Министерство образования и науки Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

В.Г. КОБЫЛЯНСКИЙ

ОПЕРАЦИОННЫЕ СИСТЕМЫ, СРЕДЫ И ОБОЛОЧКИ

Утверждено
Редакционно-издательским советом университета
в качестве учебного пособия

НОВОСИБИРСК
2018

УДК 004.451(075.8)
К 558

Рецензенты:
д-р техн. наук, профессор *В.И. Хабаров*
канд. физ.-мат. наук, доцент *М.Г. Зайцев*

Кобылянский В.Г.
К 558 Операционные системы, среды и оболочки: учебное пособие / В.Г. Кобылянский. – Новосибирск: Изд-во НГТУ, 2018. – 80 с.

ISBN 978-5-7782-3517-5

Учебное пособие по курсу «Операционные системы, среды и оболочки» предназначено для студентов направлений «Прикладная математика и информатика» и «Математическое обеспечение и администрирование информационных систем» и содержит описание языка shell, файловых систем и инструментальных средств, предназначенных для подготовки к исполнению программ пользователей. Все темы изучаются на примере операционных систем Linux и Windows.

Пособие может быть полезным для студентов всех направлений, изучающих принципы функционирования операционных систем и технологии разработки программного обеспечения.

Работа подготовлена на кафедре «Теоретическая и прикладная информатика»

Кобылянский Валерий Григорьевич

ОПЕРАЦИОННЫЕ СИСТЕМЫ, СРЕДЫ И ОБОЛОЧКИ

Учебное пособие

Редактор *И.Л. Кескевич*
Выпускающий редактор *И.П. Брованова*
Корректор *И.Е. Семенова*
Дизайн обложки *А.В. Ладыжская*
Компьютерная верстка *Н.В. Гаврилова*

Налоговая льгота – Общероссийский классификатор продукции
Издание соответствует коду 95 3000 ОК 005-93 (ОКП)

Подписано в печать 02.04.2018. Формат 60 × 84 1/16. Бумага офсетная
Тираж 80 экз. Уч.-изд. л. 4,65. Печ. л. 5,0. Изд. 373/17. Заказ № 557
Цена договорная

Отпечатано в типографии
Новосибирского государственного технического университета
630073, г. Новосибирск, пр. К. Маркса, 20

УДК 004.451(075.8)

ISBN 978-5-7782-3517-5

© Кобылянский В. Г., 2018
© Новосибирский государственный технический университет, 2018

ОГЛАВЛЕНИЕ

Введение 5

1. ОСНОВЫ КОМАНДНОГО ИНТЕРФЕЙСА ОПЕРАЦИОННОЙ СИСТЕМЫ 6

1.1. Общие сведения 6
1.2. Переменные сценария 9
1.3. Параметры сценария 11
1.4. Операторы языка Shell 12
1.4.1. Оператор TEST 12
1.4.2. Оператор IF 13
1.4.3. Оператор CASE 14
1.4.4. Оператор FOR 14
1.4.5. Операторы WHILE и UNTIL 15
1.4.6. Использование функций в сценариях 15
1.5. Создание сценария 15
1.6. Практическое задание 16
1.7. Контрольные вопросы 17

2. ФАЙЛОВЫЕ СИСТЕМЫ СОВРЕМЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМ 18

2.1. Модель внешней памяти 19
2.1.1. Физическая модель диска 19
2.1.2. Логическая модель диска 20
2.2. Файловые системы операционной системы Windows 24
2.2.1. Файловая система FAT 24
2.2.2. Файловая система NTFS 29
2.2.3. Инструментальные средства анализа файловой системы 36
2.3. Файловые системы операционной системы Linux 42
2.3.1. Общие сведения 42
2.3.2. Файловая система s5 45
2.3.3. Файловая система ext 47

2.3.4. Команды анализа файловой системы 49
2.4. Практическое задание 50
2.4.1. Файловые системы Windows 51
2.4.2. Файловые системы Linux 53
2.5. Контрольные вопросы 53
3. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММ 55
3.1. Прохождение программ в среде операционной системы 55
3.2. Компиляция исходного модуля 57
3.3. Отладка и тестирование 60
3.4. Система управления версиями 62
3.5. Сборка программы сложной структуры 66
3.6. Практическое задание 70
3.6.1. Описание задания 70
3.6.2. Выполнение задания 74
3.7. Контрольные вопросы 76
Библиографический список 77
Краткий словарь терминов 78

