

Система проведения соревнований ejudge

Руководство участника турнира

1 Ввод регистрационных данных

Введите в адресную строку браузера адрес страницы входа в систему регистрации.

На рис. 1 приведен вид страницы, которая будет отображена.

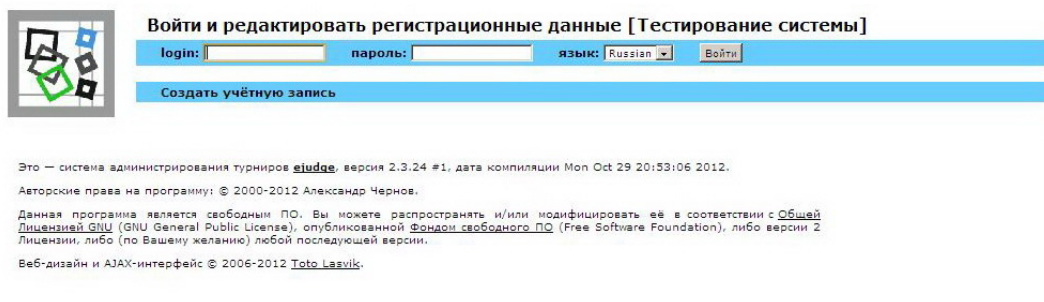


Рис. 1: Страница входа в систему регистрации

Эта страница содержит следующие элементы:

- Поля ввода **Login** и **Password (Пароль)**. Для входа в систему введите в эти поля выданные вам логин и пароль и нажмите на кнопку **Log in (Войти)**.
- **Выпадающее меню Language (Язык)**, которое позволяет выбрать язык отображения страниц.

Если login и пароль были указаны правильно, будет отображена страница просмотра информации об участнике (рис. 2).

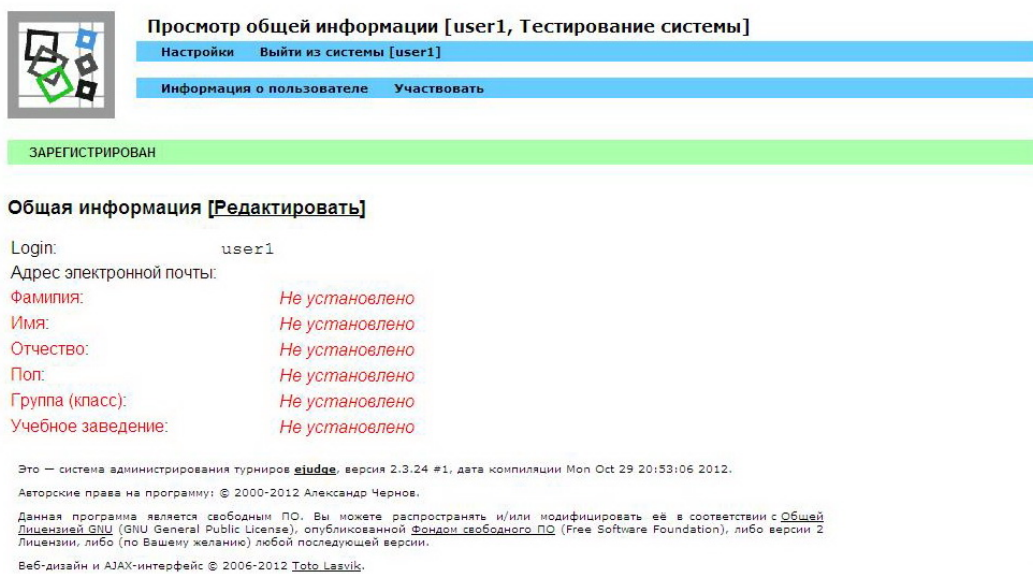


Рис. 2: Страница просмотра информации

Для заполнения недостающих данных пройдите по ссылке «Редактировать». На странице редактирования информации об участнике (рис. 3) заполните все требуемые поля и нажмите кнопку «Сохранить».

