

**Instructor Guide**  
**Get Ready for Open Source:**  
**SUSE Linux Enterprise Desktop**  
**Book 2**

Novell Training Services

2007

## Proprietary Statement

Copyright © 2007 Novell, Inc. All rights reserved.

The reproduction, photocopying, storing on a retrieval system, or transmitting of this manual is protected under a Creative Commons Attribution-NonCommercial-ShareAlike 2.5 license.

You are free to share (copy, distribute, display, and perform the work) and to remix (make derivative works) under the following conditions: you must attribute the work in the manner specified by the author or licensor, you may not use this work for commercial purposes. In addition, if you alter, transform, or build upon this work, you may distribute the result work only under a license identical to this one.

For any reuse or distribution, you must make clear to others the license terms of this work. For clarification or to apply for a waiver to any of these conditions, contact Novell, Inc.

Novell, Inc.  
1800 South Novell Place  
Provo, UT 84606-2399

## Disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to revise this publication and to make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any NetWare software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to make changes to any and all parts of NetWare software at any time, without obligation to notify any person or entity of such changes.

This Novell Training Manual is published solely to instruct students in the use of Novell networking software. Although third-party application software packages are used in Novell training courses, this is for demonstration purposes only and shall not constitute an endorsement of any of these software applications.

Further, Novell, Inc. does not represent itself as having any particular expertise in these application software packages and any use by students of the same shall be done at the students' own risk.

## Software Piracy

Throughout the world, unauthorized duplication of software is subject to both criminal and civil penalties.

If you know of illegal copying of software, contact your local Software Antipiracy Hotline.

For the Hotline number for your area, access Novell's World Wide Web page at <http://www.novell.com> and look for the piracy page under "Programs".

Or, contact Novell's anti-piracy headquarters in the U.S. at 800-PIRATES (747-2837) or 801-861-7101.

## Acknowledgements

Michael Eicks (Novell Inc.)

## Trademarks

Novell, Inc. has attempted to supply trademark information about company names, products, and services mentioned in this manual. The following list of trademarks was derived from various sources.

### Novell, Inc. Trademarks

Novell, the Novell logo, NetWare, BorderManager, ConsoleOne, DirXML, GroupWise, iChain, ManageWise, NDPS, NDS, NetMail, Novell Directory Services, Novell iFolder, Novell SecretStore, Ximian, Ximian Evolution and ZENworks are registered trademarks; CDE, Certified Directory Engineer and CNE are registered service marks; eDirectory, Evolution, exteNd, exteNd Composer, exteNd Directory, exteNd Workbench, Mono, NIMS, NLM, NMAS, Novell Certificate Server, Novell Client, Novell Cluster Services, Novell Distributed Print Services, Novell Internet Messaging System, Novell Storage Services, Nsure, Nsure Resources, Nterprise, Nterprise Branch Office, Red Carpet and Red Carpet Enterprise are trademarks; and Certified Novell Administrator, CNA, Certified Novell Engineer, Certified Novell Instructor, CNI, Master CNE, Master CNI, MCNE, MCNI, Novell Education Academic Partner, NEAP, Ngage, Novell Online Training Provider, NOTP and Novell Technical Services are service marks of Novell, Inc. in the United States and other countries. SUSE is a registered trademark of SUSE LINUX AG, a Novell company. For more information on Novell trademarks, please visit <http://www.novell.com/company/legal/trademarks/tmlist.html>.

### Other Trademarks

Adaptec is a registered trademark of Adaptec, Inc. AMD is a trademark of Advanced Micro Devices. AppleShare and AppleTalk are registered trademarks of Apple Computer, Inc. ARCServ is a registered trademark of Cheyenne Software, Inc. Btrieve is a registered trademark of Pervasive Software, Inc. EtherTalk is a registered trademark of Apple Computer, Inc. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Linux is a registered trademark of Linus Torvalds. LocalTalk is a registered trademark of Apple Computer, Inc. Lotus Notes is a registered trademark of Lotus Development Corporation. Macintosh is a registered trademark of Apple Computer, Inc. Netscape Communicator is a trademark of Netscape Communications Corporation. Netscape Navigator is a registered trademark of Netscape Communications Corporation. Pentium is a registered trademark of Intel Corporation. Solaris is a registered trademark of Sun Microsystems, Inc. The Norton AntiVirus is a trademark of Symantec Corporation. TokenTalk is a registered trademark of Apple Computer, Inc. Tru64 is a trademark of Digital Equipment Corp. UnitedLinux is a registered trademark of UnitedLinux. UNIX is a registered trademark of the Open Group. WebSphere is a trademark of International Business Machines Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation.

All other third-party trademarks are the property of their respective owners.

