

UBPARSE

1 Общие действия

1. Считывание данных из файла „Шаблон парсинга“ (каталог parse)
2. Находим файл конфигураций
3. Читаем построчно файл
4. Определяем тип строки
5. Определяем есть ли конец файла, если да то завершаем, иначе читаем файл дальше

2 Если задан ключ -r

1. Сравниваем уровень вложение строки с заданным
2. При соответствии, фильтр сообщает об успешности

3 Если задан ключ -n

1. Проверяем флаг вложенности в блок
2. При соответствии, фильтр сообщает об успешности

4 Если задан ключ -t

1. Проверяем тип записи
2. При соответствии, фильтр сообщает об успешности

5 Если задан ключ -k

1. Проверяем идентификатор настройки
2. При соответствии, фильтр сообщает об успешности

6 Если задан ключ -l

1. Проверяем тип записи на соответствие типа блок
2. Получаем имя блока
3. Вывод имени

7 Если задан ключ -f

1. Считывание данных из файла „Формат вывода“ (каталог Out)
2. Подстанока данных в указываемый формат
3. Вывод данных

8 Функций и их взаимодействия

8.1 Список используемых функций

- api.sh
 - init_pattern:
 - * Инициализация шаблона парсинга (каталог parse)
 - * Запись шаблона в память
 - * Принимает путь до файла шаблона
 - get_name:
 - * Извлечение имени блока по шаблону парсинга
 - * Возвращает имя блока
 - * Принимает регулярное выражение по которому определяется блок и строку содержащие начало блока
 - block_name:
 - * Извлечение имени из типа
 - * Возвращает имя блока
 - * Принимает тип записи
 - check_str:
 - * Определяет тип строки конфигураций
 - * Возвращает тип строки
 - * Принимает строку из файла конфигураций
 - recursive_read:
 - * Рекурсивное чтение конф.файлов
 - * Построчно возвращает обработчику тип и содержимое строки
 - * Принимает путь до файла конфигураций и набор обработчиков
 - handlers:
 - * Последовательный вызов обработчиков
 - * Обеспечивает универсальную построчную обработку
 - * Принимает набор функций который необходимо исполнить
- filter.sh

- filter_nest:
 - * Фильтр вложений
 - * Возвращает статус фильтраций
 - * Принимает уровни вложений или их диапазоны
- filter_type:
 - * Фильтр по типу записи
 - * Возвращает статус фильтраций
 - * Принимает набор типов
- filter_name:
 - * Фильтр по имени блока
 - * Возвращает статус фильтраций
 - * Принимает имена блоков
- filter_name_key:
 - * Фильтр по ключам настроек
 - * Возвращает статус фильтраций
 - * Принимает имена ключей
- out.sh
 - init_format:
 - * Инициализация формата вывода
 - * Запись формата в память
 - * Принимает путь до файла формата
 - list_block:
 - * Получение имен всех блоков
 - * Вывод всех имен блоков которые содержатся в конфигураций
 - * Не принимает параметров
 - render:
 - * Форматирование строк конфигураций
 - * Отформатированный вывод данных
 - * Принимает формат вывода
 - direct_render:
 - * Прямой вывод данных
 - * Не принимает параметров
 - number_str:
 - * Вывод номера строк и путь до файла источника
 - * Не принимает параметров
 - format_write:

- * Вывод формата записи
 - * Отформатированный вывод данных
 - * Принимает тип записи
- log_out:
- * Диагностический вывод
 - * Не принимает параметров

8.2 Взаимодействие функций

С помощью `init_pattern` программа получает информацию о конф.файлах их способу парсинга, затем с помощью функций `recursive_read` читает конфигурацию построчно которая в свою очередь с помощью функций `check_str` (внутри которой используется для формирования типов функция `get_name`) определяет тип строк и отправляет строки и их типы в функцию `handel`. `Handel` поочередно передает данные в функций который содержатся в наборе его параметров. Остальные функций вызываются с помощью ключей.