Система вернет вас на страницу просмотра информации, которая теперь будет выглядеть так, как показано на рис. 4.

Для входа в систему теперь достаточно пройти по ссылке «Участвовать». Данные будут сохранены в системе и в следующий раз, когда вам понадобится войти в турнир, можно будет использовать страницу входа в систему без заполнения регистрационных данных.

Отнеситесь внимательно к заполнению регистрационных данных. Аккаунты с некорректно заполненной информацией могут быть заблокированы.

2 Вход в систему

Введите в браузере в адресную строку адрес страницы входа.

На рис. 5 приведен вид страницы входа в систему, которая будет отображена.

Страница входа в систему содержит следующие элементы:

РЕДАКТИРОВАНИЕ ОБЩЕЙ ИНФОРМАЦИИ [user1, Тестирование системы]

ЗАРЕГИСТРИРОВАН

Общая информация

Фамилия: должно быть заполнено

Имя: должно быть заполнено

Отчество: должно быть заполнено

Пол: должно быть заполнено

Группа (класс): должно быть заполнено

Учебное заведение: должно быть заполнено

Это — система администрирования турниров **ejudge**, версия 2.3.24 #1, дата компиляции Mon Oct 29 20:53:06 2012.

Авторские права на программу: © 2000-2012 Александр Чернов.

Данная программа является свободным ПО. Вы можете распространять и/или модифицировать её в соответствии с [Общей Лицензией GNU](#) (GNU General Public License), опубликованной [Фондом свободного ПО](#) (Free Software Foundation), либо версии 2 Лицензии, либо (по Вашему желанию) любой последующей версии.

Веб-дизайн и AJAX-интерфейс © 2006-2012 [Toto Lasvik](#).

Рис. 3: Страница редактирования данных

ПРОСМОТР ОБЩЕЙ ИНФОРМАЦИИ [user1, Тестирование системы]

Настройки Выйти из системы [user1]

Информация о пользователе Участвовать

ЗАРЕГИСТРИРОВАН

Общая информация [Редактировать]

| | |
|--------------------------|-----------------------|
| Login: | user1 |
| Адрес электронной почты: | user1@...ru |
| Фамилия: | Иванов |
| Имя: | Петр |
| Отчество: | Сергеевич |
| Пол: | Мужской |
| Группа (класс): | 10-Б |
| Учебное заведение: | школа №13 г. Енакиево |

Это — система администрирования турниров **ejudge**, версия 2.3.24 #1, дата компиляции Mon Oct 29 20:53:06 2012.

Авторские права на программу: © 2000-2012 Александр Чернов.

Данная программа является свободным ПО. Вы можете распространять и/или модифицировать её в соответствии с [Общей Лицензией GNU](#) (GNU General Public License), опубликованной [Фондом свободного ПО](#) (Free Software Foundation), либо версии 2 Лицензии, либо (по Вашему желанию) любой последующей версии.

Веб-дизайн и AJAX-интерфейс © 2006-2012 [Toto Lasvik](#).

Рис. 4: Страница просмотра информации с заполненными данными

- Поля ввода **Login** и **Password (Пароль)**. Эти поля предназначены для аутентификации пользователя. Для начала работы в турнире необходимо ввести в эти поля выданные вам логин и пароль и нажать на кнопку **Log in (Войти)**.
- **Выпадающее меню Language (Язык)**, которое позволяет выбрать язык отображения страниц. В системе ejudge поддерживается два языка: English (Английский) и Russian (Русский). Желаемый язык должен быть выбран до нажатия кнопки Log in.

Если login и пароль были указаны правильно, будет отображена страница участника турнира, которая подробно рассмотрена в следующем разделе.

Если регистрационные данные не были заполнены, вам будет выдано сообщение об этом (см. рис. 6).

Пройдите по ссылке «вернитесь к заполнению» и заполните информацию об участнике, следуя инструкциям, описанным в предыдущем разделе.

Если же вход участника в турнир невозможен, отображается страница, показанная на рис. 7.

