Задание практикума II курса: Текстовый редактор

Алексей Сальников

Содержание

1 Описание

Требуется реализовать упрощённый построчный текстовый редактор, управляемый набором команд, который передаётся со стандартного потока ввода. В качестве внутреннего представления предлагается использовать список указателей на строки, размещенные в динамической памяти. Количество обрабатываемых строк и их длинна ограничена только объёмом доступной программе динамической памяти. В связи с тем, что строки, потенциально превышают ширину экрана, возможно несколько способов отображения строк «на экран» (в терминал, или в файл, если вывод был перенаправлен):

1. Длинные строки при выводе обрезаются, но в памяти хранятся полностью. Предусмотреть возможность просмотра всей строки по нажатию символов[1] ‘<’, ‘>’.

2. Длинные строки продолжаются на следующих строках «загибаются». Здесь необходимо предусмотреть маркер, который позволяет отделить визуально текущую строку от следующей.

[1] Желающие могутсамостоятельно разобраться с тонкостями работы с виртуальным терминалом в UNIX и «отлавливать» нажатие стрелочек

Предполагается, что такие параметры как ширина экрана и число строк, приходящихся на экран будут взяты из настроек терминала. Для этого необходимо изучить документацию на использование функции *ioctl* (см. man tty_ioctl). Если вывод осуществляется не в теримнал (канал, сокет, файл), то строки должны отображаться как есть, без каких либо «загибаний» и «обрезаний».

Работа с текстовым редактором начинается либо с чтения текста из файла (имя файла указывается в аргументах функции main), либо с создания нового текста, если файл оказался пустым, или не существовал, но его возможно создать по имени указанному в параметрах. Если имя файла не указано, то просто создаётся пустое множество строк.

Все строки в текстовом редакторе нумеруются с единицы и позиции в строках, так же с единицы. Если в командах, указан номер строки 0, то 0 – означает особый смысл и скогее всего отмечает место перед первой строкой.

После запуска должно показываться приглашение ко вводу «***editor:***».

2 Команды

В текстовом редакторе должны присутствовать группы команд. Предполагается, что команды выполняются в интерактивном режиме. Если поток ввода закрылся, то интерактивный режим невозможен, соответственно команда будет выполняться до тех пор, пока не потребуется реакция «с клавиатуры» в этот момент работа текстового редактора должна быть завершена, вся использовавшаяся память очищена.

Среди указываемому редактору команд могут встречаться комментарии. Комментарий начинается с символа '#' и следует до конца строки. Если комментарий присутствует в добавляемой строке в текст, то такой комментарий просто является частью текста. Комментарии, как и строчки состоящие только из пробельных символов должны просто пропускаться редактором.

Если команда написана в несколько слов, например ***print pages***, то между словами команд может следовать сколько угодно пробельных символов, в частности символы табуляции.

2.1 Группа команд просмотра текста

2.1.1 Команда set tabwidth

В связи с тем, что символы табуляции отображаются разной шириной пропуска, необходимо все символы табуляции при отображении в терминал заменять определённым числом пробелов. Данная команда регулирует число пробелов используемое при отображении. По умолчанию необходимо принять значение 8. пример команды: ***set tabwidth 4***.

2.1.2 Команда print pages

Команда поэкранного просмотра всего текста (сверху вниз и слева направо). Просмотр следующего экрана по нажатию пробел. Выход из команды по нажатию 'q', либо исчерпании строк в тексте. В случае запуска с перенаправленным выводом в файл, текст должен быть выведен без приостановок на ожидание символа пробел и символы табуляции не должны быть заменены символами пробел.

2.1.3 Команда print range

Данная команда используется для просмотра диапазона строк. После указания имени команды следуют номера выводящихся строк. Номера разделены пробельными символами и отделены от имени команды пробельными символами, исключая перевод строки. Если второе число в диапазоне не указано, то считается что вывод происходит от номера строки, заданной первым числом, до конца текста. если первое число тоже не указано, то печатается весь текст. Намопинаю, что строки нумеруются начиная с единицы. Пример команды: ***print range 10 20***.

2.1.4 Команда set wrap

Данная команда принимает на вход аргумент «*yes*» или «*no*» переключает режим отображения между «длинные строки обрезаются» и «длинные строки продолжаются». По умолчанию должен быть установлен режим «длинные строки продолжаются», а команда ***set wrap no***, должна переводить в режим «длинные строки обрезаются».

2.2 Группа команд вставки строк

2.2.1 Команда insert after

Производит вставку строки или группы строк после N-ой. Сперва указывается номер строки (формат по аналогии с предыдущей группой команд), а затем в двойных кавычках сама вставляемая строка. Если номер пропущен происходит вставка в конец текста. Если N равно 0, то – в начало текста. Пример команды:

```
insert after 3 "папа у Васи силён в\n \"математике\""
```

В строке символы могут быть экранированы символом '\'. Предусмотреть возможность встречания в строке служебных символов, таких как: перевод строки '\n', табуляция '\t', возврат каретки '\r'. Комбинация '\\' отменяет служебный смысл символа '\', тоесть означает в точности «написать» '\'.

