

Інструкція по користуванню системою ejudge

1. Вступ

Цей документ описує роботу з системою ejudge звичайного (непривілейованого) користувача. Звичайний користувач — це звичайний учасник олімпіади, турніру чи екзамену, який виконує запропоновані йому завдання згідно правил турніру.

Система ejudge допускає широкий діапазон налаштувань, в залежності від яких інтерфейс користувача може дещо відрізнятися. У цьому документі описуються всі можливості, що надаються користувачеві. Залежно від налаштувань системи частина можливостей може бути недоступною. У залежності від специфіки заходу, він може називатися «змагання», «турнір», «олімпіада», «конкурс», «іспит» і т. п. У подальшому завжди буде вживатися назва «турнір» для всіх типів змагань. Особа, яка бере участь у турнірі, буде називатися «учасник» турніру.

2. Вхід до турніру

У цьому розділі розглядається процедура входу в турнір.



рис. 2.1: Сторінка входу в систему (англійська версія)

На рис. 2.1 наведений вид сторінки входу в систему для англійської версії програми, а на рис. 2.2 - для російської версії програми.

Сторінка входу в систему містить такі елементи:

- Поля введення Login (Логін) і Password (Пароль). Ці поля призначені для аутентифікації користувачів, вже зареєстрованих на турнір. Для початку роботи в турнірі необхідно заповнити ці поля і натиснути на кнопку Log in (Увійти).
- Випадаюче меню Language (Мова), яке дозволяє вибрати мову відображення сторінок. У поточній версії системи ejudge підтримується дві мови: English (Англійська) і Russian (Російська). Бажана мова обирається до натискання кнопки



рис. 2.2: Сторінка входу в систему (російська версія)

Log in. Основна мова (мова за замовчуванням) турніру встановлюється адміністратором турніру. Якщо основну мову не встановлено, нею вважається англійська мова. Меню, що випадає Language може бути відсутнім, якщо адміністратором турніру заборонено змінювати мову відображення сторінок.

У рядку введення пароля Ви повинні ввести правильний пароль участі в турнірі. У системі ejudge підтримується два види пароля: пароль для редагування реєстраційних даних (так званий реєстраційний пароль) і пароль участі в турнірі. При створенні нового користувача системою генерується саме реєстраційний пароль, який потім може використовуватися для редагування інформації про користувача. Пароль участі в турнірі може генеруватися адміністратором турніру вже після закінчення реєстрації всіх користувачів і перед початком турніру.

У більшості випадків (як правило, у всіх відкритих турнірах) пароль участі в турнірі автоматично збігається з реєстраційним паролем. Однак особливо в очних турнірах пароль участі в турнірі може бути відмінним від реєстраційного пароля. Іншими словами, якщо учасникам безпосередньо перед початком турніру були роздані паролі для участі, вводиться повинні саме вони, а не реєстраційні паролі, які, можливо, використовувалися учасником під час реєстрації.

Зверніть увагу, що для появи сторінки входу в турнір в адресному рядку браузера необхідно вказати адреси, що містить так званий ідентифікатор турніру (contest_id). Ідентифікатор турніру призначається адміністратором турніру при його створенні. Як правило, вхід в турнір відбувається з іншої веб-сторінки, в якій на посиланні, що веде на сторінку входу в турнір, вже вказано ідентифікатор турніру. У такому випадку зверніться до адміністратора турніру і уточніть у нього адресу веб-сторінки для входу в турнір.

Якщо логін і пароль були вказані правильно і користувач має повноваження брати участь у турнірі, буде відображена сторінка учасника турніру, яка детально розглядається в подальших розділах. Якщо ж вхід учасника в турнір неможливий, відображається сторінка, показана на рис. 2.3.

Доступ учасника до турніру може бути закритий з багатьох причин, які перераховані на виведеній сторінці. Найбільш часта причина, звичайно ж, це неправильно набраний пароль. Точна причина, по якій доступ користувача до турніру не можливий, не повідомляється щоб утруднити несанкціоноване проникнення в систему. Тому, якщо при спробі входу в турнір виникла помилка, спочатку спробуйте повторити набір реєстраційного імені та пароля, а якщо це не допомагає, зверніться до адміністратора турніру.