Доступ участника к турниру может быть закрыт по многим причинам, которые перечислены на выводимой странице. Но 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.

3 Информация о турнире

Предположим, что участник успешно авторизовался. После авторизации участник попадает на страницу информации о турнире. Если турнир еще не начался, страница информации о турнире может иметь вид, показанный на рис. 8.

На странице отображается имя участника (на данном рисунке – это admin), название турнира (Тестирование системы). Кроме того, в строке состояния (строка зеленого цвета) отображается текущее время на компьютере участника и состояние

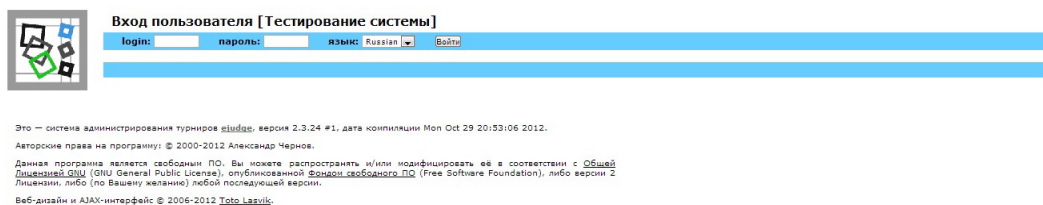


Рис. 5: Страница входа в систему ejudge

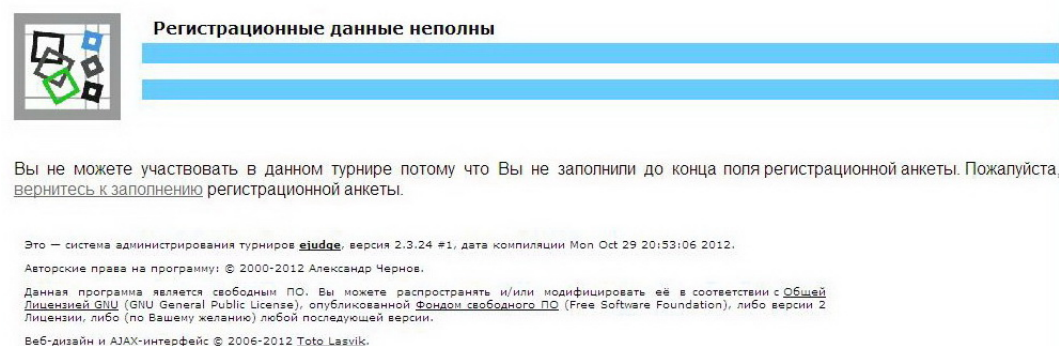


Рис. 6: Сообщение о неполной регистрации

турнира (НЕ НАЧАЛСЯ). Дополнительно отображается статистика об участниках турнира.

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

4 Начало турнира

Как правило турнир стартует в момент астрономического времени, указанный администратором заранее. Время старта указывается в строке состояния турнира как показано на рис. 9.

Когда настанет время начала турнира, показанное в строке состояния турнира, или когда будет объявлено, что турнир начался, участник турнира должен нажать кнопку «Refresh» браузера. После этого страница информации о турнире будет перезагружена и примет вид, показанный на рис. 10.

В дополнение к уже рассмотренным пунктам меню «Настройки», «Выход из системы», «Инфо» появились пункты меню «Итог», «Посылки», «Отправить вопрос», «Сообщения».

Строка состояния турнира теперь содержит текущее время сервера, которое обновляется каждую секунду, сообщение «ТУРНИР ИДЕТ» и время, оставшееся до конца турнира, которое также обновляется каждую секунду. На рабочем поле страницы появились закладки, соответствующие названиям задач («А» и «В»). Изменилась информация о турнире, которая теперь содержит следующую информацию:

- Состояние турнира («Турнир идет»). Кроме того, здесь отображается дополнительная информация о состоянии турнира, например, информация о приостановке проверки решений 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.
- Время, прошедшее от начала турнира, и время, остающееся до конца турнира.
- Статистика об посещаемости турнира.

