

Раздел «Язык Си» . SVN :

Subversion. Ликбез

- `svn checkout http://repository.url/svn/name` – извлекаем файлы проекта из репозитория, сокращение: `svn co`;
- `svn update` – получаем обновления из репозитория, сокращение: `svn up`;
- `svn update -r rev_num ./file_name` – извлекаем ревизию файла с номером `rev_num`;
- `svn add ./file_name` – добавляем файл в репозиторий (не важно текстовый или бинарный);
- `svn rename ./old_file_name ./new_file_name` – переименовываем файл в репозитории;
- `svn remove ./file_name` – удаляем файл/директорию из репозитория;
- `svn status` – просматриваем локально измененные файлы, сокращение: `svn st`;
- `svn status -u` – просматриваем локально измененные и изменившиеся в репозитории файлы, сокращение: `svn st -u`;
- `svn diff ./file_name` – показывает локальные изменения в файле построчно;
- `svn diff -r rev_num1:rev_num2 ./file_name` – показывает различия между ревизией `rev_num1` и `rev_num2` файла;
- `svn revert ./file_name` – откатывает локальные изменения файла (выгружает из репозитория последнюю закоммиченную ревизию);
- `svn revert -R ./` – откатывает все локальные изменения файлов;
- `svn log ./file_name` – список ревизий с комментариями;
- `svn blame ./file_name` – показывает авторов изменений файла построчно, синоним: `svn annotate`;
- `svn propset svn:ignore ./file_name .` – добавляем файл в список игнорируемых файлов;
- `svn propset svn:keywords "Id Author Date" ./file_name` – установка атрибутов файла;
- `svn cleanup` – снимает блокировки с файлов;
- `svn unlock http://repository.url/svn/file_name` – снять блокировку файла (URL можно узнать с помощью команды `svn info ./file_name | grep URL`, его и нужно передавать в `svn unlock`);
- `svnadmin setlog --bypass-hooks /path/to/repository -r rev_num ./commit_text_file` – заменяет текстовое описание коммита, где `rev_num` – номер ревизии, `commit_text_file` – путь к файлу, содержащему новый комментарий к коммиту;
- `svn help command_name` – выводит помощь по команде `command_name`, например, «`svn help update`»;
- `svn merge -r rev_to_rollback:rev_good ./file_name` – откатываем ревизию номер `rev_to_rollback` до ревизии `rev_good`, причем все изменения старше `rev_to_rollback` сохраняются (Например, у файла есть ревизии 11, 12 и 13. Хотим откатить 12-ую ревизию, но так, что бы изменения 13-ой остались в силе. Делаем тогда так: `svn merge -r 12:11 ./file_name`);
- `svn copy http://repository.url/svn/name/trunk/ http://repository.url/svn/name/branches/new_branch_name/` – создаем ветку с названием `new_branch_name` из главной линии разработки;
- `svn merge --dry-run -r rev_num1:rev_num2 http://repository.url/svn/name/trunk/` – проверяем, что будет изменено при объединении веток, где `rev_num1` – номер ревизии, когда ваша ветка была «открыта», или это м.б. номер предыдущего объединения (слияния), `rev_num2` – версия главной линии разработки, с которой производим объединение. Необходимо отметить, что все изменения будут применены для директории, в которой выполнялась эта команда;
- `svn merge -r rev_num1:rev_num2 http://repository.url/svn/name/trunk/` – синхронизирует вашу ветку с главной линией разработки с учетом ревизий: `rev_num1` – номер ревизии, когда ваша ветка была «открыта», или это м.б. номер предыдущего объединения (слияния), `rev_num2` – версия главной линии

Поиск**Поиск****Раздел «Язык Си»****Главная**
Зачем учить C?
Определения**Инструменты:****Поиск**
Изменения
Index
Статистика**Разделы****Информация**
Алгоритмы
Язык Си
Язык Ruby
Язык
Ассемблера
EI Judge
Парадигмы
Образование
Сети
Objective C**Login>>**

разработки, с которой производим объединение. Необходимо отметить, что все изменения будут применены для директории, в которой выполнялась эта команда;

- `svn ls http://repository.url/svn/name/branches/` – показать список доступных веток

[Отсюда](#)

todo

- Введение, термины (репозиторий, ревизия, ветка, проект)
- разбить по темам, отформатировать. `svn help` – первым и в рамочку

(с) Материалы раздела "Язык Си" публикуются под лицензией GNU Free Documentation License.