# Contents

- 1 Introduction 1**
  - 1.1 Get Ready for Open Source . . . . . 1
  - 1.2 Audience . . . . . 1
  - 1.3 Prerequisites . . . . . 1
  - 1.4 Setup Requirements . . . . . 2
  - 1.5 Key Course Objectives . . . . . 2
  - 1.6 Agenda . . . . . 2
    - 1.6.1 Administer Linux with YaST . . . . . 2
    - 1.6.2 The Linux Shell and Tools . . . . . 2
    - 1.6.3 Linux Services for Developers . . . . . 3
  
- 2 Administer Linux with YaST 4**
  - 2.1 Manage the Network . . . . . 4
  - 2.2 Manage Users and Groups . . . . . 5
  - 2.3 Install and Remove Software . . . . . 6
  - 2.4 Configure a Printer . . . . . 7
  
- 3 The Linux Shell and Tools 8**
  - 3.1 Organize Files at the Command Line . . . . . 8
  - 3.2 Understand bash Commands and Features . . . . . 14
  - 3.3 Understand Advanced Command Execution . . . . . 15
  - 3.4 The vi Text Editor . . . . . 17
  - 3.5 The Basics of bash Programming . . . . . 18
  - 3.6 Understand the sed Stream Editor . . . . . 29
  
- 4 Linux Services for Developers 30**
  - 4.1 The Version Control System Subversion . . . . . 30
  - 4.2 The Web Server Apache . . . . . 34



---

# 1 Introduction

## 1.1 Get Ready for Open Source

The course “Get Ready for Open Source: SUSE Linux Enterprise Desktop” is designed for schools and universities. After three years the students should be able to take part at an open source development project.

Book 1 – Use the Linux GUI and the most important applications.

Book 2 – Do some basic administration tasks. Use the bash shell and write some simple shell programs.

Book 3 – Write programs by using the programming language Mono/C#.

Each book has content for about 60 hours and a lot of examples and exercises. The students are motivated to learn Linux and developing software by exploration.

## 1.2 Audience

The course is designed for students of around 14 years of age who already know the Linux GUI and the most important Linux applications.

## 1.3 Prerequisites

The course is designed for programming beginners. Students should be familiar with the GUI and most important applications of the SUSE Linux Enterprise Desktop. Programming and administration knowledge is not needed.

## 1.4 Setup Requirements

Each student and the instructor should have a computer with an SUSE Linux Enterprise Desktop 10 installation.

## 1.5 Key Course Objectives

During this course, you will learn how to complete the following tasks:

- ☞ Understand Important System Administration Tasks
- ☞ Manage Files with the Shell
- ☞ Write Shell Programs
- ☞ Setup Services that are Useful when Developing

## 1.6 Agenda

Chapter	Schedule
Administer Linux with YaST	8 hours
The Linux Shell and Tools	44 hours
Linux Services for Developers	8 hours
Sum	60 hours

### 1.6.1 Administer Linux with YaST

Section	Schedule
Manage the Network	2 hours
Manage Users and Groups	2 hours
Install and Remove Software	2 hours
Configure a Printer	2 hours

### 1.6.2 The Linux Shell and Tools

---

Section	Schedule
Introduction	1 hours
Organize Files at the Command Line	7 hours
Understand bash Commands and Features	4 hours
Understand Advanced Command Execution	6 hours
The vi Text Editor	4 hours
The Basics of bash Programming	16 hours
Understand the sed Stream Editor	6 hours

### 1.6.3 Linux Services for Developers

Section	Schedule
The Version Control System Subversion	4 hours
The Web Server Apache	4 hours

## 2 Administer Linux with YaST

### 2.1 Manage the Network

**Learning target:** Students understand the basic of computer networks and how to integrate SUSE Linux Enterprise Desktop 10 into an existing network.

#### Answers Exercise “Understand IP Addresses”

In IP-based networks, each computer (or each network interface of a computer) has a unique, 32-bit IP address. For the sake of readability, these 32 bits are not shown as a sequence of 32 zeros and ones, but are divided into 4 bytes. These four bytes, called octets, are separated by dots (32-bit/4-byte dot notation, or dotted quad notation) and are recorded either as decimal or binary numbers.

For example, 32 bits “in sequence” from the machine’s point of view looks like the following:

```
11000000 10000001 00110010 00000001
```

Readable IP representation in decimal format looks like the following:

```
192.129.50.1
```

An IP address consists of the network prefix (the front part of the IP address) and a host number (the end part of the IP address). The network prefix helps to determine the network class in which the host is located. By means of the IP address, data is delivered to the required host in the destination network.

### Answers Exercise “Manage the Network”

You can find these information in the YaST Network module (YaST → Network Devices → Network Card → Traditional Method with ipup) *rightarrow* Select a network card in the upper frame → Edit.

- ① In the Address tab: Is Automatic Setup via DHCP selected?
- ② In the Address tab: Content of the textbox IP Adress
- ③ In the Address tab, select Hostname and Name Server → Accept (if needed) → Textboxes Name Server X
- ④ In the Address tab, select Routing → Default Gateway

### Answers Exercise “View Your Network Settings”

This is described above the exercise.

## 2.2 Manage Users and Groups

**Learning target:** The students understand how to manage local user and group accounts.

### Answers Exercise “Manage Users and Groups”

For example,...

- ① Teachers (normal)
- ② IT Teachers
- ③ Students/Pupils
- ④ Webpage Carer
- ⑤ Network Administrator
- ⑥ Secretary

### Answers Exercise “Create a New User”

Start YaST → Select **Security and Users** → Select **User Management** → Select **Add** → Enter the user’s information, e. g.

User’s Full Name	Tux Penguin
Username	tux
Password	novell
Confirm Password	novell

Select **Accept** → Select **Yes** (depends) → Select **Yes** (depends) → Select **Finish**

The new user account is needed later in this course.

### Answers Exercise “Administer Groups”

Start YaST → Select **Security and Users** → Select **Group Management** → Select the appropriate option from the **Set Filter** menu.

## 2.3 Install and Remove Software

**Learning target:** Students are able to install software packages that are available for SUSE Linux Enterprise Desktop 10.