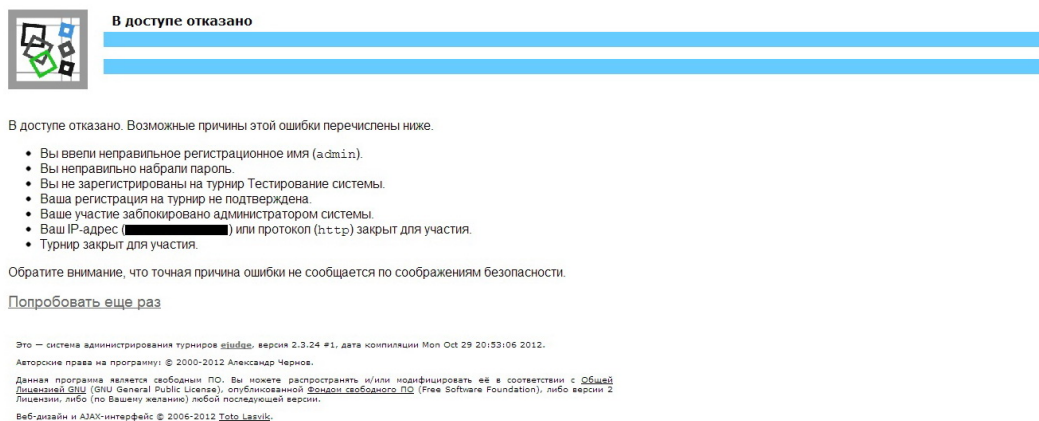


Рис. 7: Страница, отображаемая при неудачном входе

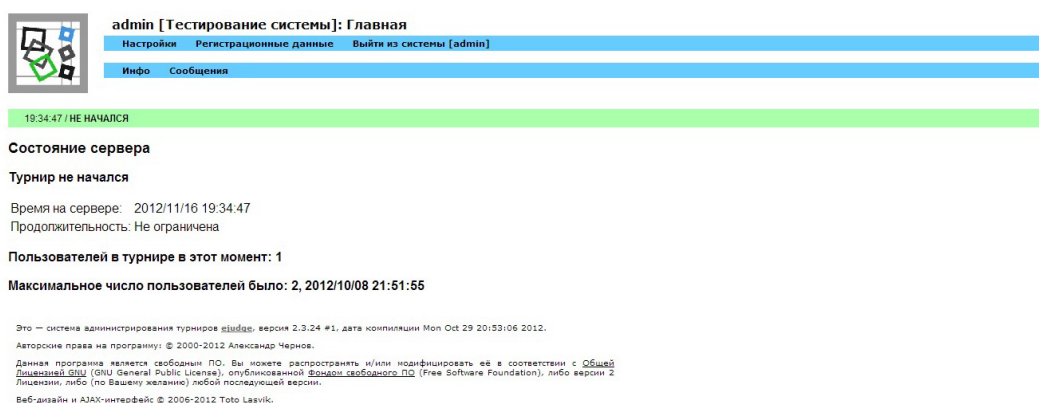


Рис. 8: Страница после успешного входа в турнир

5 Действия во время турнира

На рис. 10 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:

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

При выборе пункта меню «Итог» отображается страница результатов проверки каждой задачи. Пример страницы результатов проверки показан на рис. 11

Обратите внимание, что закладка задачи «В» имеет красный цвет, поскольку задача В не полностью решена, а закладка задачи «А» — зеленая. В этой таблице отображается количество баллов, полученных за каждую задачу, и номер посылки, которая будет давать вклад по данной задаче в окончательный результат участника (лучшая попытка).

При выборе пункта меню «Посылки» отображается журнал посылок данного участника. Пример страницы журнала посылок показан на рис. 12.

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 послылки всех участников турнира нумеруются последовательно, начиная с 0. Номер посылки является ее 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.

Столбец «Размер» — это размер посылки в байтах.

Столбец «Задача» — это короткое имя (идентификатор) задачи.

