N.V. Savelyeva, V.V. Tryhuk

Linux OS Administration.
Lab Assignments

CONTENTS

PREFACE

The practicum «Linux OS Administration. Lab Assignments» is intended for studying the subject «Linux OS Administration» that is taught in English language to third-year students of the specialty «Applied Mathematics». This practicum follows the purpose of the mentioned subject and corresponds to its syllabus (see the detailed syllabus of the discipline in the table below). This syllabus contains tasks for holding classes in computer laboratories, appendices include questions for pre-test and final control test, Linux command-line basics such as console commands, service profiles, the meaning of common configuration files and important directories. The practical assignments are more complicated in comparison to lab assignments. Besides, practical assignments require building the test environment of at least two computers (or virtual machines). Thus, according to the program of the discipline «Linux OS Administration» it is recommended for students to work in a team of two people when performing practical tasks. Lab assignments are easier and require only one Linux-system, students should work independently in this case.

Significant place in the course is devoted to laboratory and practical training, which is dealing with specific practical problems of administration of local computer networks. All the assignments are built in order to provide interrelated deep understanding of the basics of the administration of local Linux-based computer networks and servers which are running Linux operating system. As a result such learning objectives of the discipline can be achieved: getting practical skills on sustainable administration of local computer networks, servers and workstations, as well as programming skills in the bash scripting language; deepening the knowledge and skills obtained by students previously from studying related disciplines (e.g. «Computer Networks», «Operating Systems», etc.), which reveals the intra-disciplinary and cross-curricular connections.

The knowledge of:

- stages of the Linux booting process,
- the architecture of the file system in Linux operating systems,
- Linux console commands,
- the purpose and principles of the basic network services

must be the learning outcomes of the theory part of the discipline «Linux OS Administration». And, in turn, satisfying the requirements of the syllabus, the practicum «Linux OS Administration. Lab Assignments» will help a teacher to make a student to be able to:

- provide remote access to Linux-based systems and shared computer resources (NFS, Samba, FTP, SSH);
- configure network firewall (the iptables service);

- configure network services DNS, DHCP, Apache, SQUID, Sendmail, NIS, NTP, CUPS, Kerberos;
- monitor and manage processes, services and tasks in Linux OS;
- automate administrative tasks and optimize the work of system administrators by creating programs in the bash scripting language.

Table – The syllabus of the subject «Linux OS Administration»

Section/topic number	**Section/topic title**	**The number of classroom hours**			
		Total	Lectures	Laboratory classes	Practical classes
1	**Linux as a workstation**	**34**	**16**	**16**	**2**
1.1	Introduction to Linux operating system (OS), its evolution and diversity	**2**	1	1	-
1.2	The installation and booting processes of Linux OS. The concept of run-level. Start-up scripts. Setting up physical devices.	**10**	6	4	-
1.3	File systems for Linux OS. The virtual file system in Linux. File types and file attributes. Basic commands to work with files in Linux. Analysis, and changing settings in the configuration files for general use. Mounting.	**4**	2	2	-
1.4	Creation, deletion and modification of user and group accounts. The distribution of privileges between users and groups.	**4**	2	2	-
1.5	The text editor «Vim», its interface and basic commands. Using «Vim» to edit text files and create C programs.	**1,5**	0,5	1	-
1.6	Working with the bash shell. Programming in the bash scripting language: variables, looping and branching.	**2,5**	0,5	-	2
1.7	Control over the Linux OS. Monitoring processes, services and jobs. The Linux scheduler «Cron». Logging of system events. Conducting auditing in Linux OS.	**10**	4	6	-
2	**Linux as a network server**	**32**	**8**	**-**	**24**
2.1	Implementation of the firewall in Linux OS. The principle of operation of the service «iptables» and its settings.	**2**	2	-	2
2.2	Providing remote access to resources. Setting up of network services NFS, Samba, FTP, SSH, Apache.	**12**	2	-	10
2.3	Setting up of network services DNS, DHCP, SQUID, Sendmail, NIS, CUPS, NTP, Kerberos.	**18**	2	-	12
2.4	Duties and responsibilities of system administrators. Major recommendations to competent administration and improving security.	**2**	2	-	-
Total		**66**	**24**	**16**	**26**

1 LAB ASSIGNMENTS

Instruments: Red Hat Enterprise Linux 6 (RHEL 6 or simply RHEL) – this may be either virtual or real machine(s).

1.1 Linux installation process

Purpose: studying Linux installing process in details and getting acquainted with Linux OS (Operating System).

Objectives:

1. Configure important parameters during the installation process (e.g. mount points, passwords, software packages, etc.).
2. Learn how to use a console text editor (in particular, «Vim»).

Preparation: Make the system to be booted from a CD/DVD first.

Implementation

1. Boot from a DVD and initiate installing. Specify the following parameters:
   a. Create a swap partition of the size 1024 MB, other space give for the root mount point.

      Question A. *Can a user make his own partition? How?*

   b. Do not create a boot loader password (but remember that it can be created during the installation process).

      Question B. *What boot loader is implemented in RHEL?*

   c. Change the default installation of RHEL as a basic server to «Desktop» and customize a software selection (include C or C++ compilers).
2. Following the guide complete the installation process.
3. Run the text editor called «Vim».

   Question C. *What other text editors for Linux do you know?*
4. Practice in creating, editing and saving text files, then fill the table below:

Table 1.1 – Vim Commands

No.	***Command***	***Description***
1.		Switch the command mode to the insert mode (and back)
2.		Save the document
3.		Save the document and quit
4.		Quit without saving
5.		Copy a line (*n* lines) and paste
6.		Delete a line (*n* lines)
7.		Search for a pattern within the document

5. Prepare a final lab report containing the following information:
   - answers to the questions typed in *italic* (questions A-C);
   - the filled table 1.1;
   - your own conclusions.

1.2 Boot experiments

Purpose: studying Linux boot process in details.

Objectives:

1. Learn how to work in different run-levels, create and manage start-up scripts, reset the root password.
2. Study the syntax of common configuration files.

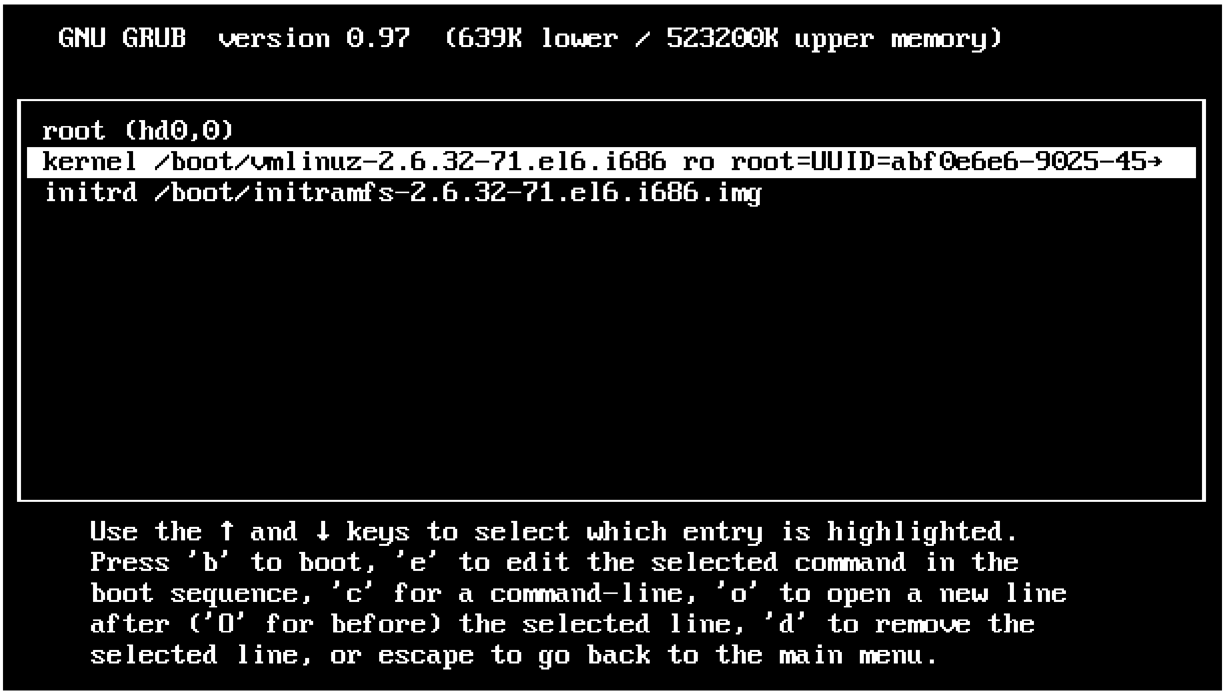
Implementation

Task 1. Reset the root password

Having a physical access to the system it is possible to reset a root password [e.g. if it is forgotten]. This can get done on the GRUB stage of loading Linux. The aim of the task is to reset the password of the root user account.

1. Boot the system and come to the GRUB menu (by pressing any key when you see the invitation screen "*Press any key to enter the menu*"). When you see the menu, press "**`e`**" to edit commands before booting of the highlighted OS.

2. In the next window (see the picture below) highlight the menu item corresponding to the kernel (*vmlinuz*). For this use the arrow keys and then press "**`e`**".

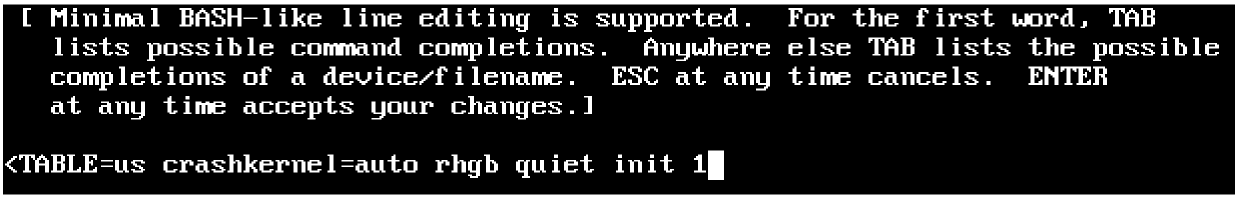


```
   GNU GRUB  version 0.97  (639K lower / 523200K upper memory)

 root (hd0,0)
 kernel /boot/vmlinuz-2.6.32-71.el6.i686 ro root=UUID=abf0e6e6-9025-45→
 initrd /boot/initramfs-2.6.32-71.el6.i686.img

     Use the ↑ and ↓ keys to select which entry is highlighted.
     Press 'b' to boot, 'e' to edit the selected command in the
     boot sequence, 'c' for a command-line, 'o' to open a new line
     after ('O' for before) the selected line, 'd' to remove the
     selected line, or escape to go back to the main menu.
```

3. In the next window after the word **`quiet`** type the command **`single`** or **`init 1`**.

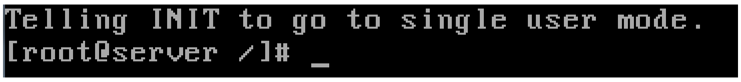


```
 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename.  ESC at any time cancels.  ENTER
   at any time accepts your changes.]

<TABLE=us crashkernel=auto rhgb quiet init 1
```

After this press the "Enter" key (it will take you to the previous menu).