ВВЕДЕНИЕ

Современные молодые программисты нередко слабо знакомы с алгоритмами, которые реализует операционная система, управляя системными ресурсами и доступом к ним исполняемых программ. Высокая производительность и большие объемы памяти современных компьютеров дают возможность не задумываться об экономии этих ресурсов, а интегрированные среды разработки программного обеспечения (ПО) скрывают от своих пользователей все детали, связанные с подготовкой программ к выполнению, и не позволяют в полной мере понять такие важные вопросы, как структура адресного пространства процесса, методы сборки и отладки исполняемого модуля, работа с картой памяти и т. д.

Целью настоящего пособия является рассмотрение вопросов, обычно вызывающих затруднения при изучении дисциплины «Операционные системы». В первом разделе рассмотрены основы командного интерфейса, второй раздел содержит описание различных файловых систем, а третий – посвящен изучению инструментальных средств разработки ПО. В конце пособия приведен краткий словарь используемых терминов. Все темы изучаются в основном на примере операционной системы Linux, по ОС Windows рассмотрены только вопросы функционирования файловых систем FAT5 и NTFS.

Каждый раздел содержит теоретический материал, комплексное практическое задание и вопросы для самоконтроля. Практическое задание представляет собой набор действий, выполнив которые, вы сможете более глубоко усвоить теоретический материал. Для выполнения задания, в дополнение к разделам пособия, желательно изучить приведенные источники информации, содержащие методические рекомендации и примеры программ.

Практическое задание можно выполнять в компьютерных классах факультета прикладной математики на сервере **students.ami.nstu.ru** или на домашнем компьютере. Удаленное подключение к серверу можно проводить с помощью программы **putty**.

Изучение пособия желательно начинать с первого раздела, остальные разделы могут изучаться в произвольной последовательности.

1. ОСНОВЫ КОМАНДНОГО ИНТЕРФЕЙСА ОПЕРАЦИОННОЙ СИСТЕМЫ

Прочитав этот раздел, вы узнаете:

- *формат командной строки*;
- *классификацию команд*;
- *способы передачи данных в сценарий*;
- *определение следующих терминов*: *командный файл, сценарий, командный интерпретатор, стандартные переменные.*

1.1. ОБЩИЕ СВЕДЕНИЯ

Командный интерфейс исторически был первым типом интерфейса, предоставляемого операционной системой (ОС) пользователю, и представлял собой набор команд, каждая из которых выполняла определенное действие. Этот интерфейс требует от пользователя более глубоких знаний устройства компьютера и ОС, поэтому он прежде всего предназначен для применения IT-специалистами.

Основное достоинство командного интерфейса – доступность для пользователя абсолютно всех возможностей, заложенных в систему разработчиками ОС, а недостаток – его сложность, так как число команд может быть достаточно большим и каждая команда может иметь свой набор ключей (опций), задающих определенный режим выполнения этой команды.

Командная строка в общем случае имеет следующий формат:

команда [*ключи*] [*аргументы*]

В качестве аргументов команды выступают имена устройств, каталогов или файлов, а ключи уточняют действие команды. Ключи и аргументы не обязательны для всех команд, т. е. существуют команды, не имеющие параметров (например, для очистки экрана).

Обработка командной строки выполняется интерпретатором команд, входящим в состав ОС. В семействе ОС Windows основным интерпретатором команд является программа **cmd**, в семействе Linux – программы **sh** или **bash**. Часть команд интерпретатор выполняет самостоятельно, такие команды называются *внутренними*. Остальные команды реализуются программами, поставляемыми вместе с ОС в виде отдельных файлов, и называются *внешними*. Пути поиска файлов внешних команд хранятся в системной глобальной переменной PATH.

Алгоритм исполнения команд следующий. Сначала интерпретатор пытается выполнить заданную команду самостоятельно, считая ее внутренней. Если он не находит имя команды в своей внутренней таблице, то начинает поиск исполняемого файла, имя которого совпадает с именем команды, во всех каталогах, указанных в переменной PATH. При нахождении такого файла интерпретатор запускает соответствующую программу, иначе – выводит сообщение об ошибке «*Файл не является внутренней или внешней командой, исполняемой программой или пакетным файлом*».