Столбец «Язык» — это идентификатор языка программирования, который был использован при решении задачи.

Столбец «Результат» — это результат проверки посылки.

Столбец «Пройдено тестов» содержит количество тестов, на которых тестируемая программа выдала правильный результат.

Столбец «Баллы» содержит количество набранных баллов.

Рис. 9: Строка состояния турнира, показывающая время его начала

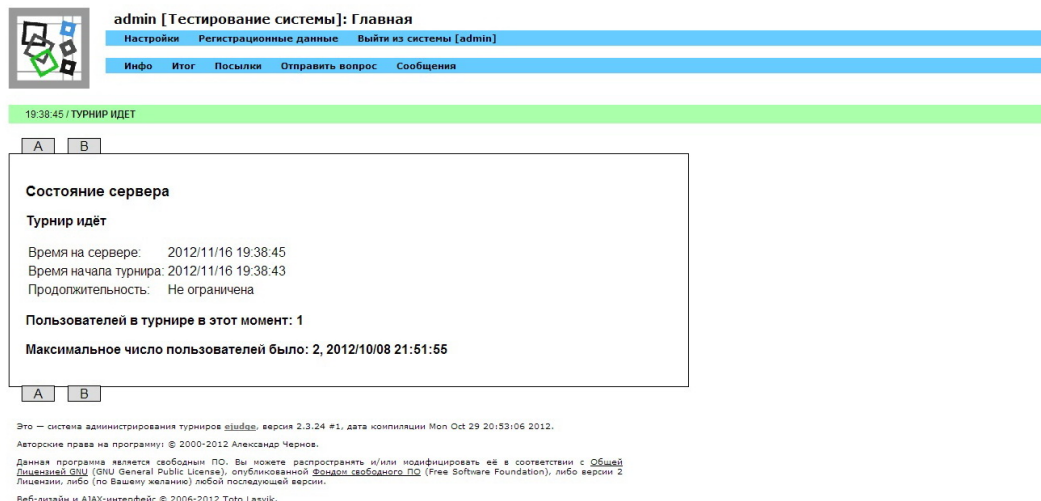


Рис. 10: Страница информации о турнире после его начала

Нажав на ссылку «Просмотр» в столбце «Посмотреть исходный текст» можно посмотреть на исходный текст данной посылки.

Нажав на ссылку «Просмотр» в столбце «Посмотреть протокол» можно открыть страницу просмотра протокола тестирования данной посылки. Примерный вид протокола тестирования показан на рис. 13.

Протокол тестирования содержит информацию о прохождении тестируемой программы на тестах. 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. 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.

При выборе пункта меню «Отправить вопрос» отображается страница посылки вопроса жюри, показанная на рис. 14. В выпадающем меню «Задача» можно указать задачу, по которой задается вопрос, а в поле «Тема» сформулировать тему вопроса. Затем в поле ввода набирается собственно вопрос. Общая длина вопроса ограничена 1024 символами. Для отсылки вопроса необходимо нажать на кнопку «Отправить».

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 желтый цвет, и в ней появляется надпись n непрочитанных сообщений (как показано на рис. 15). Участник может нажать на ссылку «Сообщения» и перейти на страницу просмотра сообщений, показанную на рис. 15.

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.

Столбец «Флаги» может принимать следующие значения:

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

Столбец «Время» содержит время отправки сообщения. Время измеряется от начала турнира.

«Размер» – это размер сообщения в байтах. Как было сказано выше, размер сообщения от участника жюри ограничен 1024 байтами. 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. 16.

