|  |  |
| --- | --- |
| Федеральное агентство железнодорожного транспорта  Омский государственный университет путей сообщения  Кафедра «Автоматика и системы управления»  РАБОТА С ТЕКСТОМ И КОДИРОВКАМИ  Лабораторная работа № 1  по дисциплине «Информационные технологии» | |
|  | Студент гр. 22м                              Н.И. Яковлев                              В.А. Зиферман                              А.И. Знаенок  « 10 » сентября 2022 г.  Руководитель –  старший преподаватель кафедры «АиСУ»                            Т. В. Васеева  «    »                  2022 г. |
| Омск 2022 | |

# Цель работы

Изучение работы редактора «Visual Studio Code» и редактора двоичных файлов «Frhed» в ходе выполнения работы; изучение кодировок.

# Выполнение работы

## Текстовый файл

После создания текстового файла, мы открыли его в редакторе «Visual Studio Code», по умолчанию кодировка UTF-8 (см. рисунок 1).

|  |
| --- |
|  |
| Рисунок  – Текстовый файл в «Visual Studio Code» |

## Текстовый файл в «Frhed»

Далее мы открыли текстовый файл в «Frhed» и увидели с левой стороны шестнадцатеричное представление содержимого текстового файла, а также дополнительные информацию в статус-панели программы. Также по умолчанию кодировка стояла ANSI, её по нажатию на надпись удалось переключить на OEM (см. рисунок 2).

|  |
| --- |
|  |
| Рисунок  – Текстовый файл в «Frhed» |

## Даты и их представления в системах счисления

Дописывая в правой стороне даты рождения, можно наблюдать, как сразу же появляются шестнадцатеричные представления введенных символов (см. рисунок 3).

|  |
| --- |
|  |
| Рисунок  – Дописанные даты рождения |

Если нажать на символ в правой части экрана, то программа помимо выделения шестнадцатеричного эквивалента, выводит дополнительную информацию. Для удобства мы выделили тремся цветами различные представления символов при их расшифровке. Красным – шестнадцатеричное представление, синем – двоичное, фиолетовым – десятичное (см. рисунки 4-9).

|  |
| --- |
|  |
| Рисунок  – Представление символа «0» в различных системах счисления |
|  |
| Рисунок  – Представление символа «7» в различных системах счисления |
|  |
| Рисунок  – Представление символа «8» в различных системах счисления |
|  |
| Рисунок  – Представление символа «4» в различных системах счисления |
|  |
| Рисунок  – Представление символа «2» в различных системах счисления |
|  |
| Рисунок  – Представление символа «1» в различных системах счисления |

## Кодировки

Следуя заданию, по нажатию в панели статуса программы «Frhed» по надписи «OEM», кодировка изменилась на «ANSI» (см. рисунок 10).

|  |
| --- |
|  |
| Рисунок  – Текстовый файл в кодировке ANSI |

Далее, для каждой из предложенных кодировок, мы изменяли кодировку текстового файла, открывали новый экземпляр программы «Frhed» и далее сравнивали (см. рисунки 11 - 13).

|  |
| --- |
|  |
| Рисунок  – Текстовый файл в кодировке «Windows 1251» |
|  |
| Рисунок  – Текстовый файл в кодировке «KOI8-RU» |
|  |
| Рисунок  – Текстовый файл в кодировке «CP 866» |

В ходе более подробного изучения вопроса, стало ясно, что изменений для заданного заданием текста попросту нету. Это связано с тем, что каждая из предложенных кодировок отталкивалась от своих истоков, т.е. от кодировки ANCII. Для наглядности, было решено перевести заданную фразу и дописать её новой строчкой на русском языке.

После этого, мы отчетливо можем видеть (см. рисунок 14) некоторые изменения как в количестве символов (нижний правый угол, поле size), так и в самом представлении букв кириллицы. Стоит отметить очевидный факт на лицо – в кодировке «UTF-8» поле размер является наибольшим из всех четырех кодировок.

Для того, чтобы пояснить на первый взгляд отклонение, нам потребуется глубже рассмотреть реализацию кодировки «UTF-8». Стоит внести факт о том, что данная кодировка полностью реализует кодировку «ANCII», следовательно все символы из кодировки «ANCII» буду находится в пределах одного байта. Далее, стоит обратить внимание на то, какую проблему решает кодировка «UTF-8». В отличии от своевременно других кодировок, которые имели фиксированное количество байт на символ, в данной кодировке используется специальная битовая маска (рисунок 15), благодаря которой у кодировки нету фиксированного количества байт. Таким образом, символы кириллицы в данной кодировке занимают два байта (см. рисунок 16), в отличии от остальных и представляются в виде двухбайтовой маски «UTF-8».