4. Press “`b`” to boot the system with the new argument. The system will boot into the single user mode and you will see bash prompt like below:



```
Telling INIT to go to single user mode.
[root@server /]# _
```

5. Now reset the `root`’s password. ***Question A.*** *What is the command to be used*?
6. Reboot the system into multiuser console mode. ***Question B.*** *What is the command to be used*?

Task 2. Change start-up parameters

In most cases system administrators might have a wish to change some start-up parameters like setting up the message before login prompt, the message of the day to be displayed after a user successfully logs in, run own scripts while the system is booting. The aim of this task is to:

- *set up messages to be displayed before and after a user logs in;*
- *create a script and make it to be executed automatically at start-up.*

1. Set up the message of the day after every successful log in.
   a. In the file `/etc/motd` type the message “*Don’t make administrators angry by keeping them hungry!*” without quotes.
   Question C. *How do you think why is the file called “motd”*?
   b. Reboot the system into multiuser console mode.

2. Set up the message to be displayed before the prompt to log in:
   a. In the file `/etc/issue` add the following two lines:

```
User No. \n, you are welcome to log in to \n.
Week-end is coming!
```

   b. In the file `/etc/rc.local` before the line `touch /var/lock/subsys/local` add the following command:

```
echo “Now it is $(/bin/date)”>>/etc/issue
```

   c. Reboot the system. ***Question D.*** *What happened*? *Why*?
   d. Correct displaying unwanted (repeated) messages.
   Question E. *What will be your actions*?

3. In the directory `/etc/init.d/` create your own start-up script called `mystart`.
   a. Create a file called `mystart` and make it executable.
   b. Open the file `mystart` with Vim and type the following lines:

```
#!/bin/bash
echo “NETWORKING=YES” > /etc/sysconfig/network
echo “HOSTNAME=server.rh6” >> /etc/sysconfig/network
echo “GATEWAY=192.168.1.1” >> /etc/sysconfig/network
ifconfig eth0 192.168.1.15 netmask 255.255.255.0 up
echo THE SCRIPT myscript HAS DONE ITS ASSIGNMENTS
sleep 10
```

Question F. *Explain each line of the script using comments.*

c. Create a soft link to the script **mystart**:

```
ln -s /etc/init.d/mystart /etc/rc.d/rc3.d/S121mystart
```

d. Explain each line in the file **/etc/sysconfig/network-scripts/ifcfg-eth0**, change the ONBOOT parameter to YES and append a line like:

```
DNS1=82.209.200.16
```

e. Create another bash script and put it anywhere out of **/etc/init.d/** directory. Make a soft link to it (**/etc/rc.d/rc3.d/S…**). Check if it works: reboot the system. ***Question G.*** *Did the second script run*? *Why*?

Task 3. Change command-line prompt messages

The aim of this task is to create aliases and change the default style of the bash prompt for different users.

1. Change the **root**'s prompt style in the script **~/.bash_profile** (interactive, login shell) as follows:

```
PS1='R(\u@\h \w|\d \t)$ '
export PS1
```

Question H. *Where is root's home directory located*?

2. Change the **nato**'s prompt style in the script **~/.bash_profile** (which is responsible for the interactive and login shells) as follows:

```
PS1='N(\u@\h \w|\d \t)$ '
```

Question I. *Where is nato's home directory located*?

3. Change the prompt styles for users **root** and **nato** – for that in the script **~/.bashrc** (which is responsible for the interactive shell only) set the following aliases to several commands as shown below:

```
alias c="clear"
alias cp="cp -i"
alias rm="rm -i"
alias ls="ls -l"
PS1='R[\u@\h \w|\d \t]$ '
```

(for **root**)

```
(and PS1='N[\u@\h \w|\d \t]$ '
```

(for **nato**)

4. Create a new user account **john** and explore the behavior of login and interactive shells for three different users: try to open some new tabs in the terminal window, type commands, switch users, then reopen the terminal window.

5. Fill the table below and explain the differences.

Table 1.2 – Users and shells

		What changes have been applied? (**~/.bash_profile**, or **~/.bashrc**, or no changes applied) Note: write full paths		
		root	**nato**	**john**
1.	(interactive, login shell or tty)			
2.	(interactive shell)			

5. Being in the **tty1** (as **root**) type a command **shutdown -r +2 Alarma!!!**, after that switch to the **tty2** and wait for some time.
Question J. *What is tty and how many ttys are available*?

6. Prepare a final lab report containing the following information:
    - answers to the questions typed in *italic* (questions A-J);
    - the filled table 1.2;
    - the history of users **root**, **nato** and **john**;
    - your own conclusions.

1.3 Files, users, groups

Purpose: get practical experience of user and group accounts management.

Objectives:
1. Create, delete, modify users and groups using CLI and GUI.
2. Study the use of file permissions and other file attributes.

Implementation

1. Using console commands perform the following actions:
    a. Create users **bender** and **flexo** with passwords *futurama* and *iamarobot* correspondingly.
    b. Check whether new users present in the system.
    Question A. *How can it get checked? Describe at least two-three ways.*
    c. Log in as **bender** and change the password to *ilovelinux*.
    d. As **root** change **bender**'s password back to *futurama*.
    e. Create a group account **mafia** and make users **bender** and **flexo** to be its members.

2. Using GUI create user accounts called **bond** (full name: *James Bond*) and **hp** (full name: *Harry Potter*) with passwords *007007* and *hphphp* correspondingly.

3. Create a group account **friends** and add these users to the group **friends**.

4. Give full permissions to the directory **/tmp** and take the sticky bit off **/tmp**.
Question B. *What is the use of the sticky bit?*
Question C. *How can we take the sticky bit off the directory*?
Question D. *How can we check whether the file or directory has a sticky bit?*

5. To practice with management of files and directories, user and group accounts, implement the following steps:
    a. Login as **hp** and create a file **averyusefulfile** in **hp**'s home directory.
    b. Acting as **hp** try to copy **averyusefulfile** to the **bond**'s home directory and then place it into **/tmp** with write and read permissions for all users.
    c. Using another **tty** log in as **bond** and append some information to the **averyusefulfile** located in **/tmp**.
    d. Using one more **tty** log in as **flexo** and try to delete the **averyusefulfile** located in **/tmp**.

e. Acting as **hp** make sure the file **averyusefulfile** is absent in **/tmp.**
f. Using a separate **tty** log in as **root** and from **hp**'s home directory copy the old backup of the **averyusefulfile** with read/write permissions again to **/tmp** and give the sticky bit to the **/tmp** directory.
g. Acting as **bond** append some information to the **/tmp/averyusefulfile** again; acting as **flexo** try to delete it; acting as **bender** (from a different **tty**) append some unwanted information to the **/tmp/averyusefulfile**.
h. Acting as **hp** have a look at the modified file, then remove unwanted information from the **/tmp/averyusefulfile** and change the owner group of this file (set the group **friends** to own this file). Deny the file to be modified by other users.
i. Acting as **root** remove **/tmp/averyusefulfile**, remove user accounts **bond**, **hp**, **bender**, **flexo** as well as group accounts **mafia** and **friends**.

6. Prepare a final lab report containing the following information:
- answers to the questions typed in *italic* (questions A-D);
- the history of users **root** and **bender**, **flexo**, **hp**, **bond**;
- your own conclusions;
- taking the substeps of the step 5 as a base, compose a fairy tail how the sticky bit appeared in Linux systems (tell this story when submitting your final lab report).

1.4 Mounting

Purpose: Get the knowledge how to mount external drives to the system.
Objectives: 1. Install some software packages from the mounted DVD.
2. Mount USB pen drives with Cyrillic support.

Implementation

Task 1. CD/DVD Mounting

1. Boot the system with graphical support and login as **nato**.

2. Insert (or imitate inserting if running a virtual machine) a CD or DVD containing software packages for RHEL 6 (if it is automounted by the system then unmount it manually).

3. Install a package which is not present in the system (check this before) – for example, a file manager Midnight Commander (the corresponding package starts with **mc**) and run it. ***Question A.*** *What commands are to be executed*?

4. Install the packages **rhythmbox** and **totem**. Launch installed software from a command-line. ***Question B.*** *What are these packages for*?

5. Acting as **root** create some mount points and make at least one to be mounted automatically at start-up. ***Question C.*** *How can this get done*?

Task 2. USB Drive Mounting

Preparation. It is supposed that:

- the system is booted with graphical support;
- the USB drive contains at least a directory `GreatSongs` with 5-10 mp3-files with the names `*Remix*` or `*remix*` (extensions may be either `mp3` or `MP3`).

1. Insert a USB pendrive containing some mp3-files (if it is mounted automatically by the system then unmount it manually).

2. Mount the USB pendrive and:
   a. Show the *full* content of it;
   b. Count total and free space of the mounted drive;
   c. If files stored on the mounted drive contain Cyrillic symbols then remount it with the appropriate parameters of file system type and code page.

3. Acting as `nato`, in the home directory create two playlists named:
   a. `Playlist1` of all mp3 files stored on the USB pendrive;
   b. `Playlist2` of mp3-files stored in the directory `GreatSongs` and satisfying the masks `*Remix*` and `*remix*`.
   Question D. *What will be a single command for this purpose*?
   c. Repeat previous command with the additional property to copy all found documents to `/home/nato/mp3`.

4. Prepare a final lab report containing the following information:
   - answers to the questions typed in *italic* (questions A-D);
   - the history of the users `root` and `nato`;
   - your own conclusions.

1.5 Watching processes, services, jobs

Purpose: studying basic administrative functions and getting practical experience of monitoring Linux-based systems.

Objective: learn console commands that monitor running processes, services, jobs, CPU and memory usage.

Implementation

Task 1. Process Control

1. Display the list of all running processes.

Question A. *What are the names of processes with the PIDs 0 and 1*?

2. Display the full list of all running processes of a user `nato`.

3. Execute the `top` command and:
   a. Change the sequence of the fields;
   b. Add 2-3 fields more to be displayed;
   c. Remove 2-3 fields from the output table.

4. Using `watch` command make a simple analog of `top`.
Question B. *What command will you use to watch?*

5. Do the following:
   a. Create a simple web-page and try to access the local web-site;
   b. Stop the *service* of Apache and try to access the local web-site;
   c. Run Apache again and make sure the web-page is displayed in a browser;
   d. Stop the running Apache process(es) and try to access the local web-site. ***Question C.*** *Describe at least two approaches to stop Apache process(es).*

6. Display the statistics of memory usage. ***Question D.*** *What is the amount of virtual memory available at the current machine?*

7. Display the priorities of running processes. Decrease the priority of any user process and make sure the changes are applied.

Task 2. Jobs Control

1. Create some jobs and change their behavior:
   a. Run in the background a bash-script containing an infinity loop;
   b. Run in the foreground a compiled C program containing an infinity loop;
   c. Run in the foreground `vmstat 2`;
   d. Run in the background any «long» job up to your choice;
   e. Take several times these jobs to background and back to the foreground.