рис. 2.3: Вид сторінки, яка відображається при неуспішному вході в турнір

- Неправильно введене реєстраційне ім'я (login). Необхідно перевірити, що при введенні реєстраційного імені не були переплутані великі і малі літери, так як реєстраційні імена чутливі до регістру букв. Для полегшення перевірки введене реєстраційне ім'я відображається в дужках.
- Неправильно введений пароль. Перевірте правильність введення пароля. На відміну від реєстраційного імені введений пароль не відображається. Перевірте, що вводиться дійсно пароль участі в турнірі, якщо це потрібно.
- Відсутня реєстрація на вказаний турнір. Реєстраційне ім'я та пароль вказані правильно, але даний користувач не зареєстрований на турнір. Якщо турнір відкритий або модерується, то поверніться назад на сторінку входу в турнір і зверніться за посиланням Реєстрація. Якщо турнір закритий, зверніться до адміністратора турніру.
- Непідтверджена реєстрація на турнір. Реєстраційне ім'я та пароль вказані правильно, користувач зареєструвався на турнір, але цей турнір - модерується, і адміністратор турніру не підтвердив участь у ньому. Щоб перевірити статус реєстрації на модерується турнір поверніться назад на сторінку входу в турнір і зверніться за посиланням Реєстрація.
- Участь заблоковано. Незважаючи на те, що реєстраційне ім'я та пароль вказані правильно, користувач зареєстрований на турнір і його реєстрація підтверджена, але адміністратор заблокував доступ цьому користувачеві («забанили» його) за будь-які неприпустимі з точки зору адміністратора проступки.
- Неправильний IP-адреса або протокол. IP-адресу користувача закритий для доступу до турніру адміністратором. Крім того, можливо, що потрібен доступ по протоколу https. IP-адреса клієнта і ім'я протоколу (http, https) виводяться для полегшення діагностики. Слід зауважити, що дана причина, хоча і можлива, але малоімовірна.
- Закритий турнір. Турнір вже закінчився і тому закритий для участі адміністратором. Слід зауважити, що завершення турніру саме по собі не означає, що він стає закритий для входу учасників. Учасники можуть дивитися таблицю результатів, протоколи тестування, здані ними рішення, подавати апеляції і

т. п. Лише після того, як адміністратор визнає, що подальший вхід користувачів в цей турнір недоцільний, він може його закрити.

3. Участь у турнірі

У даній главі розглядається інтерфейс учасника турніру, що доступні при проведенні турніру.

3.1. Сторінка інформації про турнір

Припустимо, що учасник успішно авторизувався як описано в попередньому розділі. Після авторизації учасник потрапляє на сторінку інформації про турнір. Якщо турнір ще не почався, сторінка інформації про турнір може мати вигляд, показаний на рис. 3.1.

На сторінці відображається ім'я учасника (на даному малюнку - це ejudge), назва турніру (Test contest). Крім того, у рядку стану (рядок зеленого кольору) відображається поточний час на комп'ютері користувача і стан турніру (НЕ РОЗПОЧАВСЯ). Додатково відображається статистика про учасників турніру.

Зверніть увагу, що всі дати і часи вказуються по годинах сервера в часовому поясі, в якому знаходиться сервер. Саме по них буде вестися звіт часу від початку і до кінця турніру, обчислення штрафних балів і т. п. Якщо час сервера і час за Вашому годиннику відрізняються, враховуйте це розходження надалі. Тим не менш, якщо сервер підключений до мережі Інтернет, його час, як правило, буде синхронізовано з світовим астрономічним часом з точністю до часток секунди.

До початку турніру учасник може виконувати наступні операції (меню операцій займає дві блакитних рядка вгорі сторінки).

- Установки - зміна установок: мови відображення сторінки і пароля користувача.
- Вийти з системи - завершити сеанс роботи з системою і вийти з турніру. Учасник може знову увійти в турнір в будь-який момент.
- Інфо - перегляд інформації про турнір. Саме з цієї сторінки починається робота користувача в турнірі.
- Повідомлення - перегляд повідомлень від суддів турніру. До початку турніру учасник може тільки переглядати повідомлення від суддів. Після початку турніру учасник може отримати можливість надсилати повідомлення суддям. Можливість як посилки повідомлень суддям, так і перегляду повідомлень може бути відключена адміністратором турніру. Тоді пункт меню «Повідомлення» не буде відображатися.



рис. 3.1: Вид сторінки, яка відображається після успішного входу в турнір

3.2. Початок турніру

Турнір стартує або в момент астрономічного часу, вказаний адміністратором заздалегідь, або безпосередньо по команді адміністратора турніру. Якщо старт турніру запланований на певний час, час старту вказується у рядку стану турніру як показано на рис. 3.2.

09:07:45 / НЕ НАЧАЛСЯ / Начало в: 09:30:00

рис. 3.2: Рядок стану турніру, що показує час його початку

Коли настане час початку турніру, показане в рядку стану турніру, або коли буде оголошено, що турнір почався, учасник турніру повинен натиснути кнопку «Refresh» веб-браузера. Після цього сторінка інформації про турнір буде перезавантажена і прийме вид, показаний на рис. 3.3.

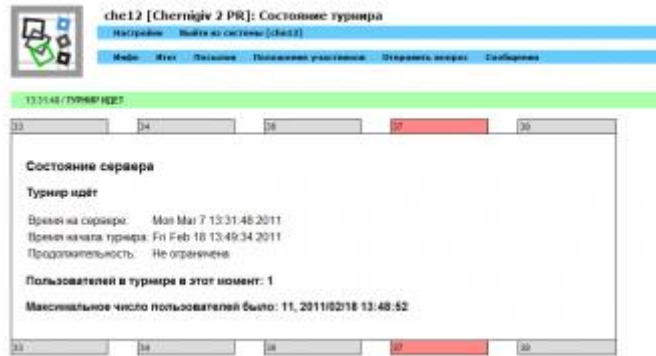


рис. 3.3: Сторінка інформації про турнір після його початку

На додаток до вже розглянутих пунктів меню «Налаштування», «Вихід з системи», «Інфо» з'явилися пункти меню «Результат», «Посилки», «Положення команд», «Надіслати питання», «Повідомлення». Всі пункти меню будуть детально розглянуті в наступних розділах. Зверніть увагу, що в залежності від налаштувань турніру частина пунктів меню може бути відсутнім або називатися по-іншому.