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

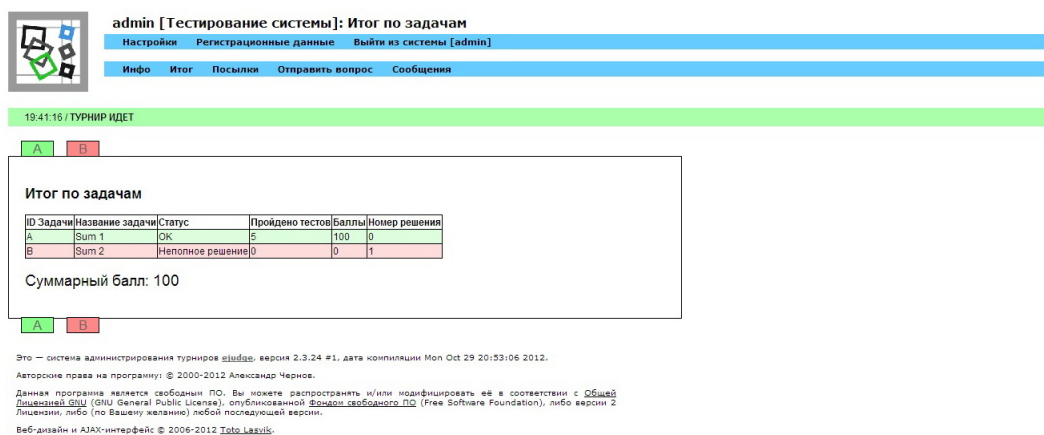


Рис. 11: Страница суммарной информации по задачам

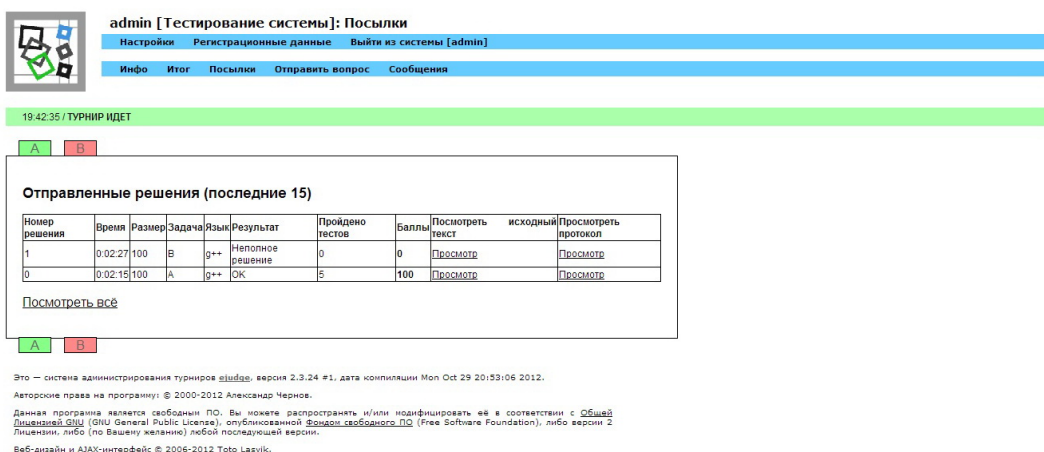


Рис. 12: Страница журнала посылок

6 Сдача задач

Нажатием на закладку задачи можно перейти на страницу сдачи решения. Вид страницы сдачи решения выглядит как показано на рис. 17. На странице сдачи задач может отображаться условие задачи или ссылка на условие задачи.

Чтобы послать задачу на проверку, вам нужно с помощью выпадающего меню «Язык» выбрать язык программирования, на котором вы писали решение (доступные языки – Pascal, C, C++) и через меню выбора (кнопка Browse) файла выбрать файл с исходным текстом вашей программы. После этого для отправки решения необходимо нажать кнопку «Отправить».

Для обработки Вашей задачи системе требуется некоторое время. Чтобы узнать вердикт системы на посланную Вами задачу, через какое-то время обновите страничку в вашем браузере и Вам будет показан вердикт системы по посланной задаче.

Учтите, что количество попыток по каждой задаче не должно превышать 5 (попытки, получившие вердикт Compilation Error, не учитываются). Все последующие попытки системой приниматься не будут. В зачет идет лучшая из попыток.

