контролируется ограничение как процессорного, так и реального времени.

Процессорное время — это время, затраченное процессором на выполнение непосредственно тестируемой программы. Это время складывается из времени работы пользовательского кода (собственно реализованного алгоритма, времени на форматирование вывода и пр.) и времени работы системных функций, вызванных в процессе работы программы (например, операция открытия файла требует некоторого времени на поддержку внутренних структур данных ядра операционной системы). Время, когда процесс не выполнялся, а ожидал завершения операции чтения данных с диска или был приостановлен для выполнения более приоритетного процесса, не вносит вклад в истраченное процессорное время.

Реальное время вычисляется по часам реального времени, поэтому учитывает простои процесса. Как правило, реальное время незначительно отличается от процессорного времени. Однако, если процесс все время ожидает поступления данных или просто «засыпает» на большое время, то процессорное время для него останавливается, и затраченное реальное время будет значительно больше потреблённого. Для предотвращения ситуаций, когда такое решение «подвешивает» процесс тестирования, контролируется реальное время, и если тестируемая программа не завершилась за отведённый ей интервал реального времени, она снимается с выполнения с диагностикой TL (Time-limit exceeded).

Возможна и другая ситуация. Если операционная система достаточно сильно загружена другими процессами, то тестируемая программа будет часто прерываться для выполнения других процессов, и поэтому для неё реальное время может быть больше виртуального. Поэтому ограничение на реальное время устанавливается в 2-3 раза больше ограничения на процессорное время.

Если программа в процессе работы создаёт несколько нитей, то процессорное время всей программы суммируется по всем нитям. Если компьютер поддерживает одновременное выполнение нескольких нитей (например, на многоядерном процессоре или многопроцессорной системе), то реальное время будет в соответствующее количество раз меньше процессорного времени.

Вердикт тестирования. Результатом тестирования программы является один из вердиктов, перечисленных в таблице 4.1. В случае, если возникла ошибка CF (Check failed), обратитесь к администратору турнира. Эта ошибка означает, что при проверке Вашей программы случился сбой в проверяющей системе, из-за чего программа не была проверена.

код	англ.	рус.	комментарий
OK	OK	OK	тест пройден
CE	Compilation error	Ошибка компиляции	
RT	Run-time error	Ошибка выполнения	
TL	Time-limit exceeded	Превышено максимальное время работы	
PE	Presentation error	Неправильный формат вывода	
WA	Wrong answer	Неправильный ответ	
CF	Check failed	Ошибка проверяющей системы	
ML	Memory limit exceeded	Превышен лимит по памяти	
SE	Security violation	Нарушение правил безопасности	

Таблица 4.1: Вердикты тестирования

Тестируемая программа запускается на выполнение, при этом контролируются время её работы, размер используемой ею памяти, выполняемые ей операции. Рассмотрим возможные ситуации, которые могут возникнуть при тестировании, подробнее.

- Программа попыталась выполнить запрещённую операцию, например, удалить файл или создать сетевое соединение. В этом случае программа может быть снята с выполнения с вердиктом «Нарушение правил безопасности». Обратите внимание, что за нарушение правил проведения турнира Вы может быть дисквалифицированы.

- Программа превысила ограничение на размер памяти или стека. В этом случае программа снимается с выполнения и диагностируется ошибка «Превышен лимит по памяти». Возможные причины этого следующие:

  – Бесконечная рекурсия в программе и, как следствие, переполнение стека.

  – Объявлен слишком большей локальный или глобальный массив.

  – В процессе работы программа запрашивает слишком много динамической памяти.

  Для программ, выполняющихся в среде DOS, превышение ограничения по памяти не диагностируется. Например, программа, которая входит в бесконечную рекурсию, скорее всего завершится с ошибкой «Ошибка выполнения».

- Программа превысила ограничение на процессорное время. В этом случае программа снимается с выполнения и диагностируется ошибка «Превышено максимальное время работы». Возможные причины этого следующие:

  – Из-за ошибки в программе она зациклилась.

  - В программе реализован неэффективный алгоритм, из-за которого она не укладывается в ограничения по времени на больших входных данных.

- Программа превысила ограничение на реальное время. В этом случае программа снимается с выполнения и диагностируется ошибка «Превышено максимальное время работы». Возможные причины этого следующие:

  - Вместо чтения из файла программа ожидает ввода с клавиатуры.

- В процессе выполнения программы возникла ошибка выполнения, такая как выход за пределы массива, обращение по несуществующему адресу или деление на ноль. В этом случае программа снимается с выполнения и диагностируется «Ошибка выполнения». Обратите внимание, что в некоторых случаях ошибка выполнения может быть следствием превышения программой максимального размера памяти.

- Программа завершается с ненулевым кодом возврата в результате выполнения процедуры `halt(n)` в Паскале или `exit(n)` в Си и Си++. В этом случае диагностируется «Ошибка выполнения».