### Answers Exercise “Install Software – Part I”

Start YaST → Select **Software** → Select **Software Management** → Enter **locate** in the the **Search** textbox → Select **Search** → Select **findutils-locate** → Select **Accept** → Select **No**

The software of the package **findutils-locate** is needed later in this course.

## Answers Exercise “Install Software – Part II”

Start YaST → Select Software → Select Software Management → Enter `ksh` in the the Search textbox → Select Search → Select `ksh` → Select Accept → Select No

The software of the package `ksh` is needed later in this course.

## 2.4 Configure a Printer

**Learning target:** Students are able to add a printer that is connected directly or via network.

### Answers Exercise “Configure a Printer”

Start YaST → Select Hardware → Select Printer → Select Add → Select New Queue for existing Printer → Select Next → Select your printer → Select Next

Depending on your printer, you have different options:

- ☞ Select PPD file → Select Edit → Select another PPD file with a lower resolution from the list → Select OK
- ☞ Select Filter Options → Select Edit → Select an option that is responsible for the resolution from the upper list → Select a value that is responsible for the resolution from the bottom list → Select Next

## 3 The Linux Shell and Tools

### Answers Exercise “Understand the Linux Shells”

- ❶ A – Bourne Again Shell
- ❷ C – Tenex C Shell
- ❸ B – Korn Shell

### 3.1 Organize Files at the Command Line

**Learning target:** The students are able to do the most important file management tasks at the command line.

### Answers Exercise “Move in the File System”

Current Position	Command	Final Position
/var/log/	cd cups	/var/log/cups/
/usr/X11R6/	cd bin	/usr/X11R6/bin/
/usr/X11R6/	cd /bin	/bin
/usr/share/	cd ..	/usr
/etc/postfix/	cd /	/
~	cd ..	/home
/home	cd ..	/
/media/cdrom/	cd ../../lib	/lib
/sbin	cd	~

### Answers Exercise “List Directory Content – Part I”

Directory	Number of Subdirectories
/	18
/sbin	1
/var/log/	9
/etc/init.d/	9
/usr/share/doc/	3
/bin	0

### Answers Exercise “List Directory Content – Part II”

In the following geeko is used for the current user.

File/Directory	User	Group	Permission
~/ .nautilus/	geeko	users	rwxr-xr-x
/tmp	root	root	rwxrwxrwt
/var/spool/mail/	root	root	rwxrwxrwt
/etc/init.d/autofs	root	root	rwxr--r--
/usr/bin/passwd	root	shadow	rwsr-xr-x
/etc/passwd	root	root	rw-r--r--
/etc/shadow	root	shadow	rw-r-----
/usr/bin/wall	root	tty	rwxr-sr-x

### Answers Exercise “Move and Copy Files”

- 1 The permissions are equal. The file owner and owning group changed.

```

1 geeko@da51:~> ls -l /etc/init.d/syslog
2 -rwxr-xr-- 1 root root 3627 2006-06-16 09:18 /etc/init.d/syslog
3 geeko@da51:~> cp /etc/init.d/syslog ~
4 geeko@da51:~> ls -l syslog
5 -rwxr-xr-- 1 geeko users 3627 2006-10-12 06:55 syslog
6 geeko@da51:~>

```

- 2 mv syslog testfile
- 3 cp /media/mountpoint/example/grep-demo ~

**Answers Exercise “Create a Directory”**

- ❶ `mkdir ~/Marketing`
- ❷ `mv ~/testfile ~/Marketing/`
- ❸ `mkdir -p /tmp/School/Mathematics`
- ❹ `cp -r ~/Marketing /tmp/School/Mathematics/`
- ❺ `mkdir /tmp/School/German`

**Answers Exercise “Delete Directories and Files”**

- ❶ `rm /tmp/School/Mathematics/Marketing/testfile`
- ❷ `rmdir /tmp/School/German/`
- ❸ `rm -r /tmp/School/`

**Answers Exercise “Link Files”**

- ❶ 1 link (boot -> .)
- ❷ `ln -s ~/Marketing/testfile ~/symlink`
- ❸ `ln /Marketing/testfile /hardlink`
- ❹ `rm Marketing/testfile`
- ❺ Test symbolic link:

```
1 geeko@da51:~> cat symlink
2 cat: symlink: No such file or directory
```

Test hard link:

```
1 geeko@da51:~> cat hardlink
2 #! /bin/sh
3 #
4 # Copyright (c) 1996-2001 SuSE GmbH Nuernberg, Germany.
5 # Copyright (c) 2002-2003 SuSE Linux AG, Nuernberg, Germany.
6 #
...

```

### Answers Exercise “Find Files”

- ① Enter `find /bin -size +1024k`
- ② Enter `find /etc -group lp`
- ③ Enter `find / -name grep`

### Answers Exercise “Locate Files”

- ① `/usr/bin/updatedb`
- ② `/etc/X11/XF86Config`
- ③ `/etc/samba/smb.conf`
- ④ `/home/geeko/hardlink`
- ⑤ `/var/log/boot.log`
- ⑥ `/sbin/installkernel`

### Answers Exercise “Find Executable Files”

- ① `/usr/bin/grep`
- ② `/usr/bin/which`
- ③ `/bin/mkdir`
- ④ `/opt/gnome/bin/nautilus`
- ⑤ `/usr/X11R6/bin/OOo-writer`
- ⑥ `/usr/lib/jvm/jre/bin/ktab`
- ⑦ `/usr/X11R6/bin/fvwm2`

### Answers Exercise “Specify the Type of a Command”

`echo` – External command (`/bin/echo`)