Рядок стану турніру тепер містить поточний час сервера, яке оновлюється кожну секунду, повідомлення «ТУРНИР ЙДЕ» і час, що залишився до кінця турніру (якщо турнір має обмежену тривалість), яке також оновлюється кожну секунду.

На робочому полі сторінки з'явилися закладки, відповідні назвам завдань («33», «34», «35», «36»). Змінилася інформація про турнір, яка тепер містить таку інформацію.

- Стан турніру («Турнір йде»). Крім того, тут відображається додаткова інформація про стан турніру, наприклад, інформація про припинення перевірки рішень, припинення друку і т. п.
- Астрономічний час на сервері.
- Астрономічний час початку турніру, тривалість турніру і очікуваний час закінчення турніру. Тривалість турніру і, відповідно, очікуваний час закінчення турніру можуть бути змінені адміністратором в ході турніру. Для турнірів необмеженої тривалості ця інформація не відображається, якщо тільки адміністратором не було встановлено час закінчення турніру.
- Час заморозки таблиці результатів. У правилах проведення турніру може бути передбачена «заморозка» таблиці поточних результатів, доступною учасникам турніру і глядачам. Це означає, що за певний час до кінця турніру (зазвичай за 1 годину) таблиця результатів перестане оновлюватися, тобто учасники і глядачі бачать становище учасників на момент початку заморозки.

У ході турніру може виникнути ситуація, коли рішення учасників будуть перетестеровано через зміни обмежень на рішення, зміни тестів і т. п. Хоча, звичайно ж, ця ситуація небажана, на турнірах вона зустрічається, на жаль, достатньо часто. У цьому випадку заморожена таблиця буде оновлена, щоб відображати стан учасників на момент початку заморозки після перетестування.

Для турнірів необмеженої тривалості, заморожування не передбачена ніколи.

- Час, що минув від початку турніру, і час, що залишився до кінця турніру.
- Статистика про відвідуваності турніру.

3.3. Дії під час турніру

У даному розділі розглядаються дії, доступні учаснику під час турніру.

На рис.2 3.5 показано, що кожного завдання, запропонованого на турнірі, відповідає закладка з ім'ям завдання. Колір закладки в ході турніру може змінюватися і залежить від того, чи вирішена ця задача даними учасником чи ні. Можливі кольори закладки наведені нижче у таблиці 3.1

Якщо хоча б одна із закладок відзначена жовтим кольором, тобто будь-яке рішення учасника знаходиться в процесі перевірки, сторінка в браузері буде автоматично перезавантажуватися кожні 5 секунд, поки всі завдання учасника не будуть перевірені.

білий - поточне завдання (незалежно від того, вирішена вона чи ні)

сірий - не було спроб вирішення завдання

червоний - завдання не вирішено або вирішено не повністю
 зелений - задача вирішена повністю
 жовтий - будь-яке з рішень завдання в даний момент перевіряється

Таблиця 3.1: Кольори закладок

3.3.1. "Підсумок" - підсумок по завданнях

При виборі пункту меню «Результат» відображається сторінка результатів перевірки кожного завдання. Приклад сторінки результатів перевірки зображений на рис. 3.4.

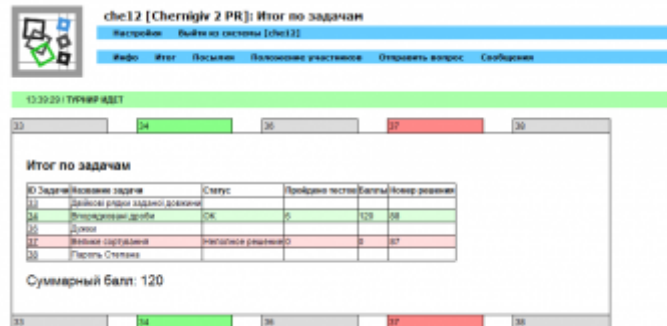


рис. 3.4: Сторінка сумарної інформації за завданнями

Зверніть увагу, що закладка завдання «37» має червоний колір, оскільки завдання А не вирішено, а закладка завдання «34» - зелена.

У таблиці для кожного завдання відображається посилка, яка буде давати внесок по цьому завданню в остаточний результат учасника. Правила вибору такої посилки залежать від типу турніру.

3.3.2. "Посилки" - Перегляд журналу посилки

При виборі пункту меню «Посилки» відображається журнал посилки даного учасника. Приклад сторінки журналу посилки зображений на рис. 3.5.

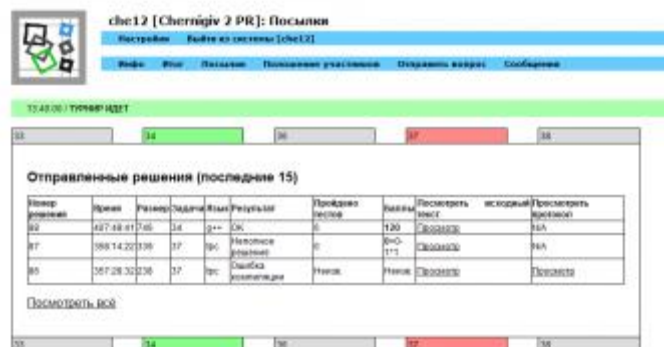


рис. 3.5: Сторінка журналу посилки

За замовчуванням показуються останні 15 посилки. Якщо було зроблено більш ніж 15 посилки, переглянути список всіх посилки можна натиснувши на посилання «Переглянути всі».

