

## Лабораторная работа 2. Закрепление навыков работы с Svn и Trac.

### Оглавление

1. [Условие задачи](#)
2. [Задание 1](#)
3. [Задание 2](#)
4. [Задание 3](#)
5. [Задание 4](#)
6. [Задание 5](#)
7. [Задание 6](#)
8. [Задание 7](#)
9. [Задание 8](#)
10. [Вопросы и задачи для самопроверки](#)

Целью лабораторной работы является закрепление навыков работы с системой управления версиями SVN и системой ведения проекта Trac.

В ходе выполнения лабораторной работы студенты должны

1. приобрести навыки работы с историей изменений в репозитории, научиться анализировать эти изменения;
2. закрепить навыки, полученные в ходе первой лабораторной работы (команды "типичного цикла работы" с рабочей копией, работа с системой отслеживания ошибок и документирования в Trac).

### Условие задачи

Исходными данными для выполнения лабораторной работы является текст программы, который находится в Вашем репозитории по адресу:

`http://dev.iu7.bmstu.ru/svn/ИМЯ_ПРОЕКТА/bisect`

Эта программа претерпевала различные изменения, в результате которых в ревизии В в программу была внесена ошибка. Ошибку обнаружили только в ревизии С ( $C > B$ , ошибка присутствует во всех ревизиях, начиная с В). Кроме того, до ревизии А ( $A < B < C$ ) программа находилась в активной стадии разработки и могла либо работать неправильно, либо не компилироваться.

[X,A)	[A,B)	[B,C]
Программа не компилируется или не работает	Программа компилируется и работает	Программа компилируется и не работает

Замечание.

Ошибка не связана с добавлением ввода нового параметра с клавиатуры, проверкой значений параметров и т.п.

### Задание 1

Создайте отдельную страницу в Wiki, ссылку на которую разместите на первой странице Wiki (первая страница wiki называется WikiStart).

На этой странице Вы будете записывать ответы на задания.

### Задание 2

Найдите ревизию X, в которой в Вашем репозитории появилась папка bisect.

В ответе на созданной странице Wiki укажите

- номер задания;

- команду, с помощью которой Вы решили задачу (команда приводится со всеми ключами за исключением вашего логина и пароля);
- часть или весь вывод этой команды, подтверждающий правильность вашего ответа.

Для того чтобы скопировать текст из окна cmd, необходимо

- выбрать пункт "Изменить -> Пометить" в системном меню (пиктограмма в левом верхнем углу) этого окна;
- пометить нужную область окна;
- нажать Enter;
- вставить текст в нужное место.

### Задание 3

Найдите ревизию C, в которой внесены последние изменения в папку bisect.

Ответ нужно оформить точно так же, как в "Задании 2".

### Задание 4

Найдите ревизию D, которая принадлежит диапазону [A,B), т.е. такую ревизию, когда программа компилируется и работает.

*Алгоритм поиска* (вход: ревизия X, выход: ревизия D)

1.  $Y = X$
2. Извлечь ревизию Y
3. Проверить выполняется ли тест
4. Если тест выполняется, ревизия D найдена, в противном случае  $Y = Y + 1$  и переход на шаг 2

Для проверки работоспособности программы необходим *тест*. Программа вычисляет определенный интеграл методом трапеций. Метод трапеций дает точный результат, когда подынтегральная функция является прямой. Такая подынтегральная функция обязательно есть в программе.

Для компиляции программы не обязательно запускать IDE Lazarus. Программу можно скомпилировать из командной строки (т.е. из окна cmd) с помощью команды

```
C:\Lazarus\Lazbuild integral.lpi
```

При этом Вы должны находиться в папке, которая содержит файл integral.lpi.

Ответ на данное задание должен включать в себя:

- уравнение прямой, которое используется в программе;
- выбранный Вами интервал интегрирования;
- значение интеграла на этом интервале;
- команду, с помощью которой Вы извлекаете нужную ревизию (оформление см. в "Задании 2").

Кроме того, процесс поиска значения ревизии D необходимо проиллюстрировать следующей таблицей:

Y	Результат теста
7	Не пройден
...	...

*Замечание 1.*

Если Вы пользуетесь IDE Lazarus для компиляции программы, перед извлечением новой версии программы необходимо выходить из IDE и удалять все результаты компиляции.

*Замечание 2.*

Правильно работающая программа может не запрашивать никаких данных у пользователя

(например, значения этих данных могут быть заданы в самой программе). Главное, чтобы значение интеграла было верным.

*Замечание 3.*

$D = A$  (для описанного алгоритма)

## Задание 5

Найдите ревизию В.

Для поиска значения ревизии В воспользуемся *алгоритмом деления отрезка пополам* (вход: D, C; выход: В)

1.  $Mid = (Left + Right) \div 2$
2. Получить ревизию Mid из репозитория
3. Проверить выполняется ли тест в ревизии Mid.
4. Если тест выполняется, то ошибка между ревизиями Mid и Right (т.е. для продолжения поиска необходимо положить  $Left = Mid$ ), если не выполняется – между ревизиями Left и Mid (т.е. для продолжения поиска необходимо положить  $Right = Mid$ )

Критерий окончания поиска: для ревизии Mid тест не выполняется и  $Mid - Left = 1$ .

Начальное значение Left равно D, а Right равно C.

"Проиллюстрируйте" поиск ошибки следующей таблицей:

Left	Right	Mid	Результат теста
1	9	5	Не пройден
...	...	...	...

## Задание 6

Создайте отчет об ошибке (ticket).

Отчет об ошибке должен включать:

- номер ревизии, в котором найдена ошибка;
- описание шагов воспроизведения ошибки;
- описание данных, на которых ошибка воспроизводится.

В ответе на это задание приведите ссылку на созданный ticket.

## Задание 7

Сравните ревизии В-1 и В файла integral.lpr.

В ответе приведите команду, с помощью которой Вы это сделали, и ее вывод.

Какими комментариями сопровождалась фиксации изменений в ревизиях В-1 и В? Приведите в ответе текст этих комментариев. Помогли ли они Вам?

Определите, какие изменения привели к возникновению ошибки. Постарайтесь понять логику изменений. Возможно, для этого придется проанализировать несколько предыдущих ревизий.

Продублируйте причину ошибки в ticket (в отдельном комментарии!).

*Замечание.*

Не забудьте взять ticket в работу.

## Задание 8

Исправьте ошибку в **последней** ревизии, зафиксируйте Ваши изменения.

Чтобы получить последнюю ревизию

- откатите все сделанные Вами локальные изменения;

- выполните команду `svn update`.

В ответе на задание укажите команду, с помощью которой Вы откатали изменения.

*Замечание.*

Не забудьте закрыть отчет об ошибке.

### **Вопросы и задачи для самопроверки**

1. Необходимо найти в репозитории файл, изменения в который вносились несколько раз.  
Какую команду (или команды) вы будете использовать?
2. С помощью какой команды можно сравнить разные версии одного и того же файла?
3. Перечислите несколько способов получения из репозитория определенной версии файла.
4. Какая команда показывает состояние рабочей копии?
5. Каким образом можно отменить локальные изменения?