`ls` – External command (`/bin/ls`) and alias (`/bin/ls $LS_OPTIONS`)

`which` – External command (`/usr/bin/which`) and alias (`_which`)

`_which` – Function

`test` – Shell built in and external command (`/usr/bin/test`)

### Answers Exercise “Create a tar Archive”

☞ `tar -cvf /tmp/bin.tar /bin`

☞ `cd /tmp`

☞ `tar -xvf /tmp/bin.tar bin/chown` (To see the correct path of the file inside the archive enter `tar -tvf bin.tar`)

☞ Original file:

```
1 geeko@da51:/tmp> ls -l /bin/chown
2 -rwxr-xr-x 1 root root 38332 2006-06-16 09:57 /bin/chown
```

Unpacked file:

```
1 geeko@da51:/tmp> ls -l /tmp/bin/chown
2 -rwxr-xr-x 1 geeko users 38332 2006-06-16 09:57 /tmp/bin/chown
```

### Answers Exercise “Compress a File – Part I”

After zipping the `tar` archive, you have to unzip the file before you can do the next compress rate. Use `gunzip bin.tar.gz`.

☞ `gzip -1 /tmp/bin.tar` → file size: 3507493 bytes

☞ `gzip -6 /tmp/bin.tar` → file size: 3263956 bytes

☞ `gzip -9 /tmp/bin.tar` → file size: 3251522 bytes

### Answers Exercise “Compress a File – Part II”

After zipping the `tar` archive, you have to unzip the file before you can do the next compress rate. Use `bunzip2 bin.tar.bz2`.

☞ `bzip2 -1 /tmp/bin.tar` → file size: 3153442 bytes

☞ `bzip2 -6 /tmp/bin.tar` → file size: 2980394 bytes

☞ `bzip2 -9 /tmp/bin.tar` → file size: 2960475 bytes

☞ Do the following

- ❶ Enter `cp /tmp/bin.tar.bz2 ~`
- ❷ Enter `mkdir ~/Training`
- ❸ Enter `cd ~/Training`
- ❹ Enter `tar -xjvf ~/bin.tar.bz2`

### Answers Exercise “Manage File Permissions – Part I”

Enter `chmod -R g-x,o-x ~/Training/`

### Answers Exercise “Manage File Permissions – Part II”

☞ Permission 777

**Owner** All rights

**Group** All rights

**Others** All rights

☞ Permission 755

**Owner** All rights

**Group** Read and execute

**Others** Read and execute

☞ Permission 600

**Owner** Read and write

**Group** No rights

**Others** No rights

☞ Permission 644

**Owner** Read and write

**Group** Read

**Others** Read

☞ Permission 2755

**Owner** All rights

**Group** Read and execute (SGID is set)

**Others** Read and execute

**Answers Exercise “Manage File Ownership”**

Enter `chgrp -R dialout ~/Training/`

**3.2 Understand bash Commands and Features**

**Learning target:** The students are able to use the shell effectively

**Answers Exercise “Use the bash History”**

Type `Ctrl+R` followed by the command you are looking for.

**Answers Exercise “Use bash Variables – Part I”**

By default:

```
1 geeko@da51:~> echo $PATH
2 /home/geeko/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/gnom
e/bin:/opt/kde3/bin:/usr/lib/jvm/jre/bin:/usr/lib/mit/bin:/usr/lib/mit/sbin
```

**Answers Exercise “Use bash Variables – Part II”**

```
1 geeko@da51:~> echo 'Variable $PATH is set to ' "$PATH"
2 Variable set to /home/geeko/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/us
r/games:/opt/gnome/bin:/opt/kde3/bin:/usr/lib/jvm/jre/bin:/usr/lib/mit/bin:/usr/
4 lib/mit/sbin
geeko@da51:~>
```



```
1 geeko@da51:~> echo 'Variable $PATH is set to ' "$PATH"
2 Variable is set to /home/geeko/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:
/usr/games:/opt/gnome/bin:/opt/kde3/bin:/usr/lib/jvm/jre/bin:/usr/lib/mit/bin:/u
4 sr/lib/mit/sbin
geeko@da51:~>
```



```
1 geeko@da51:~> echo "Variable \$PATH is set to $PATH"
2 Variable $PATH is set to /home/geeko/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:
/bin:/usr/games:/opt/gnome/bin:/opt/kde3/bin:/usr/lib/jvm/jre/bin:/usr/lib/mit/b
4 in:/usr/lib/mit/sbin
geeko@da51:~>
```

### Answers Exercise “Use bash Aliases”

Enter `alias spy="tar -tzvf"`. To store this alias for future sessions, you can add this command to the file `~/.alias`.

### Answers Exercise “Use Search Patterns to Expand Names”

- ❶ 1 file, enter `ls /bin/sa*`
- ❷ 3 files, enter `ls /bin/[xyz]*`
- ❸ 9 files, enter `ls /bin/*s`
- ❹ 11 files, enter `ls /bin/???`
- ❺ 2 files, enter `ls /bin/*[0-9]*`

## 3.3 Understand Advanced Command Execution

**Learning target:** The students understand piping, redirection and regular expressions.

### Answers Exercise “Use Piping and Redirection – Part I”

- ❶ Enter `ls / > ~/Redirection`
- ❷ Enter `echo $PATH >> ~/Redirection`
- ❸ Enter `cd /root 2>> ~/Redirection`

To see the content of Redirection, enter `less Redirection`.

