Мы разрабатываем систему VCS на языке Python – это система контроля версий, которая позволяет отслеживать изменения в проекте: изменения в коде программы, а так же появление новых файлов или удаление старых.

Наша система контроля версий обладает следующим функционалом:

1. Система авторизации
2. Система прав
3. Пользовательские функции для работы с проектом (commit, push, pull, load и т.д.)
4. Ветвление

Рассмотрим более детально каждый из пунктов.

**Система авторизации**

При запуске нашей программы, первой запускается функция авторизации. Она предлагает войти в систему, используя уже существующий аккаунт или создать новый. Если пользователя с введённым логином не существует, то программа выдаст ошибку авторизации и предложит создать новый аккаунт.

Для хранения информации об аккаунтах пользователей используется следующая структура:

{ username\_1: password; username\_2: password; }

Которая находится по следующему пути /global/bin/users.txt.

**Ветвление основа**

Ветвление в нашей системе контроля версий – это своеобразный механизм, позволяющий пользователю быстро переключаться между задачами и решать их (в пределах проекта). По умолчанию, проект, и все его изменения, хранятся в ветке master ( /project\_name/master/ ). Но, при возникновении необходимости изменить что-либо в проекте, при этом не меняя уже имеющийся код, пользователь может создать ещё одну ветку и сохранять изменения проекта уже в этой ветке. После внесения изменений (одного или нескольких коммитов и последующих пушей. Про них рассказано в пункте Пользовательские функции) пользователь может слить все сделанные изменения из своей ветки в ветку master.

**Система прав**

Наша система контроля версий использует систему прав для проверки доступа пользователя к определённому функционалу. Для работы используется 2 структуры:

users\_rights\_for\_project – используется для хранения списка пользователей, которые обладают тем или иным уровнем доступа к функционалу VCS в данном проекте.

{ project\_name: { master: [ admin, ivan, dima, denis ]; branch1: [admin, ivan, dima, petya] } }

---------------------------------------------------------------------------------------------------------------------------

users\_requests – используется для хранения приглашений для пользователя на вступление в тот или иной проект.

{username: { master: [ [project\_owner1, prj\_1] ]; branch1: [project3\_owner, prj\_3] }

**Принцип работы системы прав**

1. Пользователь заходит в свой проект и вызывает функцию добавления новых пользователей add\_users\_to\_project, затем он может добавить нового пользователя или в весь проект (во все ветки), или в какую-то ветку проекта. Выбрав один из вариантов, пользователю выводится список всех пользователей, из которого он выбирает тех, кого хочет добавить. Выбрав и подтвердив свой выбор, в структуру users\_requests каждому выбраному пользователю отправляется приглашение вида { branch: [ project\_owner, prj ] }.
2. Когда в систему заходит пользователь, то сразу же вызывается функция, проверяющая, есть ли у этого пользователя какие-нибудь приглашения, и если есть, то он должен выбрать, какие из запросов принять, а какие нет. Независимо от того, какие запросы выберет пользователь, все запросы на его имя будут удалены. Когда пользователь выберет, в какие проекты он хочет вступить, и подтвердит свой выбор, откроется файл со структурой users\_rights\_for\_projects, в которую будет, по выбранным в приглашении проектам и веткам, добавлено его имя.

Аналогично функции add\_users\_to\_project, работает функция del\_users\_from\_project, которая предлагает удалить пользователей из всего проекта или из какой-то ветки, а затем, выбрав и подтвердив свой выбор, из структуры users\_rights\_for\_projects будут удалены выбранные пользователи из соответствующих веток.

Уровень доступа определяется двумя функциями: have\_user\_high\_lvl\_of\_rights и have\_user\_some\_lvl\_of\_rights. Высшим уровнем пользователя обладает только админ и создатель проекта, то есть первая функция возвращает True, если имя пользователя находится в нулевом или первом элементе. Вторая ф-ция возвращает True для выбранной ветки, если в выбранной ветке есть его имя.

При создании ветки, 2-им элементом будет обозначено имя пользователя, создавшего эту ветку.

**Функции для работы с проектом**

После авторизации и проверки приглашений, пользователю предлагают следующий набор функций:

* choose - выбрать свой проект
* make – создать новый проект
* load – загрузить проект из локальной директории
* exit – выйти

choose - выводит на экран информацию обо всех проектах, доступных пользователю и принимает на вход имя проекта, который хочет открыть пользователь.

make – предлагает создать новый проект и, в случае, если проект будет создан, автоматически переходит в этот проект.

load – эта функция позволять загрузить проект (при условии, что пользователь обладает необходимым уровнем доступа), которой он, например, удалил.

exit – предлагает пользователю выйти из аккаунта, а затем выйти из VCS совсем.

**Функции, доступные после выбора проекта**

После того, как пользователь выберет проект, будет автоматически выбрана одна из доступных веток.

* change\_branch – выводит список веток, доступных пользователю и предлагает выбрать одну из них.
* make\_branch – функция для создания новой ветки.
* set\_project – функция для смены проекта.
* commit – функция для фиксирования изменений в проекте в папке local в стеке (структура для фиксирования изменений).
* push – функция для отправки изменений в папку global в текущую ветку. Так же в стек в папку global в выбранной ветке будут занесены соответствующие изменения.
* show projects – выводит список всех проектов, доступных пользователю.
* update – функция, сравнивающая изменения в стеках в папках local и global в выбранной ветке и, если есть какие-то изменения, то обновляет проект пользователя в папке local до последней версии.
* log –
* del\_last\_commit – функция, удаляющая последний коммит в папке local в выбранной ветке.
* merge –

Практически по всех функциях в начале выполнения проверяется, обладает ли выбранный пользователь необходимым уровнем доступа и если нет, то выполнение функции прерывается, а пользователю выводится соответствующее сообщение.

**Ветвление**