2. Terminate all jobs.
Question E. *Describe at least two approaches to get this done*.

3. Prepare a final lab report containing the following information:
   - answers to the questions typed in *italic* (questions A-E);
   - the history of users `nato` and `root`;
   - your own conclusions.

1.6 Cron – the Linux scheduler

Purpose: make regular tasks scheduled to be executed automatically with the help of the Cron service.

***Objectives*:** 1. Schedule jobs (commands and shell scripts) to run periodically at certain times and dates.
2. Allow and deny different users to use jobs scheduled by Cron.

Implementation

Task 1. Schedule jobs

1. Add the following jobs to the crontab file (here `cmd` stands for some abstract command):

```
0,30 8-17 * * 1-5 cmd
0 12 1,15 * 5 cmd
```

```
17 3 * * 1 cmd
0 15 * * 5 echo "Time for staff meeting" | write $LOGNAME >/dev/null 2>&1
0 15 * * 5 write $LOGNAME >/dev/null 2>&1 %Time for the%staff meeting
1 0 * * * echo -n "" > /var/spool/mail/root
*/5 * * * * /home/user/test.py
0 13 * * * notify-send --urgency critical --expire-time=10000 -i typing-
monitor -h int:x:500 -h int:y:500 "Lunch time"
```

Note: to use the **`notify-send`** command you may be prompted for installing the package **`libnotify-bin`**.

2. In **`nato`**'s home directory create a script **`cron.sh`** containing the command:

```
notify-send "Notice:" "Running jobs reduces system performance"
```

and add the following line to the crontab file:

```
* * * * * DISPLAY=:0.0 /home/nato/cron.sh
```

Note: to debug the work of Cron jobs you may change the system date by the command below:

```
date MMDDhhmmCCYY.ss
```

3. Assume the system should get turned off every day at 5.00 P.M. Automate this job.

4. Make the following tasks to be executed every hour:
   a. **`cuckoo`**, where 'cuckoo' is an alias for the command which displays the concatenated string "*Dear user, now it is* " with the current date and time.
   Question A. *What is an alias and how can it get created*?
   b. **`cuckoo.sh`** which displays the word "`Cuckoo!`" **`x`** times, where **`x`** is equal to the number of hours in A.M./P.M. mode (e.g. if it is 15.00 or 3.00 the script must display "`Cuckoo!`" three times (of course, without quotes).

5. Improve your Cockoo-effect:
   a. Make "`Cuckoo!`" notifications mentioned above to appear in a pop up window with the **`rhytmbox`** logo icon.
   Note: icons can be taken from **`/usr/share/icons/hicolor/48x48/apps/`**.
   b. **Optional*: attach a sound to the above action.

6. Every day in the midnight let a backup file (titled **`cron_yyyy_mm_dd.bak`**, where **`yyyy`** is the current year, **`mm`** is the current month, **`dd`** is the current day) of the main Cron's configuration file get created. To solve this task create a bash-script **`backup.sh`** (or a C/C++ program) which will also provide the following features:
   a. If the file exists it should be updated;
   b. Backup files should be stored in **`root`**'s home directory.
   c. Don't store more than 7 backup files.
   Question B. *What permissions are to be set to the file backup.sh (or to the compiled executable of the created C/C++ source code)*?

Task 2. Cron permissions and restrictions

Preparation. Create user accounts **holmes** and **watson**.

1. Allow users **root** and **holmes** only to use Cron jobs.
Question C. *What Cron's configuration files are to be modified and how*?

2. Deny the only user called **watson** to use Cron jobs.
Question D. *What Cron's configuration files are to be modified and how*?

3. Prepare a final lab report containing the following information:
- explanations of every job listed in the first point of the task 1;
- answers to the questions typed in *italic* (questions A-D);
- listings of all Cron's configuration files you have edited;
- the listing of the file **backup.sh** (or the source code of a C/C++ program);
- your own conclusions.

1.7 Audit

Purpose: learn how system events are being registered.
Objectives: 1. Watch different kinds of events happening to watched files.
2 **Optional*. Use graphical capabilities of Linux audit service.

Implementation

Task 1. Installation, configuration and running the audit service

1. Install the audit service if it is not present in the system and configure it to run automatically on run-levels 2, 3, 5.

2. Configure the audit service how, and how often the audit logs should be written to disk:
   a. Set the flush parameter to `incremental`;
   b. Make the kernel to flush the data to disk after every 10 records.

3. Start the audit service.

Question A. *How to enable and disable audit while audit service is running*?
Question B. *How to detect whether the audit service is enabled*?

Task 2. Using the audit service

Preparation. For the optional assignment marked with asterisk it is supposed that the components **graphviz**, **mkbar**, **mkgraph**, **gnuplot** are also installed (visit *http://www.graphviz.org/*, *http://people.redhat.com/sgrubb/audit/visualize/mkbar*, *http://people.redhat.com/sgrubb/audit/visualize/mkgraph*, *http://www.gnuplot.info/*). *Note*: to install **gnuplot** download the archive from the web-site, then unpack all archieved files to a separate directory and from that directory execute one by one the following commands: **./configure**, then **make** and finally **make install**.

1. Let this get done during the current boot only:
   a. Set watches to register the executions of a compiled C program and a bash-script.
   b. Query for the events happened to the watched files. ***Question C.*** *Where are audit logs located and what audit component are they processed by*?
   c. Display the list of audit rules you have created and delete a certain rule.
   Question D. *What information is stored in the file titled audit.rules*?

2. Let this work after the system gets rebooted:
   a. Add a watch to the file **`/etc/passwd`** to register write and read events. ***Question E.*** *What information can we obtain by setting up such a watch*?
   b. Query for the events happened to the watched file:
      - today;
      - any time.

   Question F. *Explain the following command:*

```
aureport -ts 12/12/2009 12:00 -te 15/11/2012 00:00 -l
```

3. **Optional*: explore graphical capabilities of Linux audit system:

```
aureport -u -i | awk '/^[0-9]/ { printf "%s %s\n", $4, $7 }' | sort |
uniq | ./mkgraph
```

Question G. *What does this command do*?
Question H. *What are mkgraph and mkbar for*?

4. Prepare a final lab report containing the following information:
   - answers to the questions typed in *italic* (questions A-H);
   - listings of **`audit.rules`** and **`auditd.conf`**;
   - your own conclusions.

1.8 Bash scripting language

Individual lab assignment consists of 8 tasks for each student (see the break-down into 20 variants in the table 1.3 and the list of tasks below the table 1.3).

Table 1.3 – Individual assignments breakdown

Variant No.	**Tasks**							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1.	1	6	11	16-1	17-a	18	19	20
2.	2	7	12	16-2	17-b	18	19	20
3.	3	8	13	16-3	17-c	18	19	20
4.	4	9	14	16-4	17-d	18	19	20
5.	5	10	15	16-5	17-e	18	19	20
6.	1	7	14	16-5	17-b	18	19	20
7.	2	9	13	16-4	17-e	18	19	20
8.	3	10	12	16-2	17-c	18	19	20
9.	4	6	15	16-1	17-b	18	19	20
10.	5	8	11	16-3	17-a	18	19	20

11.	1	10	13	16-5	17-c	18	19	20
12.	2	9	12	16-4	17-d	18	19	20
13.	3	7	11	16-3	17-b	18	19	20
14.	4	8	15	16-2	17-a	18	19	20
15.	5	6	14	16-1	17-e	18	19	20
16.	1	8	12	16-3	17-e	18	19	20
17.	2	6	11	16-5	17-d	18	19	20
18.	3	9	15	16-4	17-b	18	19	20
19.	4	10	14	16-2	17-a	18	19	20
20.	5	7	13	16-1	17-c	18	19	20

According to the table 1.3 create simple bash-scripts to solve the required 8 tasks from the following list:

1. Display the sum of all digits of the number `n` that is given from a keyboard.
2. Compare three numbers and print the largest one. Give the numbers from the command line.
3. For the given number `n` display the sequence `n`, …, `2`, `1` using **`while`** loop.
4. Find out a factorial for a given number `n`.
5. Take a number (`0-10`) and display it in words (for example: `2-two`).
6. Write the given string in its reverse letters order. Check whether the given string is a palindrome or not.
7. Display the given number in its reverse digits order. Check whether the given number is a palindrome or not.
8. Check whether the given service (e.g. **`httpd`**, **`vsftpd`**, etc.) is running or not. If not then propose the user to run it.
9. Display the username, current date and time, and current directory. Try to ping some computers from the local network and display the message whether the ping was successful or not.
10. Count your age (full years, full months, full days). Display a calendar for the current year with your birthday selected. The date of birth is to be given from a keyboard.
11. Determine whether the given file exists or not. The name of the file is supplied as a command-line argument, also check for the sufficient number of command-line arguments.
12. Display all the contents of the file as well as the information regarding to the owner of the file, the number of lines and characters of the file.
13. Make a backup of the given folder from the home directory of the current user into a file called **`abc.tar`**. First check if the given folder exists.
14. Display all the contents of the file from the given line number to the next given number of lines. For example, if we call this script as **`filecon-`**

tents.sh and run it as **sh filecontents.sh 5 5 abc**, the output should contain line number 5 to next 5 line of the **abc** file.

15. Create a text file **statistics.info** which contains the information about the given string: the length of the string, the number of words, spaces, vowels and consonants. Delete all unnecessary spaces from the source string and display both the source string and the result string on the screen.
16. Display the following information:
    1. the type and version, release number, kernel version of the installed OS;
    2. the home directory of the current user, the current working directory and the current path setting;
    3. the number of currently logged users, the list of all available shells and the current shell;
    4. the information about the CPU (processor type, speed, etc.) and the computer memory;
    5. the information about the hard disk (the size of the hard disk and its partitions, cache memory, model, etc.) and display all mounted file systems.
17. Display the following patterns on the screen:

```
    1        1      1              1           x
   2 2       22     12           2 1 2         0x
  3 3 3      333    123        3 2 1 2 3       00x
 4 4 4 4     4444   1234     4 3 2 1 2 3 4     000x
5 5 5 5 5    55555  12345  5 4 3 2 1 2 3 4 5   0000x
   a)        b)     c)            d)           e)
```

18. Create a simple calculator (using the **case** structure), which takes arguments from the command line, e.g. **2+3**, **2*3,** etc. and displays the result. It should perform the following five operations: addition, subtraction, multiplication, division, modulo.
19. According to the system time display one of the messages below:
    - *Good Morning!*
    - *Good Afternoon!*
    - *Good Evening!*

    The script should run as soon as the current user logs in. Put the contents of this script into the file **~/.bash_profile**.
20. Make a single script **your_surname.sh** by putting all the previously created scripts into functions. The final script must also contain the function of displaying the main menu that has items in accordance to the individual assignments. Use a case statement for the menu function.

2 PRACTICAL ASSIGNMENTS

Purpose: study the work of essential network services and their configuration files within learning basic administrative functions and getting practical experience of managing Linux-based LANs.