### Answers Exercise “Use Piping and Redirection – Part II”

- ❶ Enter `ls -l | wc -l`
- ❷ The construction displays the number of files in a directory.

**Answers Exercise “Use Piping and Redirection – Part III”**

- ❶ `ls -l | wc -l → 0`
- ❷ `cd /root → 1`
- ❸ `less wall → 1`
- ❹ `echo $LOREM → 0`
- ❺ `echo $? → 0`

**Answers Exercise “Understand Regular Expressions – Part I”**

- ❶ `^.t → 8 hits`  
All lines with a “t” at the second position.
- ❷ `it\> → 16 hits`  
All lines with a word that ends with “it”.
- ❸ `m{2} → 6 hits`  
All lines with a double-“m”. Use `egrep`.
- ❹ `[^.,]$ → 20 hits`  
All lines without fullstop and comma at the end.

**Answers Exercise “Understand Regular Expressions – Part II”**

- ❶ 11 lines  
`egrep -n "[ABCDEFGHIJKLMNOPQRSTUVWXYZ]" loremipsum.txt`
- ❷ 6 lines  
`egrep -n "\<in\>|\<ad\>" loremipsum.txt`
- ❸ 12 lines  
`egrep -n "\<.{3}\>" loremipsum.txt`

**Answers Exercise “Manage Shell Processes”**

- ❶ Enter `acroread /usr/share/doc/manual/sled-gnome-user_en/sled-gnome-user_en.pdf`
- ❷ Press `Ctrl+Z`. Acrobat Reader does not response to any action.
- ❸ Enter `bg`. Acrobat Reader works as usual.

## 3.4 The vi Text Editor

**Learning target:** The students are able to edit files with vi.

**Answers Exercise “Move Cursor in vi – Part I”**

Key	Direction
<code>K</code>	Up
<code>J</code>	Down
<code>H</code>	Left
<code>L</code>	Right

**Answers Exercise “Move Cursor in vi – Part II”**

Just do it.

**Answers Exercise “Learn the Working Modes”**

Just do it.

**Answers Exercise “Enable Syntax Highlighting”**

Start vi → Enter `:e ~/.vimrc` → Press `I` → Enter `syntax on` → Press `Esc` → Enter `:wq`

To test syntax highlighting, enter `vi /usr/share/doc/release-notes/SUSE.Linux.Enterprise.Desktop.10/RELEASE-NOTES.en.html`.

**Answers Exercise “Create Macros”**

- ❶ Move the cursor to the very beginning of the file (upper left corner).
- ❷ Enter `Q`, `8`.
- ❸ Type `I`
- ❹ Type , ,
- ❺ Press `Esc`
- ❻ Press `H`, `H`
- ❼ Press `J`
- ❽ Press `Q`

To test the macro, type `6`, `@`, `8`.

**Answers Exercise “Create Abbreviations”**

Start vi → Press `:` → Enter `iabbrev shebang #!/bin/bash` and press `Enter`

To test syntax highlighting, switch to insert mode and enter `sheebang`.

## 3.5 The Basics of bash Programming

**Learning target:** The students are able to write simple shell scripts.

**Answers Exercise “Types of Variables”**

Operations for numbers:

- ❶ Addition
- ❷ Subtraction
- ❸ Multiplication
- ❹ Division

Operations for strings:

- ❶ Append characters
- ❷ Change case
- ❸ Select substrings
- ❹ Count characters/words

### Answers Exercise “Compute Integers”

For example:

```
1 #!/bin/bash
2
3 a=6
4 b=9
5 c=12
6 d=$(( $a * $a ))
7 e=$(( $b * $b ))
8 f=$(( $c * $c ))
9
10 echo "$a^2=$d"
11 echo "$b^2=$e"
12 echo "$c^2=$f"
```

Output:

```
1 6^2=36
2 9^2=81
3 12^2=144
```

### Answers Exercise “Compute Float Numbers”

bc can show the decimal places (e.g., for division) after entering `scale=number` in direct mode. To get decimal places in a shell script, enter `scale=number` in front of the operation, separated by semicolon.

For example:

```
1 geeko@da51:~> echo "scale=2; 9/2" | bc
2 4.50
```

Sample solution of the exercise:

```
1 #!/bin/bash
2
3 a=1.5
4 b=2.75
5 c=7.2
6 d=$(echo "$a*$a" | bc)
7 e=$(echo "$b*$b" | bc)
8 f=$(echo "$c*$c" | bc)
9
10 echo "$a^2=$d"
11 echo "$b^2=$e"
12 echo "$c^2=$f"
```

Output:

```
1 1.5^2=2.2
2 2.75^2=7.56
3 7.2^2=51.8
```

## Answers Exercise “Comparing Files, Strings and Numbers – Part I”

For verification:

```
1 geeko@da51:~> ls -l /var/log/messages
2 -rw-r----- 1 root root 16845 2006-10-12 06:52 /var/log/messages
```

For example:

```
1 #!/bin/bash
2
3 a=/var/log/messages
4
5 test -e $a
6 echo "-e $a :" $?
7
8 test -s $a
9 echo "-s $a :" $?
10
11 test -r $a
12 echo "-r $a :" $?
13
14 test -w $a
15 echo "-w $a :" $?
16
17 test -x $a
18 echo "-x $a :" $?"
```

Output:

```
1 -e /var/log/messages : 0
2 -s /var/log/messages : 0
3 -r /var/log/messages : 1
4 -w /var/log/messages : 1
5 -x /var/log/messages : 1
```

**Answers Exercise “Comparing Files, Strings and Numbers – Part II”**

For example:

```
1 #!/bin/bash
2
3 a=Linux
4 b=School
5
6 test $a == $b
7 echo "$a == $b: $?"
```

Output:

```
1 Linux == School: 1
```

**Answers Exercise “Comparing Files, Strings and Numbers – Part III”**

For example:

```
1 #!/bin/bash
2
3 a=13
4 b=11
5
6 test $a -gt $b
7 echo "$a -gt $b: $?"
```

Output:

```
1 13 -gt 11: 0
```

**Answers Exercise “Comparing Files, Strings and Numbers – Part IV”**

**Attention!** `bc`'s return value is different to the bash return values. “0” means the relation is false and “1” means the relation is true. See `man 1 bc`.

