Лабораторная работа №1 “Разработка оболочки”

Оболочка (shell), или интерпретатор командной строки, - это базовый пользовательский интерфейс операционной системы. Ваш задача состоит в том, чтобы написать простую оболочку - myshell - со следующими свойствами.

1. Оболочка должна поддерживать следующие внутренние команды.
    a. cd <directory> - смена текущего каталога по умолчанию на <directory>. Если аргумент <directory> отсутствует, вывести текущий каталог. Если каталог отсутствует, вывести соответствующее сообщение об ошибке. Эта команда должна также соответствующим образом изменять переменную среды PWD.
    b. clr - очистка экрана.
    c. dir <directory> - вывод содержимого каталога <directory>. г. environ - вывод всех переменных среды.
    d. environ - вывод всех переменных среды.
    e. echo <comment> - вывод на экран <comment>, после которого выполняется переход на новую строку (множественные пробелы/табуляции могут быть сокращены до единственного пробела).
    f. help - вывод руководства пользователя с использованием фильтра more.
    g. pause - приостановка операций оболочки до нажатия клавиши <Enter>.
    h. quit - выход из оболочки.
    i. Среда оболочки должна содержать переменную shell=<pathname>/myshell, где <pathname>/myshell - полный путь к выполнимому файлу оболочки (не "прошитый" путь к вашему каталогу, а тот, откуда была выполнена оболочка)
2. Все прочие входные данные командной строки интерпретируются как вызовы программ, которые должны выполняться оболочкой с использованием механизмов fork и exec как собственные дочерние процессы. Программы должны вы полняться в среде, содержащей переменную parent=<pathname>/myshell, где <pathname>/myshell такие же, как описано выше, в п. 1.i.
3. Оболочка должна быть в состоянии получать данные командной строки из файла. То есть оболочка вызывается с аргументом командной строки *myshell batchfile* предполагается, что файл batchfile содержит набор командных строк для обработки оболочкой. По достижении конца файла должно быть выполнено завершение работы оболочки. Очевидно, что если оболочка вызывается без аргумента командной строки, то она запрашивает ввод от пользователя через приглашение на дисплее.
4. Оболочка должна поддерживать перенаправление ввода-вывода для stdin и/или stdout, т.е. командная строка *programname argl arg2 < inputfile > outputfile* должна выполнить программу programname с аргументами argl и arg2 с заменой файлового потока stdin файлом inputfile, а файлового потока stdout - файлом outputfile. Перенаправление stdout должно также быть возможным для внутренних команд dir, environ, echo и help. При перенаправлении вывода символ > должен приводить к созданию outputfile, если такового не существует, и его усечению, если он имеется. При перенаправле нии >> файл outputfile создается, если он еще не существует, а если существует, выполняется дозапись в конец файла.

5. Оболочка должна поддерживать фоновое выполнение программ. Амперсанд (&) в конце командной строки указывает, что оболочка должна вернуться к командной строке сразу после запуска данной программы.
6. Приглашение командной строки должно содержать путь к текущему каталогу.

Примечание. Можно считать, что все аргументы командной строки (включая символы перенаправления <, > и >> и символ фонового выполнения &) будут отделены от других аргументов командной строки пробельными символами - одним или несколькими про белами и/или символами табуляции.

Требования к проекту.

1. Разработайте простую оболочку командной строки, которая удовлетворяет выше указанным критериям, и реализуйте ее на конкретной платформе UNIX.
2. Напишите простое руководство, описывающее, как использовать оболочку. Руководство должно содержать достаточно подробностей, чтобы новичок в UNIX мог его использовать. Например, вы должны объяснить концепции перенаправления ввода-вывода, программной среды и фонового выполнения программ. Руководство ДОЛЖНО быть названо readme и должно быть простым текстовым документом, который может быть прочитан стандартным текстовым редактором.В качестве примера требуемых глубины и типа описания вы можете рассмотреть онлайн-руководства для csh и tcsh (man csh, man tcsh). Эти оболочки, естественно, обладают гораздо большей функциональностью, чем разрабатываемая вами, а потому ваши руководства не обязательно должны быть такими большими. Вы НЕ должны включать в руководство инструкции по сборке, в том числе спис ки файлов или исходный код. Это должно быть руководство оператора, а не руководство разработчика.
3. Исходный код ДОЛЖЕН быть тщательно прокомментирован и соответствующим образом структурирован, чтобы ваши коллеги могли легко понимать и поддерживать ваш код.
4. Передаваемая на проверку работа должна содержать только файл (или файлы) исходного кода, включая makefile и файл readme. Никакая выполняемая программа не должна быть включена. Проверяющий должен выполнить построение вашей оболочки из предоставленного исходного кода. Если представленный код не компилируется, работа считается невыполненной.
5. makefile ДОЛЖЕН генерировать бинарный файл myshell. Вот пример про стейшего makefile
6. В показанном выше примере в представленном каталоге должны быть файлы

```
myshell: myshell.c utility.c myshell.h
    gcc -Wall myshell.c utility.c -o myshell
```

7. В показанном выше примере в представленном каталоге должны быть файлы

```
makefile
myshell.c
utility.c
myshell.h
readme
```