Операционные системы и среды
Л.р.3. Обработка текстовой информации. Регулярные выражения.

Цель:

Изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит.

Теоретическая и методическая часть

...

Практическая часть

Общая постановка задачи:

Написать скрипт (скрипты) для *sed*, *awk* и т.д., либо скрипт *shell*, обращающийся к необходимым программам, для обработки входных данных (файлов) в соответствии с вариантом задания.

Необходимо предусмотреть поведение скрипта (скриптов) при ошибочных или "неочищенных" входных данных.

Варианты заданий:

1) «Автокорректор» (заглавные буквы)
2) «Сканер данных»
3) «Табличный калькулятор»
4) «Статистика»
5) «Планирование»
6) Импорт данных (БД)
7) «Пред-сортировка» файлов

1) «Автокорректор» (заглавные буквы)

Замена строчных букв на заглавные в начале предложений, т.е. в начале документа и после точки, не находящейся внутри, например, числа, а также после знаков «!», «?».

Предложение может начинаться с новой строки (т.е. предыдущая точка может находиться в одной строке, а заменяемая строчная буква – в следующей).

2) «Сканер данных» (хотя точнее можно было бы назвать «парсером»)

Извлечение из текста подстрок с характерным форматом (например, адреса, web-ссылки, наименования книг с авторами и т.п.)

В сложных случаях допустимо вводить искусственное определение формата, например для литературного источника: «Фамилия_автора запятая Инициалы точка Название в кавычках тире Год_издания Необязательная_интернет-ссылка».

3) «Табличный калькулятор»

Входные данные – таблицы (матрицы) в файле (файлах) в формате CSV («comma-separated») и признак выполняемой над ними операции.

Нераспознанные (нечисловые) элементы заменяются нейтральными/незначащими значениями (например, нулями), аналогично дополняются недостающие.

Примечание: Признак операции и дополнительные сведения о данных, например размерность, можно включить во входные файлы – это может упростить логику обработки (при использовании awk)

4) «Статистика»

Конкретная постановка задачи может быть различной, например:

– Входные данные – файл (файлы) со списками товаров в формате: наименование, количество, цена. Одно и то же наименование может встречаться многократно.

– Выходные – список товаров без повторов наименований с вычисленными для них суммарным количеством, средней ценой, общей стоимостью.

5) «Планирование»

– Входные данные – таблица, содержащая среди прочих колонки начала и окончания интервала в виде временнЫх меток (дата и время).

– Выходные – таблица, строки которой отсортированы по началу, окончанию, длительности интервала.

Даты и время в различных строках таблицы могут быть записаны в разном формате (но с однозначным критерием распознавания).

6) Импорт данных (БД)

– Входные данные – файл в формате CSV («comma separated»).

– Выходные – SQL-запросы **`INSERT`** для переноса данных в БД (рекомендуется ограничиваться «стандартным» SQL, но при желании можно следовать специфическим диалектам).

Сведения о структуре таблицы БД могут содержаться во входных данных, например: первая строка входного файла – таблицы, вторая строка – имена соответствующих полей в этой таблице.

7) «Пред-сортировка» файлов

Есть программа (например медиаплейер, что несущественно), для которой нужно обеспечить обработку файлов данных в определенном порядке, однако сама она считывает их только по алфавиту.
Решение – дополнить имена файлов префиксами, представляющими собой порядковый номер (например, 4-значный), т.е.:

`файл1``→0000.`***`файл1`***, ***`файл2`***`→0001.`***`файл2`***, ***`файл3`***`→0002.`***`файл3`*** и т.д.

Скрипт должен в зависимости от параметров переименовывать файлы (для упрощения в текущем директории) в соответствии с несколькими

правилами сортировки: в прямом или обратном порядке по дате/времени, по размеру, по алфавиту (для унификации) и т.п., в случайном порядке (перемешивание), при этом корректно отбрасывая предыдущие префиксы. Нужна также возможность убрать ранее добавленные префиксы у всех файлов.

и т.п.