For example:

```
1 #!/bin/bash
2
3 a=13.5
4 b=22
5
6 c=$(echo "$a > $b" | bc)
7 echo "$a > $b: $c"
```

Output:

```
1 13.5 > 22: 0
```

## Answers Exercise “Comparing Files, Strings and Numbers – Part V”



**Attention!** Because of the different return values of `bc`, you have to use the NOT operator, too. Alternatively, you can test for the counterpart relation.

For example:

```
1 #!/bin/bash
2
3 a=13.5
4 b=22
5 c=15
6
7 d=$(echo "$a > $b" | bc)
8 echo "$a > $b: $d"
9
10 e=$(echo "$a < $c" | bc)
11 echo "$a < $c: $e"
12
13 f=$(echo "!( $a > $b || a < $c )" | bc)
14 echo "$a > $b || a < $c: $f"
```

Alternative:

```
1 #!/bin/bash
2
3 a=13.5
4 b=22
5 c=15
6
7 d=$(echo "$a <= $b" | bc)
8 echo "$a > $b: $d"
9
10 e=$(echo "$a >= $c" | bc)
11 echo "$a < $c: $e"
12
13 f=$(echo "!( $a > $b || a < $c )" | bc)
14 echo "$a > $b || a < $c: $f"
```

Output:

```
1 13.5 > 22: 1
2 13.5 < 15: 0
3 13.5 > 22 || a < 15: 0
```

**Answers Exercise “Read from Input”**

For example:

```

1 #!/bin/bash
2
3 read a b
4
5 test $a -gt $b
6 echo "$a -gt $b: $?"

```

Output:

```

1 geeko@da51:~> sh script.sh
2 8 4
3 8 -gt 4: 0
4 geeko@da51:~> sh script.sh
5 7 7
6 7 -gt 7: 1

```

**Answers Exercise “Examine Cases – Part I”**

For example:

```

1 #!/bin/bash
2
3 read a b
4
5 if test $a -gt $b; then
6     echo "$a is greater than $b."
7 elif test $a -eq $b; then
8     echo "$a is equal to $b."
9 else
10    echo "$a is smaller than $b."
11 fi

```

Output:

```

1 geeko@da51:~> sh script.sh
2 3 7
3 3 is smaller than 7.
4 geeko@da51:~> sh script.sh
5 8 2
6 8 is greater than 2.
7 geeko@da51:~> sh script.sh
8 6 6
9 6 is equal to 6.

```

## Answers Exercise “Examine Cases – Part II”

For example:

```
1 #!/ bin / bash
2
3 read a
4
5 case $a in
6 0)
7     echo "Zero"
8     ;;
9 1)
10    echo "One"
11    ;;
12 2)
13    echo "Two"
14    ;;
15 3)
16    echo "Three"
17    ;;
18 4)
19    echo "Four"
20    ;;
21 5)
22    echo "Five"
23    ;;
24 6)
25    echo "Six"
26    ;;
27 7)
28    echo "Seven"
29    ;;
30 8)
31    echo "Eight"
32    ;;
33 9)
34    echo "Nine"
35    ;;
36 *)
37    echo "Wrong input."
38 esac
```

Output:

```
1 geeko@da51:~> sh script.sh
2 3
3 Three
4 geeko@da51:~> sh script.sh
5 9
6 Nine
7 geeko@da51:~> sh script.sh
8 22
9 Wrong input.
```

## Answers Exercise "Use Loops – Part I"

For example:

```
1 #!/bin/bash
2
3 read a
4
5 b=1
6
7 while test $b -le $a; do
8     case $b in
9         1)
10            echo "One"
11            ;;
12         2)
13            echo "Two"
14            ;;
15         3)
16            echo "Three"
17            ;;
18         4)
19            echo "Four"
20            ;;
21         5)
22            echo "Five"
23            ;;
24         6)
25            echo "Six"
26            ;;
27         7)
28            echo "Seven"
29            ;;
30         8)
31            echo "Eight"
32            ;;
33         9)
34            echo "Nine"
35            ;;
36         esac
37         b=$(expr $b + 1)
38     done
```

Output:

```
1 geeko@da51:~> sh script.sh
2 3
3 One
4 Two
5 Three
6 geeko@da51:~> sh script.sh
7 7
8 One
9 Two
10 Three
11 Four
12 Five
13 Six
14 Seven
```