Instruments: 1. Red Hat Enterprise Linux 6 (RHEL) – it will act as a server.
2. RHEL or any other Linux machine – it will act as a client.
3. Windows XP/Seven or Windows Server 2008 (when needed).

Note: each OS can be installed either on a real machine or on a virtual machine.

Preparation: 1. Set proper IP-addresses to machines to make them all belonging to the same LAN (IP-addresses should not be reset after reboot).
2. Using a **`ping`** command check whether machines "see" each other within the network.

2.1 NFS

Objective: *configure the NFS service and explore its work from the point of view:*

- *of differentiated permissions set to shared resources (directories), and*
- *of access attempts from Linux- and Windows-based client machines.*

Implementation

1. Acting as **`root`** perform the following:
   a. Create empty directories:
      - **`/tmp/public`**
      - **`/tmp/somehosts`**
      - **`/tmp/linookz`**
      - **`/tmp/linookz/inner`**

   b. In the directory **`inner`** using a single console command create empty files titled **`muxi_0`**, …, **`muxi_9`** and **`bzzz`**. Give to all created files read and write permissions for all users. Put some content into files **`muxi_0`** and **`bzzz`**. ***Question A.*** *What are the commands to be used?*

2. Install and configure the NFS service:
   a. Install all necessary NFS service packages.
   Question B. *What packages does the NFS service need?*

   b. Edit the NFS service configuration file to perform the following tasks:
      - allow a full access to the **`/tmp/public`** directory for all hosts;
      - allow a full access to the **`/tmp/linookz`** directory for the certain IP address (of a Linux-based machine) and a read-only access for another IP address (of a Windows-based machine);
      - allow a full access to the **`/tmp/somehosts`** directory for the hosts having IP addresses of the range from `192.168.x.a0` to `192.168.x.a9`;
      - allow a read-only access to the directory **`/home`** for all machines within the network.

      Question C. *What is the configuration file to be edited?*

c. Set read and write permissions for "*others*" to all directories which are to be exported.

d. Make the service to be started automatically on run-levels of a full multi-user mode.

e. Start the NFS service.

3. Demonstrate the work of the NFS service:

a. Try to connect remotely to all the shared directories (both from WinXP and Linux client machines). *Note:* to get a remote access to a shared directory from a Linux-based client machine you need to mount that directory.

b. Try to create some new directories and files, try to edit existing ones.

Question D. *How to unexport all the exported directories with no any changes in the NFS service configuration file? Of course, it is considered that all the exported files and directories should not be physically deleted.*

4. Reboot both server and client systems. Explain what happened or what might happen with the NFS service and shared resources.

Question E. *How to make shared resources to be accessible after:*

a) the server machine is rebooted? *b) the client machine is rebooted?*

5. Fill the table below by placing "**+**" if a remote access attempt was successful and "**-**" if not (create a user account `john` if it is absent in the system).

Table 2.1 – NFS resources and permissions

No.	**Resource**	**Linux client** IP: `192.168.x.ab`		**Windows client** IP: `192.168.x.cd`	
		write	*read*	*write*	*read*
1.	`/tmp/public`				
2.	`/tmp/linookz`				
3.	`/tmp/linookz/inner`				
4.	`/tmp/somehosts`				
5.	`/tmp/home`				
6.	`/tmp/home/nato`				
7.	`/tmp/home/john`				

6. Using the `history` console command show which commands did you use to:

a. Start the NFS service.

b. Create directories and files of given properties.

7. Prepare a final lab report containing the following information:

- IP addresses and OS names of all three machines,
- answers to the questions typed in *italic* (questions A-E);
- the filled table 2.1;
- the history of the users `root`, `nato` and `john`;
- your own conclusions.

2.2 Samba

Objective: *configure the Samba service and explore its work from the point of view:*

- *of differentiated permissions set to shared resources (directories), and*
- *of access attempts from Linux- and Windows-based client machines.*

Implementation

1. Acting as `root` on the server machine do the following:
   a. Create two Samba users `smbperson` and `smbuser` and at least one non-Samba user (say, `nato`).
   b. Create the following directories (if they don't exist) and set proper SELinux options to them:
      - `/tmp/public` - `/tmp/linookz`
      - `/tmp/somehosts` - `/tmp/linookz/inner`

      Question A. *What are the console commands for setting up and viewing SELinux options?*

2. Install and configure the Samba service:
   a. Install all necessary Samba service packages if needed. ***Question B.*** *How many packages does the samba service need? What are they?*
   b. Configure the Samba service to run automatically on run-levels 3 and 5. ***Question C.*** *Do we need to use the second run-level? Why?*

3. Create and manage Samba resources as shown below:
   a. Make all Samba resources to be available within `192.168.10.x` and `192.168.1.x` networks.
   b. Allow a read-write access to the `/tmp/public` directory for all hosts.
   c. Allow a full access to the directory `/tmp/linookz` for the Samba user `smbperson` only.
   d. Allow a full access to the `/tmp/somehosts` directory for the hosts with IP address range from `192.168.x.a0` to `192.168.x.a9`.
   e. Allow a read only access to the directory `/home`.

      Question D. *What is the Samba service configuration file to be edited?* *Note:* to check the `smb.conf` file for syntax use the `testparm` command.

4. Start the Samba service and demonstrate its work:
   a. List all Samba shared resources both from the Samba server and the Samba client Linux-based machine.
   b. Try to connect remotely to all the Samba shared directories (both from Windows-based and Linux-based client machines). Try to create some new directories and files, try to edit existing ones. ***Question E.*** *Is it possible to create a file in the directories linookz and linookz/inner?*

c. List all browseable Samba shared resources from both Linux-based and Windows-based client machines.
d. From both Windows-based and Linux-based client machines try to download some files from different Samba shared directories. Explain when the access to Samba directories through a browser is possible.

5. Explore remote access attempts: fill the table below by placing “r” and/or “w” if read/write operations were successful and “no” if neither read nor write operations were allowed:

Table 2.2 – Samba resources and permissions

No.	Resource	Linux client IP: `192.168.x.ab`			Windows client IP: `192.168.x.cd`		
		`smbperson`	`smbuser`	`nato`	`smbperson`	`smbuser`	`nato`
1.	`/tmp/public`						
2.	`/tmp/linookz`						
3.	`/tmp/linookz/inner`						
4.	`/tmp/somehosts`						
5.	`/home`						

6. Using the `history` console command show which commands did you use to:
a. Start the Samba service.
b. Create directories and files of given properties.

7. Prepare a final lab report containing the following information:
- IP addresses and OS names of all three machines;
- answers to the questions typed in *italic* (questions A-E);
- content of the Samba service configuration file;
- the filled table 2.2;
- the history of the users `root`, `smbperson`, `smbuser` and `nato`;
- your own conclusions.

2.3 FTP

Objective: *connect to the remote system via FTP from Linux-based and Windows-based client machines as anonymous and authorized users, to allow and deny FTP-access for different users.*

Implementation

Preparation: create a user account `ftp_user` on the server machine.

1. If needed install the package corresponding to the Very Secure FTP service onto the system which is considered to be an FTP server. Try to connect via FTP from the client machine.
Question A. *What packages are to be installed onto the FTP server and client machines?*

2. Make the FTP service to run at start-up. ***Question B.*** *How can this get done?*

3. In the main configuration file of the FTP service make settings to:
   - display a welcome message:
     `"This is FTP service. Anonymous users are welcome!"`;
   - allow uploading files by anonymous users.
     Question C. *Where is the main configuration file of the Very Secure FTP service located and which settings are to be edited?*
     Note 1: configure or stop Linux firewall if needed.
     Note 2: after installing the Very Secure FTP service on RHEL, when trying to log in (say, as **`user1`**) the following error may appear:

     ```
     500 OOPS: cannot change directory:/home/user1
     ```

     The problem is that SELinux is standing in the way. To fix this use the following command:

     ```
     sudo /usr/sbin/setsebool -P ftp_home_dir 1
     ```

     or

     ```
     setsebool -P ftp_home_dir 1
     ```

     To check if SELinux is enabled use the following command:

     ```
     sestatus
     ```

     You may disable SELinux by executing the command:

     ```
     setenforce 0
     ```

     and by setting up the following parameters in the file **`/etc/sysconfig/selinux`**:

     ```
     anon_world_readable_only=NO
     setsebool ftpd_disable_trans 1
     ```

     After changing SELinux options the FTP service should be restarted.

4. Connect to the FTP server remotely as an anonymous user:
   a. From a browser and try to download any file from the FTP server.
   b. From a Linux-based client (console mode) and try to upload any file to the server. ***Question D.*** *Where will it be stored by default?*

5. Connect to the FTP server remotely as **`ftp_user`**:
   a. From a Linux-based client (***Question E.*** *What is the command to be used?*) and perform the following actions:
      - display the current directory;
      - upload any file to the remote machine (***Question F.*** *Where will it be copied by default?*);
      - copy any file from remote machine to your home directory.
   b. From a Windows-based client (***Question G.*** *How can this get done?*) and perform the following actions:
      - list all files in the current directory of the remote machine;

- remove the file copied to the remote machine earlier (***Question H.** Where will it be copied to by default?*);
- download the file from remote machine to your home directory.

c. Close FTP-connections from both Linux-based and Windows-based client machines. ***Question I.** What is the command to be used?*

6. Make appropriate changes in the main configuration file of the Very Secure FTP service to provide the following:
   a. Deny anonymous connections.
   b. Create a user `ftp_person` and allow an FTP-access for this account.
   c. Deny FTP-access to the users `ftp_user` and `root`.

7. Make sure new settings are working – try to connect remotely via FTP as an anonymous user, as users `ftp_person`, `ftp_user` and `root`.

8. Prepare a final lab report containing the following information:
   - IP addresses and OS names of all three machines (indicate the IP address of the system the users `ftp_person` and `ftp_user` belong to);
   - answers to the questions typed in *italic* (questions A-I);
   - the content of the following files:
     i. `/etc/vsftpd/ftpusers`,
     ii. `/etc/vsftpd/user_list`,
     iii. the main configuration file of the Very Secure FTP service (omit comments);
   - your own conclusions.

2.4 SSH

***Objective:** connect to the remote system using SSH:*
- *with passwords and without cryptographic keys;*
- *without passwords but with cryptographic keys.*

Implementation

1. Install the SSH service if it is not present in the system.
***Question A.** What is the command to check whether it is installed?*
***Question B.** What are the names of SSH packages to be installed?*

2. Configure the SSH service to run automatically at start-up on run-levels 3, 5.
***Question D.** How can this get done?*
***Question E.** What is the name of the SSH service?*
***Question F.** What is the main configuration file of the SSH service?*

3. Establish an "*easy*" connection (i.e. with password & without keys):
   a. Connect to the server from Linux-based client machine, create somewhere an empty file called `empty` and close the connection.

b. Connect to the server from Windows-based client machine (e.g. using PuTTY, edit the file `empty` and close the connection.

c. On one Linux-based machine in home directory of the user (e.g. `nato`) create a file called `for_scp_nato` and copy it to, say, `john`'s home directory of another Linux-based machine using the following command:

```
scp <source> <destination>
```

Note: Using an option `-r` we can copy directories which are non-empty.