Если программа была снята с выполнения или завершилась по любой из перечисленных выше причин, вывод программы не проверяется. Если программа отработала корректно, то выходной файл, сгенерированный программой, проверяется на правильность.

Если выходной файл, требуемый по условию задачи, отсутствует или пуст, диагностируется «Неправильный формат вывода». Если содержимое выходного файла не соответствует требованиям к формату результата, также диагностируется «Неправильный формат вывода».

Если выходной файл оформлен в соответствии с требованиями к задаче, но ответ не совпадает с правильным, диагностируется «Неправильный ответ».

4.3 Особенности языков и сред программирования

Среда **Delphi**. Среде Delphi в ОС Linux соответствует компилятор Kylix. Последняя версия Kylix была выпущена в 2003 году и примерно соответствует Delphi 6. После этого Kylix не обновлялся и более не поддерживается фирмой Borland.

Для того, чтобы программа на Delphi компилировась под Kylix, имена стандартных модулей должны писаться в правильном регистре букв, например `Math`, `SysUtils`. В противном случае будет диагностирована ошибка компиляции программы.

Компиляторы **gcc** и **g++**. В среде Unix в качестве разделителей каталогов в пути используется только символ `/`, в отличие от Windows, где допускается как `/`, так `\`. Поэтому в директивах `#include` должен использоваться только символ `/`.

Неправильно:

```
#include <sys\types.h>
```

Правильно:

```
#include <sys/types.h>
```

C, C++: 64-битные целые типы. В среде Visual C++ для определения 64-битных целых типов использовался тип `__int64`. Visual C++ начиная с версии 2005 поддерживает и стандартный 64-битный тип `long long`, но для совместимости или по инерции `__int64` до сих пор широко используется. Для максимальной совместимости со старыми версиями Visual C++ следует определить пользовательский тип `int64_t` следующим образом:

```
#ifdef _MSC_VER
#define int64_t __int64
#else
#define int64_t long long
#endif
```

Для беззнаковых 64-битных целых чисел аналогичным образом можно определить тип `uint_64`.

Обратите внимание, что компиляторы GCC, работающие на Windows (в среде Cygwin или MinGW), поддерживают тип long long.

C, C++: 64-битные целые типы: ввод-вывод. Для ввода и вывода 64-битных целых чисел в Windows используется нестандартный спецификатор формата `I64`, а в Linux — стандартный спецификатор `ll`. Например, вывод одного 64-битного целого числа на стандартный поток вывода для всех Windows компиляторов (кроме среды Cygwin) записывается следующим образом:

```
printf("%I64d", val);
```

А в Linux и других системах (в частности, и в Windows в среде Cygwin):

```
printf("%lld", val);
```

Поэтому, для вывода 64-битных целых чисел переносимым образом необходимо использовать условную компиляцию, например, следующим образом:

```
#ifdef _WIN32
  printf("%I64d\n", val);
#else
  printf("%lld\n", val);
#endif
```

Последние версии компилятора Visual Studio поддерживают и спецификацию формата `ll`, но не MinGW.

`C, C++: тип long double`. В среде Visual C++ тип **`long double`** занимает такой же размер и имеет такую же точность, как и тип **`double`**. Не существует простого способа использовать 80-битовые вещественные числа расширенной точности.

Компиляторы GCC под Windows поддерживают 80-битный тип `long double` для вычислений, но отсутствуют средства ввода и вывода таких чисел (так как они отсутсвуют в стандартной библиотеке MSVCRT.DLL).

В Linux, FreeBSD и другие POSIX-системах тип **`long double`** — это 80-битные вещественные числа, для ввода и вывода которых можно использовать спецификатор формата `L` или средства потокового ввода-вывода C++, например:

```
printf("%.10Lg\n", x);
```

C++: потоковый ввод и вывода. Потоковый ввод и вывод языка C++ примерно в 10-20 раз медленнее функций `printf` и `scanf`. Если требуется ввести или вывести большое количество данных, используйте функции ввода-вывода языка C.

C, C++: 64-битный режим. Если программы отлаживаются на 32-разрядном компиляторе, а проверяться будут на 64-разрядном или наоборот, или программы отлаживаются в

тип	16 бит (bcc, bpc)	32 бита	64 бита (MSVC)	64 бита (GCC)
char	8	8	8	8
short	16	16	16	16
int	16	32	32	32
long	32	32	32	64
long long	-	64	64	64
void*	32	32	64	64

Таблица 4.2: Размеры базовых типов (в битах) в разных режимах компиляции

64-разрядном режиме Visual C++, а проверятся будут в 64-разрядном режиме gcc, учтите разные размеры базовых типов. Они приведены в таблице 4.2.

Среда **QBASIC**. По умолчанию среда сохраняет программы в некотором внутреннем (не текстовом) формате, который не поддерживается системой ejudge. При сохранении файла убедитесь, что он сохраняется как текстовый.