В командной строке можно запускать любые исполняемые файлы, они рассматриваются интерпретатором как обычные внешние команды.

В табл. 1 приведены некоторые часто используемые команды для ОС Windows и Linux. Более полное описание команд можно найти в [1, 6, 7].

Таблица 1

Действие	Linux	Windows	Пример
Создать каталог	mkdir	md	mkdir abc mykat
Удалить каталог	rmdir	rd	rmdir abc
Вывести имя текущего каталога	pwd	cd (без параметров)	pwd
Изменить текущий каталог	cd	cd	cd /u/home/bpi
Просмотреть каталог	ls	dir	ls mykat
Копировать файл	cp	copy, xcopy	cp myfile1 myfile2
Переместить файл	mv	move, ren	mv mykat mykat_new
Удалить файл	rm	del	rm my.txt
Получить справку по команде	команда /?	man	man ls
Просмотреть файл	cat	type	cat file1

Окончание табл. 1

Действие	Linux	Windows	Пример
Постранично просмотреть файл	more	more	more myfile
Вывести версию ОС	uname	ver	uname -a
Очистить экран	clear	cls	clear
Искать заданную строку в файле	grep	find	grep “winter” my.txt
Сортировать файл	sort	sort	sort -nk 2 debts.txt

В одной командной строке можно записывать несколько команд, в этом случае порядок выполнения команд зависит от используемых разделителей (табл. 2).

Таблица 2

Порядок выполнения команд	Windows	Linux
Команды выполняются последовательно	Команда1 & команда2	Команда1; команда2
Вторая команда выполняется только в случае успешного завершения первой команды	Команда1 && команда2	Команда1 && команда2
Вторая команда выполняется только тогда, когда первая команда не была выполнена	Команда1 || команда2	Команда1 || команда2
Команды выполняются последовательно, результаты выполнения первой команды передаются на вход второй команды	Команда1 | команда2	Команда1 | команда2

Современные командные интерпретаторы могут исполнять не только отдельные команды, но и программы, написанные на языке программирования **shell**. Такие программы называются *командным сценарием*, *скриптом* или *командным файлом* и используются для администрирования и управления ОС. Командный сценарий – это исполняемый текстовый файл, каждая строка которого является командой ОС или вызовом некоторой программы. Интерпретатор последовательно читает и исполняет все строки такого файла.

Наиболее развитым языком **shell** характеризуются интерпретаторы семейства ОС Linux, которые поддерживают ввод-вывод переменных,

ветвления (if-then-else, case), циклы (for, while, until), работу с функциями, пошаговый режим отладки и другие возможности, характерные для языков высокого уровня. Стандартный командный интерпретатор ОС семейства Windows **cmd** уступает по функциональности интерпретаторам Linux, однако в конце 2006 г. фирма Microsoft выпустила для ОС семейства Windows оболочку PowerShell, предоставляющую мощный интерфейс командной строки и встроенный язык сценариев.

Запуск сценария Linux может проводиться несколькими способами:

- запустить интерпретатор, указав ему в качестве аргумента имя файла сценария, например:

sh myscript.sh

- выполнить сценарий в текущем экземпляре shell, например:

. myscript.sh или **source myscript.sh**

При написании сценария необходимо соблюдать следующие требования:

- в первой строке обычно указывается имя интерпретатора, который будет исполнять сценарий, например #!bin/bash;
- в конце сценария обязательно вводится символ перевода строки (нажатие ENTER).

1.2. ПЕРЕМЕННЫЕ СЦЕНАРИЯ

Каждый запущенный экземпляр командного интерпретатора **shell** имеет определенный набор стандартных глобальных переменных, значения которых можно просмотреть командами **env** или **printenv**. В табл. 3 приведены назначения некоторых переменных.

Пользователь может использовать в сценариях глобальные переменные интерпретатора и собственные локальные переменные. Имена переменных представляют собой последовательность букв, цифр и символов подчеркивания, имя должно начинаться с буквы или символа подчеркивания. Переменные всегда имеют строковый тип и задаются сразу со значением (возможно с пустым).

При объявлении переменной ее имя и значение должны быть записаны без пробелов относительно символа равенства. Например, создадим две переменные, присвоив переменной VAR_1 значение “245”, а переменной VAR_2 – пустое значение:

```
VAR_1=245
VAR_2=
```