4. Establish a "*complex*" connection (i.e. without password & with keys):

a. Consider we have 2 different users on 2 different machines – a user `chip` on one machine (`192.168.10.21`) and a user `dale` on another one (`192.168.10.22`). *Note*: to create users run a GUI `system-config-users`.

b. Create a folder `.ssh/` in `$HOME/` on both machines with permissions `700`.

Question G. *What does the dot before ssh/ mean?*

Question H. *How can we look at permissions of the directory* `.ssh/`*?*

c. Generate a public key (run the command below on both machines):

```
ssh-keygen -t dsa
```

Note: the passphrase query can be left empty.

Question I. *What are the names of files appeared in the directory* `.ssh/` *?*

d. Using the `scp` command both users must give each other their public keys:

- on the 1st machine (for copying `chip`'s public key to `dale`'s machine):

```
scp $HOME/.ssh/id_dsa.pub dale@192.168.10.22:/home/dale/.ssh/authorized_keys
```

- on the 2nd machine (for copying `dale`'s public key to `chip`'s machine):

```
scp $HOME/.ssh/id_dsa.pub chip@192.168.10.21:/home/chip/.ssh/authorized_keys
```

e. On both machines set permissions `600` to those keys.

Question J. *What is the command to be used?*

f. Now users `chip` and `dale` may connect to each other's machines using their public keys without any passwords. Check whether the connection can be established from both machines:

```
ssh <user>@<IP-address>
```

Note: everything went successful if you were not prompted for password to use the command above.

5. Prepare a final lab report containing the following information:

- IP addresses and OS names of all three machines (indicate the IP addresses of the systems users `chip` and `dale` belong to);
- answers to the questions typed in *italic* (questions A-J);
- the history of users `chip`, `dale`, `nato`, `john`, and `root` (on both machines);
- your own conclusions.

2.5 SQUID

Objective: *configure the SQUID service in order to restrict Internet-connections by different parameters (users, protocols, download speed limits, etc.).*

Implementation

Preparation: the intranet of the university will be used. Build the environment according to the figure 2.1.

1. Install the SQUID package and make the SQUID service to run automatically at start-up on full multiuser run-levels.
Question A. *What commands are to be executed?*

2. Study the man page and the main configuration file of the SQUID service. ***Question B.*** *How is the main configuration file of the SQUID service called?* Perform the following tasks:

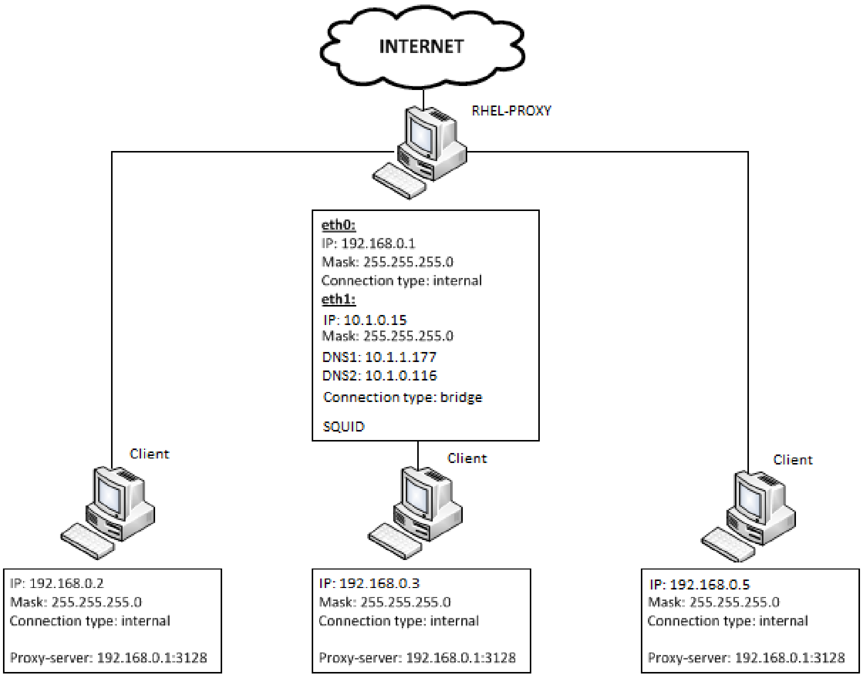


Figure 2.1 *General scheme*

a. Add three network groups:

- *full-access*: `192.168.0.1 – 192.168.0.5`,
- *other-access*: `192.168.0.6–192.168.0.100`,
- *our_network*: `192.168.0.0/24`.

b. Create a file **`/etc/squid/acl/bad_url`** containing two lines:

```
.test.brsu.by
.tour.brsu.by
```

and specify a separate Access Control List (ACL) called **`bad_url`** which refers to the file **`/etc/squid/acl/bad_url`**.

c. Make all three groups to have Internet-access, but deny access to "bad" URLs specified earlier to everyone except the members of the group called *full-access*.

d. On the `192.168.0.2` machine create a user **`frodo`** and deny both FTP and HTTPS access for him.

e. On the `192.168.0.3` machine create a user **`gendalf`** and allow Internet-access for him from 10:00 till 15:00.

f. Limit the download speed of mp3-files up to 10 Kb/s for all hosts.

g. **Optional*: create a group called *trolls* containing machines with IP addresses from `192.168.0.5` to `192.168.0.9` and limit the download speed of any files up to 10 Kb/s for *trolls* in general and up to 5 Kb/s for each member of *trolls*.

h. From the main SQUID configuration file study SQUID caching and specify the "freshness" for mp3- and avi-files to 30 days.

3. Run the SQUID service and check if settings you have made are applied. *Note:* the work of the SQUID service may be "spoiled" by Linux firewall. ***Question B.*** *What actions may get taken to Linux firewall in order to allow the work of the SQUID service?*

a. In the browser of the client machine set the IP address of proxy-server. ***Question C.*** *What is the default port number used by the SQUID service?*

b. Try to access web-sites:

- *http://www.brsu.by*,
- *http://test.brsu.by*,
- *http://hmath.brsu.by*,
- *http://tour.brsu.by*.

c. After checking the work of the SQUID service fill the table below:

Table 2.3 – SQUID permissions

Subject	**IP address or range**	**download speed** (Kb/s)	**protocols** (0 – if failure, 1 – if success)		
			HTTP	**HTTPS**	**FTP**
full-access					
other-access					
our_network					
frodo					
gendalf					
trolls					

4. Prepare a final lab report containing the following information:

- answers to the questions typed in *italic* (questions A-C);
- the content of the main configuration file of the SQUID service;
- the filled table 2.3;
- the history of users **`frodo`**, **`gendalf`** and **`root`**;
- your own conclusions.

2.6 iptables

Objective: *configure Linux firewall in order to show how it can protect the system by restricting LAN and Internet connections with different parameters.*

Implementation

Preparation: make one of Linux machines to be a proxy server (see the previous lab assignment).

1. Study the man page related to Linux firewall. ***Question A.*** *What does the term "chain" mean and how Linux firewall rules are processed?*

2. Make sure that Linux firewall is running. ***Question B.*** *What run-levels are set by default to run Linux firewall automatically at start-up?*
Note: when creating scripts containing rules for `iptables`, firstly – reset all existing policies and secondly – do not use the command `service iptables save`. ***Question C.*** *Why?*

3. On the SQUID client machine create a bash-script `ipt_a.sh` containing proper firewall rules to implement the following tasks:
   a. By default all incoming and outgoing packets should be blocked.
   b. Internet access via proxy should be allowed.
   Question D. *How can ICQ connections get denied when using proxy*?
   Question E. *How is the firewall on the SQUID server machine configured – to deny or to allow connections which are initiated out of local network? How to allow/deny them?*

4. On the same machine (from now and further we won't use it as a SQUID client, even the SQUID service may be switched off) create a bash-script `ipt_b.sh` containing proper firewall rules to implement the following tasks:
   a. Set default policies to allow all incoming and outgoing traffic.
   b. Deny access to the site *facebook.com.*
   c. Deny ICQ connections. ***Question F.*** *How can ICQ connections get denied when there is no proxy*?
   d. Deny viewing *https* web-pages.
   e. Using a module `comment` add comments to the rules **c** and **d**.
   f. Make the restrictive rules **b** and **d** to be logged by `rsyslogd` to a separate file (e.g. `/var/log/kernel/iptables_log`).

5. Create a bash-script `ipt_c.sh` containing proper firewall rules (and comments as well) to implement the following tasks:
   a. Set default policies to allow all incoming and deny all outgoing traffic.
   b. Using proper firewall rules compare the `ping` command behavior (try to ping a firewalled machine from any other machine):

- when all *icmp packets* are blocked;
- when all *ping requests* coming from `192.168.10.21` are blocked;
- when all *ping replies* to the whole network are blocked;
- when *ping packets* are allowed.

c. Deny all Samba connections.

6. Create a bash-script **`ipt_d.sh`** containing proper firewall rules (and comments as well) to implement the following tasks:

a. Set default policies to deny all incoming and allow all outgoing traffic.

b. Allow **`ssh`** and **`telnet`** connections between the current system and:

- machines of the IP range from `192.168.10.20` to `192.168.10.25`,
- two machines with MAC addresses (say, `08:00:27:A6:8C:B3` and `08:00:29:K6:5C:E4`).

7. Prepare a final lab report containing the following information:

- answers to the questions typed in *italic* (questions A-F);
- the content of files **`ipt_a.sh`**, **`ipt_b.sh`**, **`ipt_c.sh`**, **`ipt_d.sh`**;
- the history of the user **`root`**;
- your own conclusions.

2.7 DHCP

Objective: *install and configure the DHCP server.*

Implementation

1. Install a package of the DHCP service. ***Question A.*** *What are the packages to be installed on server and client machines?*

2. Configure the DHCP service to start automatically on run-levels 3 and 5.

3. Edit the main configuration file of the DHCP service:
(*Note:* the example of the DHCP service configuration file can be taken from **`/usr/share/doc/dhcp*/dhcpd.conf.sample`**)

a. Assign a single IP-address to the Windows-based client machine according to its MAC-address.

b. Make a range of ten IP-addresses to be assigned to other machines.

4. Start the DHCP service and check whether all settings are working.
Question B. *What is the port number used by the DHCP server? What command can be used to find out this port number?*
Question C. *What measures are to be taken on Windows-based and Linux-based client machines to get their IP addresses automatically from the DHCP server*?

5. Now do the opposite: assign a single IP-address to the Linux-based client according to its MAC-address and make a range of ten IP-addresses to be assigned to other machines. Check whether all settings are working.

6. Explore the file **`/var/lib/dhcp/dhcpd.leases`** and manage lease intervals in the following way:

   a. Set the maximum and default lease intervals to 48 hours and 24 hours correspondingly for all DHCP client machines.

   b. Set the maximum and default lease intervals to 1 hour and 30 minutes correspondingly for the Linux-based DHCP client machine only.

   c. Check if the lease settings are applied successfully.