Рядки в таблиці посилки розташовуються в порядку зменшення часу посилки (тобто більш пізня посилка відображається вище).

Стовпець «Номер рішення» - це унікальний номер рішення в турнірі. Всі посилки всіх учасників турніру нумеруються послідовно, починаючи з 0. Номер посилки є її унікальним ідентифікатором в рамках одного турніру. Чим більше номер посилки, тим пізніше в турнірі вона була зроблена.

Стовпець «Час» - це час посилки від початку турніру. Для турнірів великої тривалості або необмежених за часом адміністратор турніру може встановити режим астрономічного часу, тоді в цьому стовпці буде виводитися астрономічне час.

Стовпці «Розмір» - це розмір посилки в байтах.

Стовпець «Завдання» - це коротке ім'я (ідентифікатор) завдання. Коротке ім'я завдання зазвичай представляє собою коротку рядок з латинських букв і цифр.

Стовпець «Мова» - це ідентифікатор мови програмування, який був використаний при вирішенні задачі. Для завдань, вирішення яких не вимагає визначення мови, даний стовпець буде порожнім.

Стовпець «Результат» - це результат перевірки посилки. Більш докладно про можливі результати

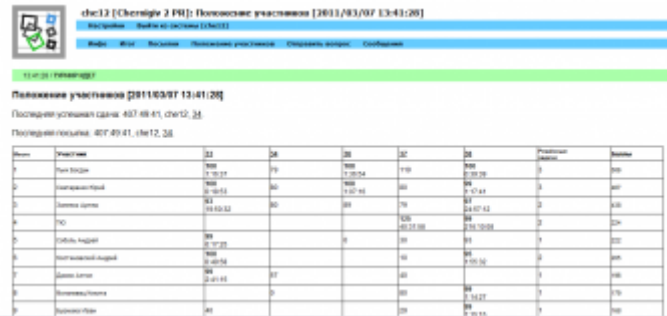
можна прочитати у відповідному розділі.

Стовпець «Помилка на тесті» містить номер тесту, на якому програма видала неправильний результат.

Натиснувши на посилання "Перегляд" у стовпці «Переглянути оригінальний текст» можна подивитися на вихідний текст даної посилки.

3.3.3. "Позиція учасників"

При виборі пункту меню «Становище учасників» відображається поточна таблиця результатів турніру. Якщо в ході турніру настав момент заморозки результатів, показується таблиця результатів на момент заморозки. На рис.2 3.6 показано вигляд сторінки поточних результатів.



The screenshot shows a web interface for a tournament. At the top, there is a header with the text "che12 [Сторінка 2 РН] | Поточне становище [2011/03/07 13:41:20]". Below the header, there is a navigation menu with options like "Вхід", "Вихід", "Настройка", "Положение участников", "Справка", "Справка". The main content area displays the title "Положение участников [2011/03/07 13:41:20]" and some statistics: "Последнее обновление: 4:07:48:41, che12_24" and "Последнее событие: 4:07:49:41, che12_24". Below this is a table with columns for "Имя", "Участник", "S1", "S2", "S3", "S4", "S5", "Рейтинг", and "Время". The table lists several participants with their respective scores and times.

Имя	Участник	S1	S2	S3	S4	S5	Рейтинг	Время
1	Том Биде	100	70	100	110	100	5	0:00
2	Степановский	100	80	100	80	80	3	0:07
3	Степановский	100	80	100	80	80	3	0:07
4	Том Биде	80	80	80	80	80	2	0:07
5	Том Биде	100	80	80	80	80	2	0:07
6	Том Биде	100	80	80	80	80	2	0:07
7	Том Биде	100	80	80	80	80	2	0:07
8	Том Биде	100	80	80	80	80	2	0:07
9	Том Биде	100	80	80	80	80	2	0:07
10	Том Биде	100	80	80	80	80	2	0:07

рис. 3.6: Сторінка поточних результатів

3.3.4. "Поставити запитання"

При виборі пункту меню «Надіслати питання» відображається сторінка посилки питання журі, показана на рис. 3.7. Адміністратор турніру може відключити можливість відправки питань журі, тоді цей пункт меню буде недоступний.

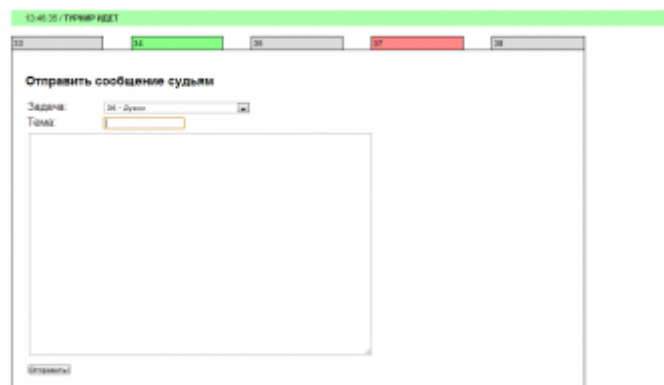
У випадяючому меню «Завдання» можна вказати завдання, за якою задається питання, а в полі «Тема» сформулювати тему питання. Потім у полі вводу набирається власне питання. Загальна довжина питання обмежена і дорівнює за замовчуванням 1024 байти, але адміністратор турніру може змінити це обмеження. Для відсилання питання необхідно натиснути на кнопку «Відправити».