|  |
| --- |
|  |
| Рисунок  – Наглядное сравнение текстового файла в различных кодировках |
|  |
| Рисунок  – Битовая маска кодировки «UTF-8» |
|  |
| Рисунок  – Символ «т» в представлении кодировки «UTF-8» |

## Работа в «Visual Studio Code»

По открытию меню «View» в «Visual Studio Code», необходимые значения уже находились в включенном состоянии (см. рисунок 17).

|  |
| --- |
|  |
| Рисунок  – «Render Whitespace» и «Render Control Characters» |

Далее, чтобы установить расширение, необходимо в правой панели открыть вкладу «Extensions», в текстовом поле ввести название расширения, выбрать необходимое и нажать на кнопку «install», после этого можно будет наблюдать установленное расширение (см. рисунок 18).

|  |
| --- |
|  |
| Рисунок  – Установленное расширение «code-eol» |

Далее, открыв файл с обозначением конца «LF», на месте конца строки, можно увидеть (см. рисунок 19) стрелочку вниз, которая символизирует перенос строки. При изменении обозначения конца на «CRLF», отображаемый значок изменился на угловую стрелку (см. рисунок 20). Также не сложно сделать вывод о том, что табуляция в «Visual Studio Code» в нашем случае создается через пробелы, а сам символ пробела обозначается полупрозрачной, серой, точкой.

|  |
| --- |
|  |
| Рисунок  – Обозначение конца «LF» |
|  |
| Рисунок  – Обозначение конца «CRLF» |

По нажатию комбинации «ctrl+k ctrl+s», открывается окно «Keyboard Shortcuts». Этого также можно добиться, пройдя по пути «File – Preferences – Keyboard Shortcuts» (см. рисунок 21).

|  |
| --- |
|  |
| Рисунок  – Окно «Keyboard Shortcuts» |

Для того, чтобы создать свою комбинацию, необходимо в правом верхнем углу окна, нажать на кнопку «Open Keyboard Shortcuts (JSON)», после чего, мы увидели открывшийся файл в формате «json». В него мы внесли изменения, согласно заданию (см. рисунок 22).

|  |
| --- |
|  |
| Рисунок  – Файл «keybindings.json» |

При нажатии сочетания клавиш «alt+k m», на месте курсора появляется символ тире (см. рисунок 23).

|  |
| --- |
|  |
| Рисунок  – Результат нажатия сочетания клавиш |

# Контрольные вопросы

## Создание горячей клавиши в «Visual Studio Code» (alt+f x)

Для создания горячей клавиши, необходимо открыть окно «Keyboard Shortcuts». Для этого следует воспользоваться комбинацией клавиш «ctrl+k ctrl+s». После чего, откроется необходимое окно (см. рисунок 24).

|  |
| --- |
|  |
| Рисунок  – Окно «Keyboard Shortcuts» |

Далее, для создания своей комбинации, потребуется открыть представление окна в формате «json». Для этого, существует кнопка в правом, верхнем углу «Open Keyboard Shortcuts (JSON)» (см. рисунок 25).

|  |
| --- |
|  |
| Рисунок  – Кнопка |

Далее, следует дополнить файл json, кодом (см. рисунок 26).

|  |
| --- |
|  |
| Рисунок  – Файл «keybindings.json» |

При нажатии комбинации «alt+f x», на месте курсора появляется надпись «Name» (см. рисунок 27).

|  |
| --- |
|  |
| Рисунок  – Результат |

## Перевод числа «3CB» в десятичную систему счисления

Число 3CB переводится из 16-ой системы счисления в 10-ую, как 971. В 16-ой системе счисления, данное число представляется в виде символа «<».

## Отличие программы «Frhed» от «Free Hex Editor Neo»

Обе программы исполняют главную задачу: быть редактором бинарного кода, но всё же имеются отличия. Первое отличие, которое сразу бросается в глаза – интерфейс. В отличии от Frhed, FHEN имеет больше различных вкладок, которые необходимы в той или иной мере для выполнения различных задач (см. рисунок 28).

|  |
| --- |
|  |
| Рисунок  – Интерфейс «FHEN» |

Все отличия перечислять достаточно долго, но можно ограничиться основными: окно истории изменений исходного текста, инспектор значения (отображает различную информацию о выбранном значении), поддержка множества файлов (окон, табов), битовые операции над значениями (NOT, OR, AND, XOR, NEGATION, SHIFTS).

В целом, можно сказать, что программа «FHEN» имеет больший набор инструментов для работы с двоичными файлами, нежели «Frhed».

Вывод

В данной лабораторной работе, мы изучили устройство работы с такими программами, как «Visual Studio Code» и «Frhed». Также, за счёт данной работы, мы более углубленно рассмотрели реализацию некоторых кодировок, увидели разницу между ними и немало важным является понимание представления символов в компьютере, особенно технических.