   Question D. *Are there any changes in /var/lib/dhcp/dhcpd.leases?*

7. Prepare a final lab report containing the following information:

   - answers to the questions typed in *italic* (questions A-D);
   - content of configuration files listed below:
     - i. **`dhcpd.conf`**,
     - ii. **`/var/lib/dhcp/dhcpd.leases`**,
     - iii. **`/etc/sysconfig/network-scripts/ifcfg-eth0`** (both from the DHCP server and Linux-based client machines),
     - iv. **`/etc/sysconfig/network`** (both from the DHCP server and Linux-based client machines);
   - your own conclusions.

2.8 Apache, PHP, MySQL

Objectives: 1. Install and configure a web-server, PHP and MySQL service;
2. Create a simple web-site to demonstrate the relation of Apache, PHP and MySQL (here instead of PHPMyAdmin the **`mysqld`** prompt will be used to maintain databases).

Implementation

Task 1. Apache

1. Install and run a service for the web server Apache.

Question A. *What is the name of the service for Apache*?

2. Start the web-server and configure it to run on run-levels 3 and 5 automatically after each reboot.

3. Create a simple web-site consisting of an HTML-page **`index.html`** displaying the message “`Hello, World`”.

Question B. *Where are all web-pages located*?

4. Try to access the web-site by the IP-address of the web-server from:

   a. the web-server machine itself,

   b. a Linux-based client machine,

   c. a Windows-based client machine.

Question C. *What do you have to do to make the access attempts successful*?

Task 2. PHP

1. Install PHP software packages.
***Question D.** What is the name of the package(s) did you choose to install*?

2. Create a simple PHP-page **`index.php`** displaying the message "`Hello, Earth!`" and check if it is working.

3. Access the web-site by the IP-address of the web-server from:
   a. the web-server machine itself,
   b. a Linux-based client machine,
   c. a Windows-based client machine.

4. In the main configuration file of PHP change the following parameters:
   a. Allow short tags (and get ready to show that it is working).
   b. Display all errors. ***Question E.** What is the name of the main configuration file of PHP and where is it located*?

5. Create a PHP-script **`ip_md5_echo.php`** which displays IP-address of the client machine and returns the md5-value of the string entered by a user.

Task 3. MySQL

1. Install MySQL software packages.
***Question F.** What are the name of the packages to be installed*?

2. Start the MySQL service and configure it to run on run-levels 3 and 5 automatically after each reboot.

3. Create a PHP-script **`ip_md5_db.php`** (make changes to the PHP-script **`ip_md5_echo.php`** created earlier to act the same way but with the only difference that all the md5-values must be stored in a database).

4. Access the created web-application by the IP-address of the web-server from:
   a. the web-server machine itself,
   b. a Linux-based client machine,
   c. a Windows-based client machine.

5. Prepare a final lab report containing the following information:
   - IP addresses and OS names of all three machines;
   - answers to the questions typed in *italic* (questions A-F);
   - the source codes of **`ip_md5_echo.php`** and **`ip_md5_db.php`**;
   - the history of:
     i. the users **`root`** and **`nato`**,
     ii. the commands used in **`mysql`** prompt;
   - your own conclusions.

2.9 DNS

Objectives: 1. Install and configure the DNS service.
2. Implement a possibility to access the web site using it's "friendly name" (instead of typing the IP address in browser) from the current machine (which is acting as a DNS server) as well as from other machines within the network.

Implementation

Preparation: 1. Boot the RHEL system with graphical support.
2. Suppose we have:
- the IP address of the RHEL machine is `192.168.1.121`;
- Apache and MySQL services are running.

3. Assume we want to access the web-site **`travel.biz`**.

1. Install the package(s) of the DNS service.
Note: since for this assignment a chroot environment is not necessary, then do not install the package called **`bind-chroot-9.7.0.-5.P2.el6.i686`**.

2. Configure the DNS service to start automatically on run-levels 3 and 5.
Question A. *What is the command to be used?*

3. Configure the DNS service:
   a. Create forward and reverse zones.
   - In the file **`named.rfc1912.zones`** append records of forward and reverse zones of the host (call the forward and reverse zones **`travel.biz`** and **`1.168.192.in-addr.arpa`** correspondingly).
     Question B. W*hich else configuration file of the DNS service can we type these zones in?*
   - Create both files of these two zones.
     Question C. *Where do they have to be created?* Get the answer from the main DNS configuration file and put necessary records there (get ready to explain each record).

   b. Make proper changes in the main configuration file of the DNS service that will allow access from all machines within the network to the website located on the same server (that is acting as a DNS server).

4. Add records to resolve IP addresses to host names for **`localhost`**, **`travel.biz`** and **`www.travel.biz`**. ***Question D.*** *What is the configuration file to be edited?*

5. After you complete editing all configuration files restart the network, DNS service, and Apache.

6. Check if DNS service is working:
   a. Try to access the web-site (which you created earlier or default one) by typing in the browser:

- *http://www.travel.biz*,
- *http://travel.biz*,
- *http://192.168.1.121*.

Try to access these sites from the current machine (which is acting as a DNS and web server) and from any other machine within the network.

Question E. *Were all attempts successful?*

b. Execute the following commands (both from the current machine which is acting as a DNS and web server and from the client machine):

- **`dig -x 192.168.1.121`**
- **`host -al travel.biz`**
- **`nslookup 192.168.1.121`**
- **`nslookup travel.biz`**

Question F. *Can we have a DNS service and a web-service running on different machines?*

7. Prepare a final lab report containing the following information:
   - answers to the questions typed in *italic* (questions A-F);
   - content of configuration files listed below:
     i. **`rndc.key`**,
     ii. the main DNS service configuration file,
     iii. **`named.rfc1912.zones`**,
     iv. the forward zone file (explain each record of the file in comments),
     v. the reverse zone files (explain each record of the file in comments),
     vi. the configuration file responsible for resolving IP-addresses;
   - the results returned by each command of the item 6-b;
   - your own conclusions.

2.10 Sendmail

Objectives: 1. Install and configure the Sendmail service;

2. Implement a possibility to send mails using the friendly name of the mail server (instead of typing the IP address of the mail server) from the current machine (which is acting as a DNS and Sendmail server) as well as from other machines within the network. *Note*: to reach this goal both DNS and Sendmail services must be installed and properly configured.

Implementation

Preparation: 1. Suppose we have the DNS service is running.

2. Assume we want to send a mail to the user **`nato`**, registered on the server machine (**`nato`**'s e-mail address will look like `nato@travel.biz`) from the current machine (RHEL) as well as from other machines within the network.

1. Install packages needed for the Sendmail service. ***Question A.*** *Which ones?*

2. Configure the Sendmail service to start automatically on run-levels 3 and 5 and show that the changes you made have been applied.
Question B. *List the commands to be executed.*

3. Make proper setting in the main configuration file of the Sendmail service.
Question C. *What is the main configuration file for the Sendmail service?*

4. Compile the edited main configuration file of the Sendmail service into a `*.cf` file and start the Sendmail service.
Question D. *What are the commands to be executed?*

5. Check the work of the Sendmail service: try to send letters between users registered in the current and remote systems.
Question E. *Where are the mailboxes located?*

6. Prepare a final lab report containing the following information:
   - IP addresses and OS names of both machines;
   - answers to the questions typed in *italic* (questions A-E);
   - the history of the users `nato` and `root` on both Linux-based machines;
   - your own conclusions.

2.11 NTP

Objectives: get the knowledge how date and time in the LAN can get synchronized from the Internet: configure the NTP server as the 3-level time server in such a way that it will get the date and time from the Internet and will assign it to the clients within the local network that contains Linux- and Windows-based machines.

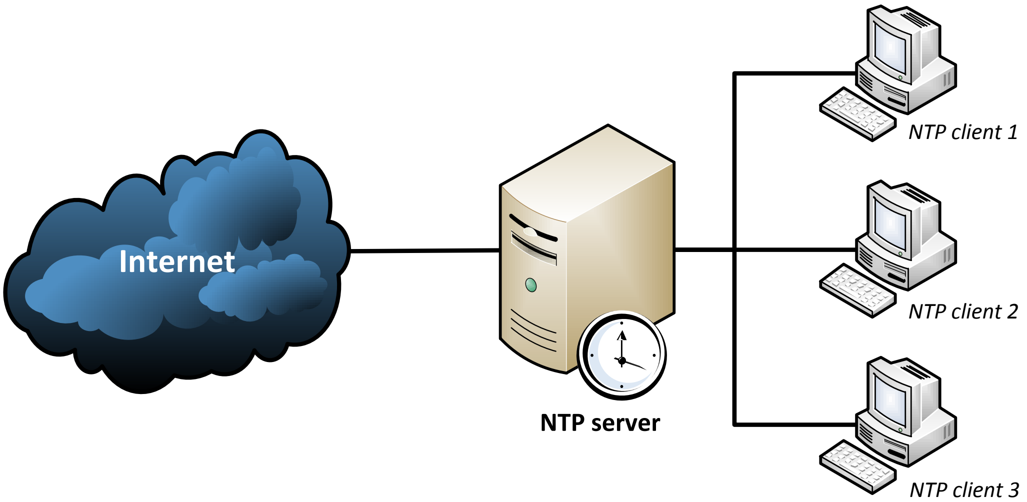


Figure 2.2 *The LAB scheme*

Implementation

1. Install the NTP service if it is not present in the system.
Question A. *What is the command to check whether it is installed?*
Question B. *What is(are) the name(s) of NTP package(s) to be installed?*

2. Make the NTP service to run automatically at start-up.
Question C. *How can this get done?*

3. Study the NTP service configuration file (***Question D.*** *Which one?*) and explain what is stored in a so called `driftfile`.
Question E. *Where are NTP log files stored?*

4. Configure the NTP service according to the following:
(***Question F.*** *What is the name of NTP service and what command can be executed to see a port number that is used by the NTP service?*)

   a. From the Internet find out at least three NTP servers and put their URLs into the main configuration file of the NTP service.

   b. Use the following commands:
      - `ntpq -p` (to check time servers);
      - `ntpdate -q localhost` (to check the stratum of the current system).

5. Start the NTP service and configure other machines to get the time from the configured one:

   a. To synchronize a Linux-based client with the configured machine use the command:

   ```
   ntpdate <IPaddr_NTP_server_machine>
   ```

   Notes:
      - If an error has occurred then use the command below to find out the problem:

        ```
        ntpdate –d <IPaddr_NTP_server_machine>
        ```
      - For debug purpose use one of the commands below:

        `date MMDDhhmmCCYY.ss` or `date –s hh:mm`

        to change the current date and time and then try to synchronize the client machine with the NTP server.

   b. To synchronize a Windows-based client with the configured machine double click on the time in the taskbar, in the “Date and Time Properties” window open the tab “Internet Time” and specify the IP address of the machine configured to be a NTP server.
      - For debug purpose change the current system date and time and then try to synchronize the client machine with the NTP server.

   Note: all three machines need to belong to the same time zone, otherwise the time will get synchronized but it will differ on different machines.