3.3.5. "Повідомлення"

Пункт меню «Повідомлення» дозволяє переглядати повідомлення, відправлені журі, і відповіді, отримані від журі. Якщо у учасника є не переглянуті їм повідомлення від журі, рядок стану турніру має жовтий колір, і в ній з'являється напис n непрочитаних повідомлень (як показано на рис. 3.8. Учасник може натиснути на посилання «Повідомлення» і перейти на сторінку перегляду повідомлень, показану на рис. 3.8.

У таблиці виводяться останні 15 повідомлень, відправлених учасником журі і по-опромінених ним від журі. Щоб побачити всі повідомлення необхідно натиснути на посилання «Подивитися всі».

У таблиці повідомлень міститься наступна інформація. Номер повідомлення - це порядковий номер повідомлення в турнірі. Всі повідомлення всіх користувачів нумеруються від 0, таким чином номер повідомлення дозволяє однозначно ідентифікувати повідомлення в турнірі.



The screenshot shows a web form titled "Отправить сообщение судьям". It has a "Задача:" field with a dropdown menu showing "04 - Дубли" and a "Тема:" field with a text input box. Below these fields is a large text area for entering the question. At the bottom left of the form is a "Отправить" button.

рис. 3.7: Відправлення питання журі

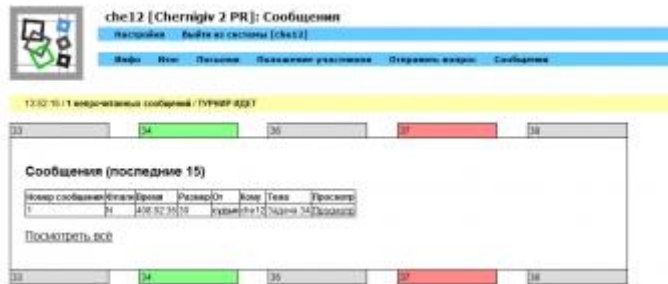


рис. 3.8: Перегляд повідомлень

Стовпець Прапорці може приймати такі значення:

U Питання ще не був прочитаний (для питань учасника журі)

R Питання було прочитано, але відповідь не ще не дано (для питань учасника журі)

A На питання було дано відповідь (для питань учасника журі) N Нове повідомлення учасникові від журі

Таблиця 3.2: Можливі значення стовпця «Прапорці»

Стовпець «Час» містить час відправки повідомлення. Час вимірюється від початку турніру, але для турнірів великий або необмеженої тривалості адміністратор турніру може включити режим відображення календарного часу.

«Розмір» - це розмір повідомлення в байтах. Як було сказано вище, розмір повідомлення від учасника журі обмежений за замовчуванням 1024 байтами. Це обмеження може змінюватися адміністратором турніру. Розмір повідомлення від журі учаснику не обмежений.

Стовпець «Вид» містить рядок судді, якщо повідомлення було надіслано журі учаснику, або реєстраційне ім'я учасника. Показуються тільки повідомлення або відправлені даними учасником, або адресовані даному учаснику, або адресовані всім учасникам. Стовпець «Кому» містить рядок судді, якщо повідомлення відправлено учасником журі, реєстраційне ім'я учасника, якщо повідомлення було надіслано журі цьому учаснику, або рядок все, якщо повідомлення було надіслано журі всім учасникам.

Стовпець «Тема» містить тему повідомлення. Натиснувши на посилання "Перегляд" можна переглянути повідомлення. Як тільки нове повідомлення, адресоване учаснику, буде переглянуто, рядок стану турніру знову прийме зелений колір.

3.3.6. "Здача завдань"

Натисканням на закладку завдання можна перейти на сторінку здачи рішення. Вид сторінки здачи рішення залежить від типу задачі. Для стандартних завдань, що вимагають здачи програм, сторінка здачи завдань виглядає як показано на рис. 3.9.

На сторінці здачи завдань може відображатися умову задачі або посилання на умову завдання. За допомогою випадаючого меню «Мова» вибирається мова програмування.

рис. 3.9: Сторінка здачи завдань

4. Оформлення розв'язків

При оформленні рішень завжди керуйтеся вимогам до вирішення, сформульованим в умові завдання, пам'ятці учасника або правилах проведення турніру. Для завдань, що не припускають написання та автоматичне тестування програм, такі вимоги, як правило, не викликають додаткових труднощів.

У цій главі описуються як загальні вимоги до рішень, так і особливості тестування рішень під управлінням системи ejudge на ОС Linux. Дана глава буде особливою корисною учасникам, програмуючим на операційних системах сімейства Windows.

4.1 Загальні вимоги

Рішення, що здається на перевірку, має представляти собою файл із вихідним текстом програми.