Компиляторы, установленные в системе для каждого языка указаны в следующей таблице.

| | |
|--------|---------------------|
| Pascal | Free Pascal 2.4.0-2 |
| C | GNU C 4.4.5 |
| C++ | GNU C++ 4.4.5 |

Все указанные компиляторы работают в среде Linux. Особенности, которые следует учесть при написании ваших программ, будут описаны последующих разделах.

7 Оформление решений

При оформлении решений всегда руководствуйтесь требованиям к решению, сформулированным в условии задачи, памятке участника или правилах проведения турнира.

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 сдается 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. Программа должна считывать входные данные со стандартного потока ввода. Результат работы



Рис. 13: Просмотр протокола тестирования



Рис. 14: Отправка вопроса жюри

должен выводиться на стандартный поток вывода. Никакие файлы использовать нельзя. Программа не должна создавать диалоговые окна (как графические, так и текстовые), подгружать другие модули и библиотеки и т.п.

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);
```

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».

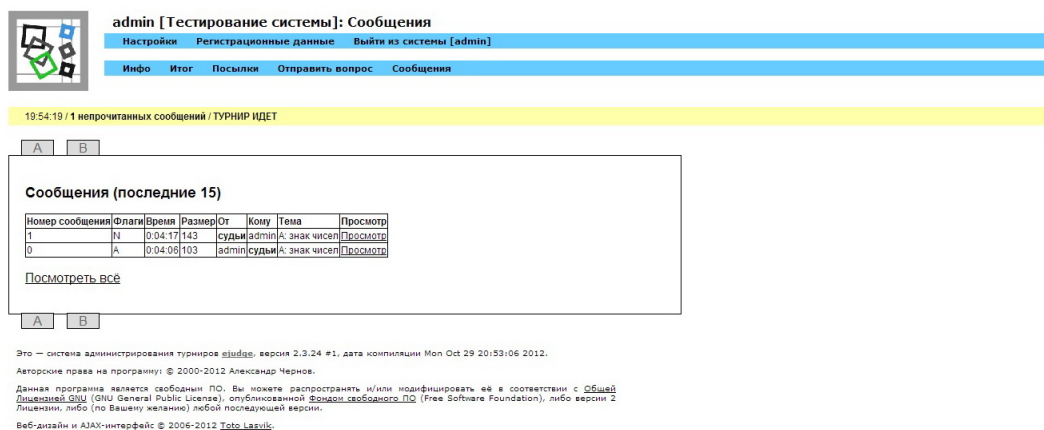


Рис. 15: Просмотр сообщений

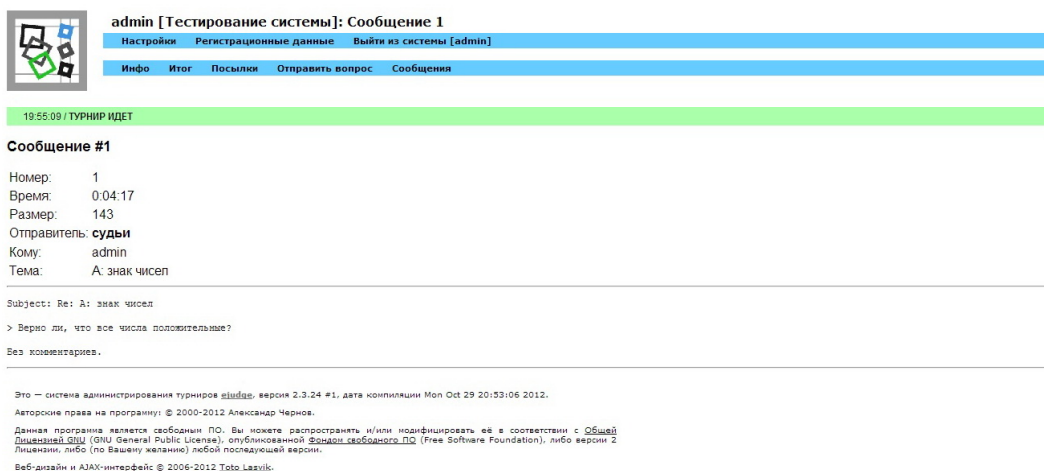


Рис. 16: Просмотр сообщения

8 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 заиклился, так же диагностируется ошибка компиляции. Такая ситуация, в частности, может возникнуть при неправильном использовании шаблонов языка Си++.

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. Попытки сдачи, приведшие к ошибке компиляции, не учитываются. Если компиляция программы завершилась успешно, исполняемый файл передается на тестирование. Если при компиляции программы компилятор выдал какие-либо диагностические предупреждения, то они теряются, и их текст участнику не доступен.

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. Затем запускается тестируемая программа на вход которой подаются данные этого теста. 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.

Для приложений 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, программа будет снята с выполнения с диагностикой ML (Memory limit exceeded).

Время работы программы на каждом тесте ограничено. Ограничения на время работы программы также определяются в условиях задач. 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 си-