## Answers Exercise “Use Loops – Part II”

Only line 7 changes. For example:

```
1 #!/bin/bash
2
3 read a
4
5 b=1
6
7 until test $b -gt $a; do
8     case $b in
9         1)
10            echo "One"
11            ;;
12         2)
13            echo "Two"
14            ;;
15         3)
16            echo "Three"
17            ;;
18         4)
19            echo "Four"
20            ;;
21         5)
22            echo "Five"
23            ;;
24         6)
25            echo "Six"
26            ;;
27         7)
28            echo "Seven"
29            ;;
30         8)
31            echo "Eight"
32            ;;
33         9)
34            echo "Nine"
35            ;;
36         esac
37         b=$(expr $b + 1)
38     done
```

## Answers Exercise “Use Loops – Part III”

For example:

```
1 #!/bin/bash
2
3 read a
4
5 for ((b=1; b<=$a; b++)); do
6     case $b in
7         1)
8            echo "One"
9            ;;
10        2)
11            echo "Two"
12            ;;
13        *)
14            echo "..."
15            ;;
16        esac
17    done
```

```

12     ;;
13     3)
14     echo "Three"
15     ;;
16     4)
17     echo "Four"
18     ;;
19     5)
20     echo "Five"
21     ;;
22     6)
23     echo "Six"
24     ;;
25     7)
26     echo "Seven"
27     ;;
28     8)
29     echo "Eight"
30     ;;
31     9)
32     echo "Nine"
33     ;;
34     esac
done

```

### Answers Exercise “Use Program Options and Parameters”

- ❶ Option `-a` is found first and the lines 5 and 6 are executed. Option `-f` is found next and the lines 11 and 12 are executed. `-f` expects another argument (line 11), so `/home/geeko` is used here.
- ❷ Option `-f` is found first and the lines 11 and 12 are executed. `-f` expects another argument (line 11), so `/home/geeko` is used here. Option `-a` is found next and the lines 5 and 6 are executed.
- ❸ Option `-b` is found first and the lines 8 and 9 are executed. Option `/home/geeko` is found next. But this is not a valid option, so the lines 14 and 15 are executed.
- ❹ Option `/home/geeko` is found first. This is not a valid option, so the lines 14 and 15 are executed.

### Answers Exercise “Create Shell Functions”

For example:

```

1 #!/bin/bash
2
3 validate() {
4     case $1 in
5         0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)

```

```

6         return 0
          ;;
8     finish)
          return 1
          ;;
10        *)
          return 2
          ;;
12        esac
14    }
16
17    # b stores the sum of entered numbers
18    b=0
19
20    # c stores the number of entered numbers
21    c=0
22
23    while true; do
24        echo "Enter a number or \"finish\""
25        read a
26        validate $a
27        case $? in
28            0)
29            b=$(expr $b + $a)
30            c=$(expr $c + 1)
31            d=$(echo "scale=3; $b/$c" | bc)
32            echo "Average: $b / $c = $d"
33            ;;
34            1)
35            echo "Finished!"
36            break
37            ;;
38            2)
39            echo "Wrong input! Enter a number or \"finished\""
40            echo "Average: $b / $c = $d"
41            ;;
42        esac
43    done

```

The variable `?` (line 27) stores the return value of the function (see page 89 of the student manual).

`scale=3;` in line 31 enables three decimal places (see section 3.5 on page 19).

Output:

```

1  geeko@da51:~> sh script.sh
2  Enter a number or "finish"
3  5
4  Average: 5 / 1 = 5.000
5  Enter a number or "finish"
6  3
7  Average: 8 / 2 = 4.000
8  Enter a number or "finish"
9  3
10 Average: 11 / 3 = 3.666
11 Enter a number or "finish"
12 abc
13 Wrong input! Enter a number or "finished"
14 Average: 11 / 3 = 3.666
15 Enter a number or "finish"
16 finish

```

Finished!

## 3.6 Understand the sed Stream Editor

**Learning target:** The students are able to manipulate a text file using sed.

### Answers Exercise “Understand sed – Part I”

```
Enter sed -n '10,15p' loremipsum.txt
```

### Answers Exercise “Understand sed – Part II”

```
Enter sed -n '1,10d;w shortlorem.txt' loremipsum.txt
```

### Answers Exercise “Understand sed – Part III”

```
Enter sed -n 's/quis/quod;/w quodlorem.txt' loremipsum.txt
```

### Answers Exercise “Understand sed – Part IV”

```
1 s/dolor/XXXXX/;  
2 y/o/O/;  
   /Duis/d;  
4 w scriptlorem.txt
```

```
Enter sed -nf script_filename loremipsum.txt
```

# 4 Linux Services for Developers

## 4.1 The Version Control System Subversion

**Learning target:** The students understand the benefits of a version control system and are able to use Subversion.

### Answers Exercise “Install Subversion”

Install Subversion as it is described in the student manual.

### Answers Exercise “Create a Subversion Repository”

- ❶ Enter `su -`.
- ❷ Enter the root password.
- ❸ Enter `mkdir /srv/repository`
- ❹ Enter `svnadmin create /srv/repository`
- ❺ Enter `chgrp -R users /srv/repository/`
- ❻ Enter `chmod -R 775 /srv/repository/`
- ❼ Enter `exit`

### Answers Exercise “Create a Subversion Directory”

- ❶ Enter `svn mkdir file:///srv/repository/project1`
- ❷ Press
- ❸ Enter `Project created.`

- ④ Press `Esc`
- ⑤ Enter `:wq`

### Answers Exercise “Import Files into Subversion”

- ① Enter `mkdir ~/my_project`
- ② Copy the file `grep-demo.tar.gz` from the book CD into your directory.
- ③ Enter `tar -xzvf grep-demo.tar.gz -C ~/my_project`
- ④ Enter  
`svn import my_project file:///srv/repository/project1/`
- ⑤ Press `I`
- ⑥ Enter Initial import.
- ⑦ Press `Esc`
- ⑧ Enter `:wq`
- ⑨ Enter `rm -r ~/my_project`

### Answers Exercise “Create a Local Copy”

Enter `svn checkout file:///srv/repository/project1 project1`

### Answers Exercise “Add a File to the Repository”

1. Enter `su - username`
2. Enter  
`svn checkout file:///srv/repository/project1 project1`
3. Enter `cd project1/`
4. For example, enter `vi help.txt`
5. Press `I`
6. Enter some text (or use copy and paste).