Файл з вихідним текстом повинен бути текстовим файлом у 8-бітній кодунанні (cp1251 - windows, cp866 - dos, koі8-r, utf8) і не повинен містити байта 0. Це значить, що не можуть бути здані файли в 16-бітній кодунанні (наприклад, windows unicpp).

Якщо здається бінарний файл або файл, що містить нульовий байт, файл не буде прийнятий на перевірку, і буде видане відповідне повідомлення про помилку.

Рішення має складатися з одного файлу з вихідним текстом. Проекти, що складаються більш ніж з одного вихідного файлу не підтримуються.

У залежності від умови задачі програма повинна зчитувати вхідні дані зі стандартного потоку вводу

(часто в цьому випадку пишуть «зчитувати з клавіатури») або з заданого файлу. У залежності від умови задачі результат роботи повинен виводитися або на стандартний потік виводу (часто званий «екраном»), або в заданий файл. Ніякі інші файли використовувати не можна. Програма не повинна створювати діалогові вікна (як графічні, так і текстові), довантажувати інші модулі та бібліотеки і т. п.

4.1.1. Робота з потоками вводу-виводу

У всіх підтримуваних мовами програмування передбачені функції виведення даних на стандартний потік виведення і введення даних зі стандартного потоку вводу.

В якості прикладу розглянемо фрагмент програми, який зчитує два цілих числа в змінні *a* і *b*, виробляє над ними деякі обчислення і виводить результат, отриманий у змінній *res*.

У програмах на мові Сі для читання даних слід використовувати функції `scanf`, `getchar` і т. д., а для виведення результату - функції `printf`, `putchar` і т. д.

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

У програмах на мові С++ можна використовувати як функції введення-виведення мови С, так і операції читання з `cin` і записи в `cout`.

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

У програмах на мові Паскаль для читання слід використовувати процедуру `read` без файлового параметра або з файлом `input`, а для запису - процедуру `write` без файлового параметра або з файлом `output`.

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

У програмах на мові Джава для читання слід використовувати методи роботи з потоком `System.in`, а результат виводити в потік `System.out`. Для читання даних можна використовувати клас `Scanner`, але рекомендується створювати `StreamTokenizer`. Специфіка мови Джава більш детально обговорюється нижче.

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

У програмах на мові Бейсік для читання слід використовувати оператор `INPUT`, а для виведення результату - оператор `PRINT`.

```
INPUT a  
INPUT b  
REM обчислення  
>>> print res
```

Зверніть увагу, що при читанні даних зі стандартного потоку введення в жодному разі не можна виводити запрошень до введення даних, наприклад, Введіть *x*: Цей текст буде виведений на стандартний потік виводу і «зіпсує» результат роботи програми. Тестування програми завершиться з помилкою «Неправильний формат виведення» або «Неправильний відповідь».

4.1.2. Робота з файлами

Якщо у формулюванні задачі потрібне введення даних з файлу і (або) висновок результату в файл, потрібно використовувати засоби роботи з текстовими файлами. Крім того, в деяких мовах програмування існує можливість прив'язати стандартні потоки введення і виведення до файлів. Припустимо, що за умовою задачі потрібно зчитувати дані з файлу `input.txt` і записувати результат у файл `output.txt`. Зверніть увагу, що імена файлів повинні записуватися в точності так само, як в умові, тобто має дотримуватися регістр літер і не повинні використовуватися будь-які абсолютні або відносні шляхи.

Таким чином, ім'я файлу `Input.txt` неправильно, тому що не збігається регістр букв, а ім'я файлу `c:\work\input.txt` неправильно, тому що містить елементи шляху до файлу, крім поточного каталогу.

У мові С для відкриття файлу можна використовувати функцію `fopen`, а для подальшої роботи функції `fscanf` і `fprintf`. Виклики функцій `fclose`, в принципі, не є обов'язковими, так як при нормальному

завершенні всі відкриті файли закриваються автоматично.

Припустимо, що зчитуються значення двох цілих змінних *a* і *b*, над ними про-водяться обчислення, і друкується значення змінної *res*.

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

Альтернативний спосіб полягає у використанні функції `freopen` для прив'язки стандартних потоків до файлів.

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

Обидва способи рівнозначні, але другий спосіб краще, оскільки в першому способі легше помилитися у тексті програми, переплутавши виклики функцій `fprintf` і `printf` або імена файлових змінних.

У мові C++ можна використовувати всі можливості мови C чи використовувати файлові потоки мови C++.

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

У мові Паскаль для відкриття файлів використовуються процедури `assign`, `reset` і `rewrite`.

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

За аналогією з мовою C можна прив'язати стандартні файли `input` і `output`.

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

У мові Бейсік файли відкриваються наступним чином.

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

4.1.3. Завершення програми

В кінці роботи програма повинна закрити всі відкриті файли і завершитися з кодом повернення 0. У всіх мовах, окрім Паскаля і Бейсіка всі відкриті файли при завершенні програми закриваються автоматично. Якщо у програмі на Паскалі не закрити вихідний файл, то можливо, що частина виведеної інформації виявиться втрачена, і, таким чином, результат роботи програми виявиться неправильним.