Для вставки группы строк используются три идущие подряд двойные кавычки, которые могут начинаться с новой строки, и заканчиваются так же тремя подряд двойными кавычками. В целом аналогично тому, как это устроено в языке программирования *Python*. пример:

```
insert after """
/*
 * Здесь  будет конец моей
 * программы на Си.
 * Спасибо, что дочитали код.
 */
"""
```

2.3 Группа команд редактирования строк

2.3.1 Команда edit string

позволяет заменить символ в N-ой строке, начиная с M-ой позиции; после указания позиций следует заменяемый символ. Если необходимо вставить специальный символ, то можно использовать '\'. Если указана конструкция вне диапазона, то необходимо выдать сообщение об ошибке и замену не производить (сам редактор должен при этом продолжать свою работу, тоесть выдать приглашение ко вводу следующей команды).

Пример команды: ***edit string 2 60000 ё*** заменит символ во второй строке в позиции 60000 на символ 'ё'.

2.3.2 Команда insert symbol

Команда аналогична ***edit string***, но символ она вставляет в нужную позицию, а не заменяет. Если происходит «выход» за диапазон вправо — то вставка в конец; если за диапазон влево — то вставка в начало.

2.3.3 Команда replace substring

Команда ищет и заменяет подстроки в диапозоне строк. После имени команды идёт диапазон строк, так же как и для команды ***print range*** (если диапазон не указан, то поиск и замена производятся во всём тексте), затем две строки в двойных кавычках, возможно каждая на новой строке. Первая из них строка образец, которая будет найдена в тексте, следующая

строка, на которую будет производится замена. В заменяемой строке может встречаться экранированный перенос строки, тогда строку результат нужно будет разбить на несколько строк.

Вместо строки образца может присутствовать символ ‘^’ – это означает, что строку нужно приписать в начало каждой строки из диапазона, или символ ‘$’ – это означает что приписать в конец каждой строки из диапозона.

При написании реализации данной команды необходимо думать об объёме используемой памяти и быстродействии операции.

2.3.4 Команда delete range

Удаляет строки. Указывается диапазон, аналогично команде ***print range***. При этом первое значение не может быть пропущено и не может быть равным 0.

2.3.5 Команда delete braces

Позволяет удалить группы символов, расположенные между фигурными скобками ‘{’ и ‘}’. Скобки тоже должны быть удалены. Предполагается, что скобки сбалансированы, внутри каждой пары скобок могут быть другие скобки. В качестве аргумента указывается диапазон строк, если он не указан, то удаление производится во всём тексте. Если открывающая скобка началась на одной строке, а закрывающая на другой, то содержимое между строками должно быть удалено, а начало строки, где встретилась первая ‘{’ должно быть склеено со строкой, где встретилась парная к ней ‘}’. Если закрывающая ‘}’ так и не встретилась, то удаляем до конца текста.

2.4 Группа технических команд

2.4.1 Команда exit

Осуществляет выход из текстового редактора. Если текст не сохранён в файл, то выдаётся предупреждение и выход не осуществляется. Если всё-равно необходимо выйти, то вводится команда ***exit force***.

2.4.2 Команда read

Открывает файл, и в случае удачи замещает им все строчки текста. Имя файла не запоминается для последующего сохранения редактируемого текста в файл. Если текстом уже было ассоциировано какое-то имя, оно не будет затронуто командой. Имя файла в команде указывается в двойных кавычках.

Пример: ***read "каталог с описанием картинок/картинка 1.descr"***

2.4.3 Команда open

Аналогична ***read***, за исключением того, что имя файла запоминается для последующего сохранения редактируемого текста в него.

2.4.4 Команда write

Сохраняет текст в файл, если имя файла не указано, то запись происходит в «запомненное» имя файла. Если имя файла не ассоциировано, то диагностировать ошибку. Если имя файла было указано, а с текстом был ассоциирован файл, то ассоциация не изменяется. Если ассоциация была пустой, то берётся ассоциация из данной команды.

2.4.5 Команда set name

Задаёт ассоциацию с именем файла. Как и в команде ***read*** ассоциация задаётся строкой в двойных кавычках. Если строка пустая, то ассоциация с файлом удаляется.

2.4.6 Команда help

Печатает помощь по текстовому редактору для пользователя. В режиме аналогичном команде ***print pages***.

3 Прочее

Были отмечены трудности с текстом написанным в кодировке UTF8 если он набран не латиницей. Особенность UTF8 заключается в том, что в тексте иногда встречаются символы переключающие язык. Сами эти символы не отображаютс, но они влияют на то, как как отображается последующий набор байт. Заданный тап отображения остаётся до следующего символа переключающего язык. Поэтому про прочтении символа иногда нужно смотреть вперёд на несколько символов, а то при вставке можно попортить свой редактируемый текст.