

И. А. Буяковская

ОСОБЕННОСТИ ПРЕПОДАВАНИЯ РАЗДЕЛА «ОПЕРАЦИОННЫЕ СИСТЕМЫ» НА ОСНОВЕ LINUX В ДИСЦИПЛИНЕ «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ»

В связи с тем, что в рамках приоритетного национального проекта "Образование" российские школы должны перейти на использование пакета свободного программного обеспечения (ПСПО) как альтернатива коммерческому. Курс предметной подготовки будущих учителей «Программное обеспечение ЭВМ» был обновлен под операционную среду Linux и свободное прикладное программное обеспечение.

В соответствии со стандартом раздел «Операционные системы» включает следующие темы: операционные системы (ОС) как средство распределения и управления ресурсами, развитие и основные функции ОС, состав ОС: внутренние (встроенные) и внешние (программы-утилиты), команды ОС, сетевые ОС.

Темы данного раздела раскрываются в рамках лекционных занятий: управление процессами, задачами и данными в операционных системах; эволюция операционных систем и их классификация; операционная система Linux; работа в KDE; сетевые операционные системы; операционные оболочки и их назначение, которые рассчитаны на 6 часов. Практические занятия посвящены знакомству с командной строкой Linux и изучению основных команд, созданию скрипта в shell, написанию простейших программ на языке Си, работу с файловым менеджером, знакомству со служебными программами и рассчитаны на 8 часов.

Ниже представлен фрагмент занятия по теме «Создание скрипта в shell», рассчитанный на 2 часа.

Теоретический материал: командный язык shell - это язык программирования высокого уровня (используется для создания скриптов в Linux). Shell (оболочка) - командный процессор, принимающий команды пользователя, анализирующий их и выполняющий необходимые действия. Наиболее часто в Linux употребляется оболочка bash, она же вызывается по команде sh. Скриптовый язык - язык программирования, разработанный для записи «сценариев», последовательностей операций, которые пользователь может выполнять на компьютере. Простые скриптовые языки раньше часто называли языками пакетной обработки (batch languages или job control languages). Сценарии обычно интерпретируются, а не компилируются (хотя часто сценарии компилируются каждый раз перед запуском). В прикладной программе, сценарий (скрипт) - это программа, которая автоматизирует некоторую задачу, которую без сценария пользователь делал бы вручную, используя интерфейс программы. Для написания пользовательских расширений могут использоваться как скрипты (в терминологии некоторых программ «макросы»), так и **плагины** (независимые модули, написанные на компилируемых языках; в некоторых программах они могут называться «утилитами», «экспортёрами», «драйверами»).

Задание 1: Создайте скрипт, подсчитывающий количество файлов в текущей папке.

Создайте в своей рабочей папке файл count и сделайте его исполняемым (наберите команду `chmod 775 count`). Теперь откройте этот файл для редактирования любым текстовым редактором (если вы работаете в `ms`, наведите выделение на этот файл и нажмите клавишу `F4`, если командной строке - наберите, например, `kwrite count`).

В самом начале файла наберите строку:

```
#!/bin/sh
```

Она необходима для того, чтобы командный процессор распознал данный файл как скрипт. Далее наберите непосредственно текст скрипта:

```
COUNT=-2 # Не считаем папки . и ..
```

```
for FILE in `ls -a` # Начало цикла
```

```
do # по списку файлов в папке
```

```
COUNT=`expr $COUNT + 1` # Увеличить счетчик на 1
```

```
done # Конец цикла
```

```
echo "Папка содержит $COUNT файлов и папок" # Вывод результата
```

- Обратите внимание, что кавычки, в которые заключены команды `ls` и `expr`, получают при нажатии буквы ``` (слева от 1) в английской раскладке. Они означают, что заключенный в них текст будет выполнен как команда, а результат ее - подставлен вместо кавычек.
- Знаком `#` отмечаются комментарии, все, что написало в строке после него, не будет считано командным интерпретатором, поэтому, в принципе, комментарии можно не записывать. Они нужны только для лучшего понимания скрипта. Единственный комментарий, который обязательно следует записать - указание командной оболочки, что для исполнения файла необходимо использовать `bash` (`#!/bin/sh`).
- Здесь `COUNT` - счетчик, который будет содержать количество файлов и папок в текущей папке. Далее в программе следует цикл, в котором переметная `FILE` принимает все значения из листинга текущей папки. На каждый файл переменная `COUNT` увеличивается на 1.

В ходе выполнения данного задания мы выявили следующую особенность: если до и после знака арифметической операции «+» не ставить пробелы то в итоге формируется не числовое значение, а строка.

Знак `$` перед именем переменной позволяет получить ее значение, а команда `expr` представляет собой простейший калькулятор командной строки, возвращая значение выражения, стоящего после нее.

Набрав и сохранив программу, выполните ее, для чего в командной строке ее можно выполнить двумя способами: набрав либо `sh count`, либо `./count`.

Задание 2: Создайте скрипт, выполняющий простые арифметические вычисления (Для этого создайте в своей рабочей папке файл count2).

В новом файле добавьте строку, необходимую для исполнения файла shell.

```
x=7 y=2 #  
a='expr $x + $y'; echo a=$a #Вычисляем значения переменн  
a='expr $a + 1'; echo a=$a #и выводим н  
b='expr $y - $x'; echo b=$b  
c='expr $x '*' $y'; echo c=$c  
d='expr $x / $y'; echo d=$d  
e='expr $x % $y'; echo e=$e
```

По аналогии создайте файл count3, содержащий скрипт выполняющий не только (целочисленные) арифметические операции, но и строковые:

```
A='expr 'cocktail' : 'cock''; echo $A  
B='expr 'cocktail' : 'tail''; echo $B  
C='expr 'cocktail' : 'co''; echo $C  
D='expr 'cock' : 'cocktail''; echo $D
```

На экран будут выведены числа, показывающее число совпадающих символов в

Далее выполняется задание на составление программы подсчитывающей отдельно количество файлов, папок и исполняемых файлов.

Список литературы

1. Соловьев А. Программирование на shell (UNIX): Учебное пособие. [Электронный ресурс] - Электронные текстовые данные (3 326 424 bytes) режим доступа: <http://www.linuxcenter.ru/lib/books/shell/> 16.03.2010 12:05
2. Емельянов Э.В. Лабораторный практикум по информатике. [Электронный ресурс] - Электронные текстовые данные (37 634 bytes) режим доступа: http://freecode.pspo.perm.ru/003/Lab_Pr.pdf 10.09.2010 10:10