В усіх мовах програмування крім C і C++ нульовий код завершення програми виробляється автоматично при її нормальному завершенні. У мовах C і C++ в кінці функції `main` необхідно використовувати оператор `return 0`.

4.1.4. Повні приклади програм

Розглянемо завершені програми для вирішення задачі про знаходження суми двох цілих чисел.

Припустимо, що у файлі `input.txt` знаходяться два цілих числа, а їх суму необхідно вивести в файл

output.txt. Вихідні числа по модулю не перевершують 10000.

Програма на мові C.

```
#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;
}
```

Зверніть увагу, що функція main оголошена як повертає ціле значення. Програма, в якій функція main оголошена як не повертає значення (void), є неправильною з точки зору стандарту мови C. Компіляція такої програми компілятором GNU C завершиться помилкою. Зверніть увагу також на оператор return в кінці функції main, який необхідний, щоб програма завершувалася з кодом повернення = 0:

Програма на мові C++.

```
#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;
}
```

Як і в мові Cі, функція main повинна повертати ціле значення. Для цього в кінці функції знаходиться оператор return.

Програма на мові 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.
```

Програма на мові Бейсік.

```
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. Перевірка програми

Здане на перевірку рішення обробляється таким чином.

Вихідний текст програми компілюється вказаним при посилці компілятором. Якщо при компіляції виникають помилки компіляції, перевірка рішення на цьому завершується зі статусом «Помилка компіляції» ("Compilation error"). Час компіляції обмежений, і якщо компіляція програми зажадала більше часу, ніж встановлене обмеження, або компілятор зациквився, так само діагностується помилка компіляції. Така ситуація, зокрема, може виникнути при неправильному використанні шаблонів мови C++.

Як правило, у разі помилки компіляції учасник може переглянути повний висновок компілятора на стандартний потік виводу і стандартний потік помилок щоб встановити і усунути причину помилки.

Деякі особливості використовуваних в Linux компіляторів, які можуть призводити до помилок компіляції, описані далі. Як правило, спроби задачі, що призвели до помилки компіляції, не враховуються при обчисленні штрафних балів учасника. Тим не менше всі спроби задачі враховуються при перевірці квот учасника на кількість і загальний розмір посилки.

Якщо компіляція програми завершилася успішно, виконуваний файл передається на тестування. Якщо при компіляції програми компілятор видав які-небудь діагностичні попередження, то вони губляться, і їх текст учаснику не доступний.

Програма вважається такою, що пройшла будь-якою тест, якщо при запуску на цьому тесті програма вклалася в обмеження за часом і по пам'яті, завершилася нормально (з кодом завершення 0) і видала правильний результат. В інших випадках програма вважається що програма не пройшла тест. При запуску програми на кожному тесті виконуються наступні дії. Тестовий файл копіюється в робочий каталог програми з ім'ям, що відповідає умові завдання. Потім запускається програма, що тестується. Під час роботи програми, що тестується контролюється обмеження процесорного часу, обмеження реального часу і обмеження розміру використовуваної пам'яті. Якщо програма завершилася з кодом повернення 0 і вклалася в обмеження щодо часу і пам'яті, запускається перевіряюча програма, яка порівнює відповідь, виданий тестованою програмою, з правильним.

Контроль обмежень по пам'яті. Для 32-бітних і 64-бітних додатків linux (мови gcc, g + +, fpc, dcc та ін), а так само для програм на скриптових мовах, для виконання яких запускається відповідна програма-інтерпретатор (perl, python, scheme та ін), контролюється максимальний розмір всього адресного простору процесу і розмір стека. У адресний простір процесу входить код програми, код і дані всіх використовуваних програмою динамічних бібліотек, дані програми і стек. Якщо в процесі роботи ці обмеження перевищуються, працює модифіковане ядро Linux, і в турнірі включена підтримка помилки ML (Memory limit exceeded), програма буде знята з виконання з діагностикою ML. Якщо діагностика цієї помилки не підтримується, то програма швидше за все завершиться з помилкою RT (Run-time error). Для 16-бітних додатків DOS доступно приблизно 500 кілобайт пам'яті, а стек не може перевищувати 64 кілобайт. Ці обмеження обумовлені моделлю пам'яті DOS і спеціально не контролюються. Діагностика помилки ML для DOS-програм не підтримується.

Для програм мовою Java при запуску віртуальної машини встановлюється обмеження на розмір купи (області даних) і розмір стека. У подальшому контроль цих обмежень проводиться віртуальною машиною Java. У разі перевищення обмежень викидається виключення і програма знімається з виконання з діагностикою RT.

Обмеження на максимальний об'єм пам'яті і максимальний розмір стека визначаються в умовах задач або в правилах проведення змагань. За замовчуванням максимальний розмір стека дорівнює 8 мегабайтам.

Контроль обмежень за часом. Час роботи програми на кожному тесті обмежена. Обмеження на час роботи програми також визначаються або в умовах завдань, або в правилах проведення змагань. При виконанні програми контролюється обмеження як процесорного, так і реального часу.