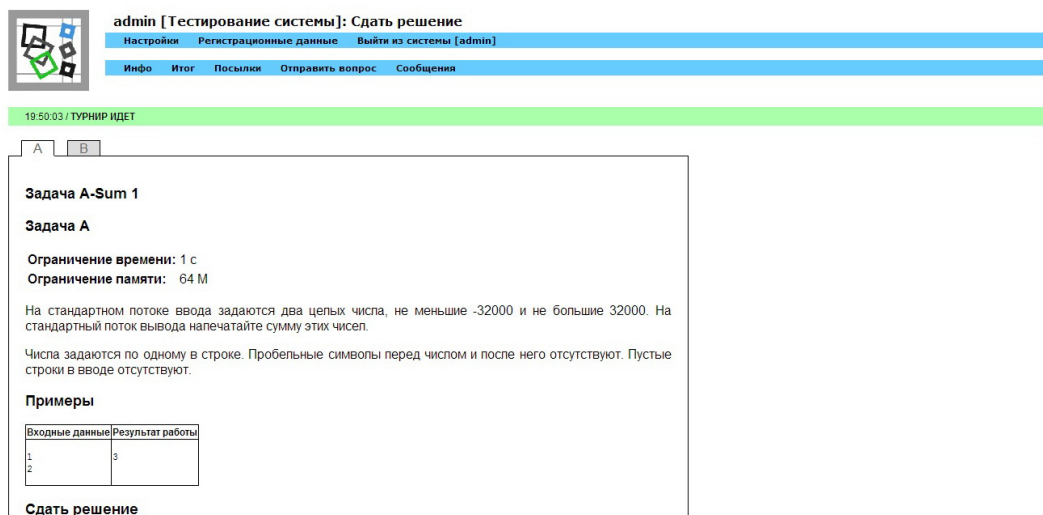


Рис. 17: Страница сдачи решения

стемы). 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 потребленного.

Для предотвращения ситуаций, когда такое решение “подвешивает” процесс тестирования, контролируется реальное время, и если тестируемая программа не завершилась за отведенный 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 нее 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 создает 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.

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

| код. | англ. | рус. |
|------|-----------------------|-------------------------------------|
| OK | OK | OK |
| 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 |

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

- Программа попыталась выполнить запрещенную 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 можете 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 большой локальный или глобальный массив.
 - В процессе работы программа запрашивает слишком много динамической памяти.
- 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 заиклилась.

- 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”.
- 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 вывод, сгенерированный программой, проверяется на правильность. Если вывод отсутствует, диагностируется “Неправильный формат вывода”. Если вывод не соответствует требованиям к формату результата, также диагностируется “Неправильный формат вывода”. Если вывод оформлен в соответствии с требованиями к задаче, но ответ не совпадает с правильным, диагностируется “Неправильный ответ”.

9 Особенности языков и сред программирования

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>
```

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. Компиляторы GCC поддерживают только тип `long long`, который и следует использовать при необходимости.

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 записывается следующим образом:

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

А в Linux и других системах:

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

При сдаче программ в систему `ejudge` следует использовать только второй вариант.

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 отсутствуют 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);
```

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.