7. Press `Esc`
8. Enter `:wq`
9. Enter `svn add help.txt`
10. Enter `svn commit`
11. Press `I`
12. Enter `Added file help.txt.`
13. Press `Esc`
14. Enter `:wq`
15. Enter `exit`
16. Enter `svn update project1/`

### Answers Exercise “Remove a File from the Repository”

- ❶ Enter `svn remove help.txt`
- ❷ Press `I`
- ❸ Enter `Removed file: help.txt.`
- ❹ Press `Esc`
- ❺ Enter `:wq`
- ❻ Enter `su - username`
- ❼ Enter `svn update project1/`
- ❽ Enter `exit`

### Answers Exercise “Rename a File”

- ❶ Enter  
`svn rename dirgrep/loremipsum2.txt dirgrep/README`
- ❷ Press `I`
- ❸ Enter `Renamed file: loremipsum2.txt -> README.`
- ❹ Press `Esc`

- 5 Enter :wq
- 6 Enter su - *username*
- 7 Enter svn update project1/
- 8 Enter exit

### Answers Exercise “Solve Conflicts”

To create a conflict do the following, for example:

1. Enter su - *username*
2. Enter vi project1/loremipsum.txt
3. Enter dd
4. Enter :wq
5. Enter svn commit
6. Press  I
7. Enter Deleted 1st line from loremipsum.txt.
8. Enter :wq
9. Enter exit
10. Enter vi loremipsum.txt
11. Press  I
12. Enter New Text
13. Press  Esc
14. Enter :wq
15. Enter svn commit
16. Press  I
17. Enter Changed 1st line in loremipsum.txt.
18. Press  Esc
19. Enter :wq

An error message appears. This message tells that there is a problem with the file `loremipsum.txt`.

1. Enter `svn diff loremipsum.txt`
2. Enter `svn update loremipsum.txt`
3. Enter `vi loremipsum.txt`
4. Enter `dd`
5. Press `J`
6. Enter `2dd`
7. Enter `:wq`
8. Enter `svn resolved loremipsum.txt`
9. Enter `svn commit`
10. Press `I`
11. Enter `Solved conflict in first line.`
12. Press `Esc`
13. Enter `:wq`

### Answers Exercise “View Subversion Logs”

Being in the `~/project1` directory, enter `svn log --verbose`.

### Answers Exercise “Set Up a Subversion Server”

- ① Enter `svnserve -d`
- ② Enter  
`svn checkout svn://IP-address/srv/repository/project1`  
`project2`  
or  
`svn checkout svn://IP-address/srv/repository/project1`  
`project2`

## 4.2 The Web Server Apache

**Learning target:** The students are able to install and setup Apache.

## Answers Exercise “Install Apache”

Enter `rpm -ihv /media/mountpoint/packages/*.rpm`

Depending on your installation, some dependencies failed when trying to install the Apache RPMs. The missing packages (e.g., libapr) are available on the SUSE Linux Enterprise Desktop DVD.

## Answers Exercise “Start Apache”

- ❶ Switch to user root by entering `su -`
- ❷ Enter `rcapache2 start`
- ❸ Start Firefox and enter `http://localhost` in the address bar.
- ❹ Enter `http://IP-address` in the address bar to test the webserver of a student.

If the students already know HTML let them create an HTML page. Alternatively, you can provide some HTML files. Copy the HTML files into the directory `/srv/www/htdocs/`.

## Answers Exercise “Limit Access Using IP Addresses”

- ❶ As user root, enter `vi /etc/apache2/default-server.conf`
- ❷ The configuration of the directory `/srv/www/htdocs/` is at the beginning of the file.
- ❸ Next to the line

```
1 Allow from all
```

add

```
1 Deny from 10.0.0.15
```

Replace 10.0.0.15 by the IP address of the student you want to block.

- ❹ Leave vi by entering `:wq`
- ❺ Enter `rcapache2 restart`

**Answers Exercise “Limit Access Using User Authentication”**

1. As user root, enter `vi /etc/apache2/default-server.conf`
2. The configuration of the directory `/srv/www/htdocs/` is at the beginning of the file.
3. Remove the following line

```
1 Deny from 10.0.0.15
```

4. Add the following lines:

```
1 AuthType Basic
2 AuthName " Restricted Files "
  AuthUserFile /etc/apache2/htpasswd
4 Require user tux
```

Replace `tux` by the login of your second user account.

5. Leave vi by entering `:wq`
6. Enter `htpasswd2 -c /etc/apache2/htpasswd tux` and replace `tux` by the login of your second user account.
7. Enter a password (e. g., `novell`) twice.
8. Enter `rcapache2 restart`
9. Test whether you can access the webserver with Firefox.
10. Enter the login of your first user account and its password. → The access will be denied.
11. Enter the login of your second user account and the password you entered with `htpasswd2`. → The access will be allowed.