Процесорний час - це час, витрачений процесором на виконання безпосередньо тестує програми. Цей час складається з часу роботи призначеного для користувача коду (власне реалізованого алгоритму, часу на форматування виводу і пр.) і часу роботи системних функцій, викликаних в процесі роботи програми (наприклад, операція відкриття файлу вимагає деякого часу на підтримку внутрішніх структур даних ядра операційної системи). Час, коли процес не виконувався, а чекав завершення операції читання даних з диска або був припинений для виконання більш пріоритетного процесу, не вносить внесок у істраченне процесорний час.

Реальний час обчислюється по годинах реального часу, тому враховує простої процесу. Як правило, реальний час незначно відрізняється від процесорного часу. Проте, якщо процес весь час очікує надходження даних або просто «засинає» на великий час, то процесорний час для нього зупиняється, і витрачений реальний час буде значно більше спожитого. Для запобігання ситуацій, коли та-де рішення «підвішує» процес тестування, контролюється реальний час, і якщо тестуєма програма не завершилася за відведений їй інтервал реального часу, вона знімається з виконання з діагностикою TL (Time-limit exceeded).

Можлива й інша ситуація. Якщо операційна система досить сильно завантажена іншими процесами, то програма буде часто перериватися виконання інших процесів, і тому для неї реальний час може бути більше віртуального. Тому обмеження на реальний час встановлюється в 2-3 рази більші обмеження на процесорний час.

Якщо програма в процесі роботи створює кілька ниток, то процесорний час всієї програми підсумовується за всіма ниткам. Якщо комп'ютер підтримує одночасне виконання кількох ниток (наприклад, на багатоядерних процесорах або багатопроцесорної системі), то реальний час буде у відповідну кількість разів менше процесорного часу.

Вердикт тестування. Результатом тестування програми є один з вердиктів, перерахованих у таблиці 4.1. У випадку, якщо виникла помилка CF (Check failed), зверніться до адміністратора турніру. Ця помилка означає, що при перевірці Вашої програми стався збій в перевірці системі, через що програма не була перевірена.

OK
коментар тест пройдено

CE	Compilation error	Помилка компіляції
RT	Runtime error	Помилка виконання
TL	Timelimit exceeded	Перевищено максимальний час роботи
PE	Presentation error	Неправильний формат виводу
WA	Wrong answer	Неправильна відповідь
CF	Check failed	Помилка перевіряє системи
ML	Memory limit exceeded	Перевищено ліміт по пам'яті
SE	Security violation	Порушення правил безпеки

Таблиця 4.1: Вердикти тестування

Тестована програма запускається на виконання, при цьому контролюються час її роботи, розмір використовуваної нею пам'яті, виконуваних їй операцій. Розглянемо можливі ситуації, які можуть виникнути при тестуванні, докладніше.

- Програма спробувала виконати заборонену операцію, наприклад, видалити файл або створити мережеве з'єднання. У цьому випадку програма може бути знята з виконання з вердиктом «Порушення правил безпеки». Зверніть увагу, що за порушення правил проведення турніру Ви може бути дискваліфіковані.
- Програма перевищила обмеження на розмір пам'яті або стека. У цьому випадку програма знімається з виконання і діагностується помилка «Перевищено ліміт по пам'яті». Можливі причини цього такі:
 - Нескінченна рекурсія в програмі і, як наслідок, переповнення стека.
 - Оголошено занадто більшою локальний або глобальний масив.
 - У процесі роботи програма запитує надто багато динамічної пам'яті.

Для програм, що виконуються в середовищі DOS, перевищення обмеження по пам'яті не діагностується. Наприклад, програма, яка входить в нескінченну рекурсію, швидше за все завершиться з помилкою «Помилка виконання».

- Програма перевищила обмеження на процесорний час. У цьому випадку програма знімається з виконання і діагностується помилка «Перевищено максимальну час роботи». Можливі причини цього такі:
 - Через помилку в програмі вона зациклилася.
 - У програмі реалізований неефективний алгоритм, через який вона не вкладається в обмеження за часом на великих вхідних даних.
- Програма перевищила обмеження на реальний час. У цьому випадку програма знімається з виконання і діагностується помилка «Перевищено максимальну час роботи». Можливі причини цього такі:

- Замість читання з файлу програма очікує введення з клавіатури.

- У процесі виконання програми виникла помилка виконання, така як вихід за межі масиву, звернення по неіснуючому адресою або ділення на нуль. У цьому випадку програма знімається з виконання і діагностується «Помилка вико-нання». Зверніть увагу, що в деяких випадках помилка виконання може бути наслідком перевищення програмою максимального розміру пам'яті.
- Програма завершується з ненульовим кодом повернення в результаті виконання процедури `halt (n)` у Паскалі або `exit (n)` в С і С++. У цьому випадку діагностується «Помилка виконання».

Якщо програма була знята з виконання або завершилася з будь-якої з перерахованих вище причин, висновок програми не перевіряється. Якщо програма відпрацювала коректно, то вихідний файл, згенерований програмою, перевіряється на правильність.

Якщо вихідний файл, необхідний за умовою задачі, відсутній або порожній, діагностують-ся «Неправильний формат виведення». Якщо вміст вихідного файлу не відповідає вимогам до формату результату, також діагностується «Неправильний формат виведення».

Якщо вихідний файл оформлений відповідно до вимог до задачі, але відповідь не збігається з правильним, діагностується «Неправильна відповідь».