6. Configure the NTP service to get the time from the current (local) machine and try to get all time client machines synchronized with the time provided by the NTP server:

   a. When the Internet connection is on.

   b. When the Internet connection is off.

7. Configure the system that is running the NTP service as a stratum 3 time server and repeat steps 5 and 6.

8. Prepare a final lab report containing the following information:
   - IP addresses and OS names of all three machines;
   - answers to the questions typed in *italic* (questions A-F);
   - the history of `root` users on both Linux-based machines;
   - your own conclusions.

2.12 Kerberos

Objective: authenticate a user with Kerberos tickets: get from a Linux-based machine remote access to shared resources located on Windows-based machines via Samba using Kerberos tickets.

Preparation

Build the environment according to the figure 2.3 (you should configure firewalls properly on each machine if needed, but for this task it is allowed simply to switch off all firewalls):

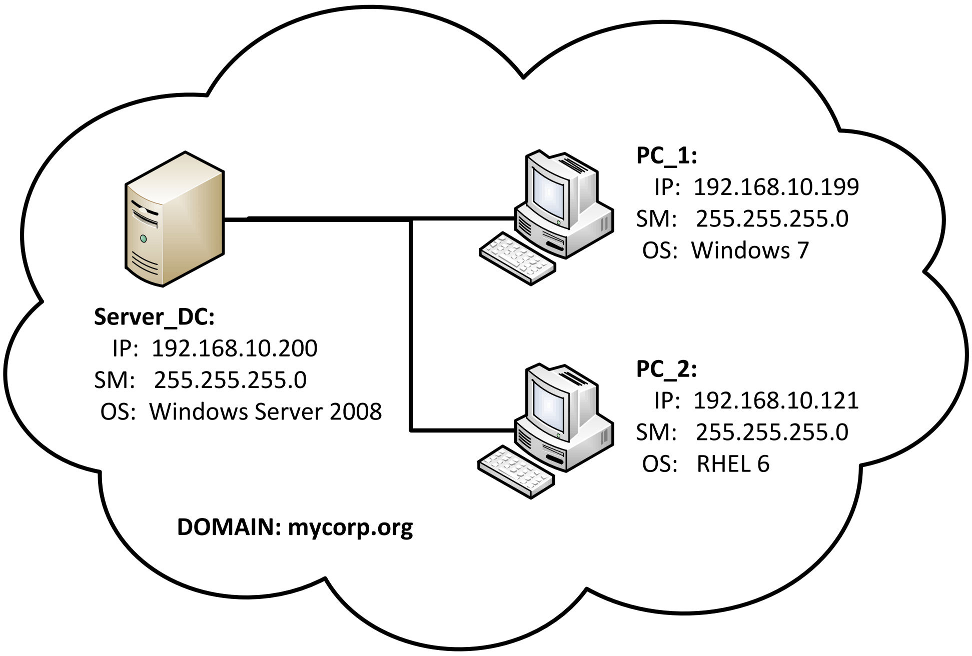


Figure 2.3 *General scheme*

Thus, here the machine running Windows Server 2008 is the domain controller, and machines running Windows 7 and RHEL 6 are the client PCs.

1. Configure the machine that is running Windows Server 2008 as a domain controller:

   a. Go *Start – Administrative Tools – Server Manager*, click *Add Roles*, and then *Next.*

   b. Select the server role called *Active Directory Domain Services*, click *Next* and then confirm installation selections by pressing the *Install* button.

   c. When the role installation process gets completed close the *Add Roles Wizard.*

   d. In the *Server Manager* window click *Active Directory Domain Services* and then from the *Advanced Tools* section run *Dcpromo.exe* for setting up a domain.

   e. Follow the instructions of the *Active Directory Domain Services Wizard* and specify the following options:

      FQDN of the forest root domain = *mycorp.org*,
      Forest functional level = *Windows Server 2008*.

      Then check the additional option *DNS Server* (other options can be left by default) and specify a strong password for the *Directory Services Restore Mode Administrator account*.

   f. After the installation process gets completed reboot the system.

   g. Go *Start – Administrative Tools – Server Manager*, from the left pane expand *Roles – Active Directory Domain Services – Active Directory Users and Computers – mycorp.org* and right click on the item *Users*. From the context menu choose *New – User* and add a new user account *nato*.

2. Bring the machine that is running Windows 7 to the domain *mycorp.org*:

   a. log in as the administrator of the client machine;

   b. go *Start*, right click on the item *Computer and* select *Properties*;

   c. from the left pane of the *System* window click on the *Advanced system settings* link;

   d. in the *System Properties* window click the button *Change* on the *Computer Name* tab and specify any desirable computer name and *mycorp.org* as a domain name this computer to be a member of (you will be prompted for the password of the domain's administrator).

   e. Reboot the client system to get changes applied.

3. Create folders **`C:\shared_DC`** and **`C:\shared_PC`** on the *Server_DC* and *PC_1* systems and make those folders shared for the domain user **`nato`** created earlier.

4. Configure the *PC_2* machine that is running RHEL 6:

   a. Specify a DNS server for the network interface (say, **`eth0`**) – in the file **`/etc/sysconfig/network-scripts/ifcfg-eth0`** add a line:

```
DNS1=192.168.10.200
```

Thus, the file **/etc/sysconfig/network-scripts/ifcfg-eth0** will contain lines like listed below:

```
DEVICE="eth0"
HWADDR="08:00:27:31:62:00"
NM_CONTROLLED="yes"
ONBOOT="yes"
IPADDR=192.168.10.121
NETMASK=255.255.255.0
DNS1=192.168.10.200
BOOTPROTO="none"
```

b. Then restart the network and execute the command below:

$ service network restart[1]

Note: when network gets restarted then in the file **/etc/resolv.conf** the following strings will appear:

```
#Generated by NetworkManager
nameserver 192.168.10.200
```

c. Now check whether the domain *mycorp.org* is accessible:

$ ping mycorp.org

Implementation

In the implementation part the Linux-based machine will be configured to use Kerberos tickets. This process includes the following steps.

1. Configuring Kerberos settings:

a. To get Kerberos tickets the packages

- **krb5-workstation-1.8.2-3.el6.i686.rpm**
- **krb5-libs-1.8.2-3.el6.i686.rpm**

are needed[2] (if they are not installed in the system, install them manually using the **rpm -i** command).

b. Make changes to the default file **/etc/krb5.conf** according to our needs:

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log
```

[1] The **$** sign means the command should be executed in the terminal window, and the symbol **#** signifies either the output of the command or (if it is mentioned in configuration files) the comment sign.

[2] The exact name of the rpm package satisfies the following format:

<name>-<version>-<release>.<architecture>.rpm

Obviously, **rpm** packages of another version, release and architecture may be used to configure the same environment as described in the assignment.

```
[libdefaults]
 default_realm = MYCORP.ORG
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
[realms]
 MYCORP.ORG = {
  kdc = mycorp.org
  admin_server = mycorp.org
 }
[domain_realm]
 .mycorp.org = MYCORP.ORG
 mycorp.org = MYCORP.ORG
```

Note: the syntax of this file is case sensitive.

c. Check the work of Kerberos with the commands listed in the table below:

`kinit nato`	Get a Kerberos ticket for the user **`nato`**
`klist`	Look at the received Kerberos ticket
`kdestroy`	Destroy the current Kerberos ticket

Note: if the message *Clock skew too great while getting initial credentials* appears you can synchronize the machine with the domain controller:

```
$ ntpdate 192.168.10.200
```

Question A. *What is the use of synchronizing the time?*

2. Getting access to the shared resources via Samba.

a. The following Samba packages are needed (if they are not installed by default install them manually using the **`rpm -i`** command):

```
samba-common-3.5.4-68.el6.i686
samba-client-3.5.4-68.el6.i686
samba-winbind-clients-3.5.4-68.el6.i686
```

Question B. *Is it necessary to have the* ***`samba-3.5.4-68.el6.i686`*** *package installed and Samba service running to access remote shared resources via Samba*?

b. Get the access to folders shared earlier as a domain user **`nato`** (but with the need to specify **`nato`**’s password):

```
$ smbclient //192.168.10.200/Shared_DC –U nato
#Enter nato’s password:
#Domain=[MYCORP] OS=[Windows Server 2008 R2 Enterprise 7600]
Server=[Windows Server 2008 R2 Enterprise 6.1]
#smb: \>
```

(***Question C.*** *Will the Linux machine (PC_2) appear in the list of domain computers on the Server_DC machine (click Start – Network) after we have connected to the shared resources via Samba?*)

and also to the folder located on the *PC_1* machine:

```
$ smbclient //PC_1.mycorp.org/Shared_PC -U nato
#Domain=[MYCORP] OS=[Windows 7 Ultimate 7600] Server=[Windows
7 Ultimate 6.1]
#smb: \>
```

(***Question D.*** *What setting allows access to the remote machine either by IP address or by its name when using the* **`smbclient`** *command?*)

c. Gain the same access using Kerberos ticket:

```
$ smbclient //Server_DC.mycorp.org/Shared_DC -k
#Domain=[MYCORP] OS=[Windows Server 2008 R2 Enterprise 7600]
Server=[Windows Server 2008 R2 Enterprise 6.1]
#smb: \>
```

Note: the same way we can access the shared folder located on the *PC_1* machine.

Question E. *After the commands*

```
$ kinit nato
$ smbclient //192.168.10.200/Shared_DC -U Administrator -k
```

get executed then what user will get connected to the shared resource?

3. In the Samba configuration file **`/etc/samba/smb.conf`** specify parameters listed below to make the *PC_2* machine acting as a domain member:

```
#=============== Global Settings ==============
[global]
   workgroup = MYCORP
   server string = Samba Server Version %v
   netbios name = MYCORP
   interfaces = lo eth0 192.168.10.0/24
   hosts allow = 127. 192.168.10.
#--------------- Logging Options --------------
   log file = /var/log/samba/log.%m
   max log size = 50
#-------- Standalone Server Options -----------
   security = user
   passdb backend = tdbsam
#------------- Domain Member Options ----------
   auth methods = winbind
   security = ADS
   realm = MYCORP.ORG
   password server = MYCORP.ORG
   encrypt passwords = yes
   winbind cache time = 300
   winbind enum users = yes
   winbind enum groups = yes
   winbind use default domain = yes
   winbind nested groups = yes
   winbind separator = /
   idmap uid = 1000-10000
   idmap gid = 1000-10000
   template shell = /bin/bash
   template homedir = /home/%D/%U
```

Note: to check up the **smb.conf** file for errors as well as to figure out the role status of the configured system use the command below:

```
$ testparm
#Load smb config files from /etc/samba/smb.conf
...
#Server role: ROLE_DOMAIN_MEMBER
...
```

4. Using the **winbind** service which is a component of the Samba suite of programs (**winbind** uses a UNIX implementation of Microsoft RPC calls, Pluggable Authentication Modules (PAMs), and the name service switch (NSS) to allow Windows NT domain users to appear and operate as UNIX users on a UNIX machine):

a. Install the **winbind** service package:

```
$ rpm -i samba-winbind-3.5.4-68.el6.i686
```

b. Add a word **winbind** to certain lines in the file **/etc/nsswitch.conf**:

```
passwd:    files winbind
shadow:    files winbind
group:     files winbind
```

c. Make the **winbind** service to start automatically after each reboot:

```
$ chkconfig winbind on
```

and start the service:

```
$ service winbind start
```

d. When the **winbind** service is running we can obtain the information about our domain, its users and groups, etc.:

Commands	Description
wbinfo -p	Check whether winbindd is alive
wbinfo -m	Get a list of trusted domains
wbinfo -i <username>	Get information about the user
wbinfo -u	Get the list of domain users
wbinfo -g	Get the list of domain groups
getent passwd	Get the list of local and domain users
getent group	Get the list of local and domain groups

5. The computer can be brought to the domain area using the **net ads** command.

a. To get the information about the domain use the command below:

```
$ net ads info
#LDAP server: 192.168.10.200
#LDAP server name: Server_DC.mycorp.org
#Realm: MYCORP.ORG
#Bind Path: dc=MYCORP,dc=ORG
#LDAP port: 389
#Server time: Sun, 29 May 2011 22:09:08 EDT
#KDC Server: 192.168.10.200
#Server time offset: -1
```

b. To join the domain use the following command:

```
$ net ads join –U Administrator
#Enter Administrator's password:
#Using short domain name – MYCORP
#Joined 'MYCORP' to realm 'mycorp.org'
#No DNS domain configured for mycorp. Unable to perform DNS
Update.
#DNS update failed!
```

Notes:

- the message of DNS Update has been displayed because the DNS server is specified manually (in the file **`/etc/sysconfig/network-scripts/ifcfg-eth0`**).
- after the 2nd command gets executed the Linux machine will appear in the list of domain computers (see *Start – Network* on the *Server_DC* machine).

Question F. *Why join a domain with the* **`net ads`** *command if the* **`smbclient`** *does not need it?*

6. Prepare a final lab report containing the following information:
    - answers to questions typed in *italic* (questions A-F);
    - explain the role of the **`winbind`** service;
    - the history of the user **`root`**;
    - your own conclusions.

APPENDICES

Appendix A. Linux Command-Line Basics

Table A-1 Common Linux console commands

Files Management	System Information
➥ `ls` – *directory listing* ➥ `ls -al` – *detailed directory listing with hidden files* ➥ `cd dir` – *change directory to* `dir` ➥ `cd ~` – *change to home directory* ➥ `pwd` – *display the current directory* ➥ `mkdir dir` – *create a directory* `dir` ➥ `rm file` – *delete a* `file` ➥ `rm -r dir` – *delete a directory* `dir` ➥ `rm -f file` – *force remove a* `file` ➥ `rm -rf dir` – *force remove a directory* `dir` ➥ `cp file1 file2` – *copy* `file1` *to* `file2` ➥ `cp -r dir1 dir2` – *copy* `dir1` *to* `dir2`*; create* `dir2` *if it doesn't exist* ➥ `mv file1 file2` – *rename or move* `file1` *to* `file2`*; if* `file2` *is an existing directory, it moves* `file1` *into directory* `file2` ➥ `ln -s file link` – *create a soft link* `link` *to the file* `file` ➥ `touch file` – *create or update* `file` ➥ `cat > file` – *redirect the standard output into* `file` ➥ `more file` – *display the contents of* `file` ➥ `head file` – *display the first 10 lines of* `file` ➥ `tail file` – *display the last 10 lines of* `file` ➥ `tail -f file` – *display the contents of* `file` *as it grows, starting with the last 10 lines*	➥ `date` – *show current date and time* ➥ `cal` – *display this month's calendar* ➥ `uptime` – *display current uptime* ➥ `w` – *display who is online* ➥ `whoami` – *who you are logged in as* ➥ `finger user` – *display information about* `user` ➥ `uname -a` – *show kernel information* ➥ `cat /proc/cpuinfo` – *display CPU information* ➥ `cat /proc/meminfo` – *display memory information* ➥ `man command` – *display the manual page for the specified* `command` ➥ `df` – *display disk usage* ➥ `du` – *display directory space usage* ➥ `fdisk -l` – *display disks partitions sizes and types (run as root)* ➥ `free` – *display memory and swap usage*
	Starting & Stopping
	➥ `halt` – *shutdown the system* ➥ `shutdown -h now` – *shutdown the system now and do not reboot* ➥ `shutdown -r 5` – *shutdown the system in 5 minutes and reboot* ➥ `shutdown -r now` – *shutdown the system now and reboot* ➥ `reboot` – *stop all processes and then reboot* ➥ `last reboot` – *display system reboot history* ➥ `runlevel` – *show current run-level* ➥ `startx` – *start the X system*

File Permissions	Installation

↳ `chmod octal file` – *change the permissions of* `file` *to* `octal`*, which are set up separately for user, group, and others by adding:*
- *4 – read (r) permission,*
- *2 – write (w) permission,*
- *1 – execute (x) permission,*
- *0 – no permission*
↳ `chown owner file` – *change the current owner of the* `file` *to* `owner`
↳ `chgrp group file` – *change the owner group of the* `file` *to* `group`
↳ `chmod +t dir` – *set the sticky bit to the directory* `dir`
↳ `chmod -t dir` – *remove the sticky bit from the directory* `dir`
↳ `chmod u+s file` – *set the SUID bit to the file* `file` *(use* `u-s` *to remove the SUID bit)*
↳ `chmod g+s file_or_dir` – *set the SGID bit to the object* `file_or_dir` *(use* `g-s` *to remove the SGID bit)*
	↳ `rpm -ivh package` – *install the rpm package called* `package`
	↳ `rpm -Uvh package` – *upgrade the rpm package called* `package`
	↳ `rpm -e package` – *delete the rpm package called* `package`
	↳ `rpm -ql package` – *list the files and state the installed version of the package called* `package`
	↳ `yum packagename install` – *install the specified package from the repository (the latest version of a package or group of packages while ensuring that all dependencies are satisfied will be installed)*
	↳ `yum packagename remove` – *remove the specified package as well as all the packages which depend on the package being removed*
	↳ *Install from the source codes:*
	`./configure`
	`make`
	`make install`

File Compression	**Process Management**
↳ `gzip file` – *compress* `file` *and rename it to* `file.gz` ↳ `gzip -d file.gz` – *decompress the file* `file.gz` *back to* `file` ↳ `tar -cf file.tar files` – *create a tar-file named* `file.tar` *that contains* `files` ↳ `tar -xf file.tar` – *extract the files from* `file.tar` ↳ `tar -czf file.tar.gz files` – *create a tar-file with Gzip compression* ↳ `tar -zxvf archive.tar.gz` *or* `tar -zxvf archive.tgz` – *decompress the files contained in the zipped and tarred archive called* `archive`	↳ `ps` – *show currently active processes* ↳ `top` – *display all running processes* ↳ `kill pid` – *stop the process that has* `pid` *as its id number* ↳ `kill -KILL pid` – *forcibly terminate the process that has* `pid` *as its id number* ↳ `killall proc` – *kill all processes named* `proc` ↳ `jobs` – *list all jobs* ↳ `bg` – *list stopped or background jobs; resume a stopped job in the background* ↳ `fg` – *bring the most recent job to the foreground; bring a specified job to the foreground*

Search Files and Strings	Network
↳ `find dir -name filepattern` – *searching for the files named like* `filepattern` *in the directory* `dir` ↳ `locate file` – *find all instances of* `file` ↳ `whereis app` – *display all possible locations of* `app` ↳ `which app` – *display which* `app` *will be run by default* ↳ `grep pattern files` – *search for the specified pattern in* `files` ↳ `grep -r pattern dir` – *search recursively for* `pattern` *in* `dir` ↳ `command | grep pattern` – *search for* `pattern` *in the output of* `command`	↳ `ping host` – *ping* `host` *(by IP address or domain name)* ↳ `netstat -vantup` – *display detailed information on network connections* ↳ `ifconfig` – *display current settings of network interfaces* ↳ `ifconfig nic IP netmask mask up` – *set the IP address* `IP` *and the subnetwork mask* `mask` *to the network interface named* `nic` ↳ `whois domain` – *get whois information for* `domain` ↳ `dig domain` – *get DNS information for* `domain`
User and Group Management	↳ `dig -x host` – *reverse lookup* `host`
↳ `useradd username` – *create a new user account named* `username` ↳ `userdel -r username` – *completely remove a user account* `username` ↳ `usermod username` – *modify a user account named* `username` ↳ `passwd username` – *set a new password for* `username` ↳ `groupadd groupname` – *create a new group account named* `groupname` ↳ `gpasswd groupname` – *set a new password for* `groupname` ↳ `su` – *login as root in current shell* ↳ `su username` – *switch user to* `username` *in the current shell* ↳ `su - username` – *switch user to* `username` *and load his environment* ↳ `exit` – *quit from a program or current shell* ↳ `sudo command` – *run* `command` *as root* ↳ `visudo` – *edit the read-only file /etc/sudoers*	↳ `wget file` – *download the file* `file` ↳ `wget -c file` – *continue a stopped download of the specified file* ↳ `ftp host` – connect to `host` via FTP ↳ `smbclient -L host -N` – *show all shared Samba resources of* `host` ↳ `smbclient ////host//shared -U smbperson` – *connect as* `smbperson` *to* `host` *for Samba resourse* `shared` ↳ `ssh user@host` – *connect to* `host` *as* `user` ↳ `ssh -p port user@host` – *connect to* `host` *on port* `port` *as* `user` ↳ `ssh-keygen -t dsa` – *generate a public key with the purpose to enable a keyed or passwordless login* ↳ `scp remoteLogin@remoteHost:file localDir` – *copy* `file` *from the remote host to the local directory* ↳ `scp file remoteLogin@remoteHost:remoteDir` – *copy* `file` *from the current local directory to the remote system*

Shortcuts	Services Management
↳ `Ctrl+C` – *halt the current command* ↳ `Ctrl+Z` – *stop the execution of the current command, resume with* **`fg`** *in the foreground or* **`bg`** *in the background* ↳ `Ctrl+D` – *log out of the current session, similar to exit* ↳ `Ctrl+W` – *erase one word in the current line* ↳ `Ctrl+U` – *erase the whole line* ↳ `Ctrl+R` – *bring up a recent command* ↳ `!!` – *repeat the last command* ↳ **`exit`** – *log out of the current session*	↳ **`/etc/init.d/`*`service`* `start|stop`** *or* **`service`** ***`servicename`*** **`start|stop`** *– start or stop the specified service* ↳ **`/etc/init.d/`*`servicename`* `status`** *or* **`service`** ***`servicename`*** **`status`** *– show the status of the service* ↳ **`/etc/init.d/`*`servicename`* `restart`** *or* **`service`** ***`servicename`*** **`restart`** *– restart the specified service* ↳ **`chkconfig --list [`*`servicename`*`]`** *– display the status list of service(s)* ↳ **`chkconfig --level 35`** ***`servicename`*** **`on|off`** *– set the specified service to autorun at start-up when a corresponding run-level is loading*

Table A-2 Common Linux Configuration Files

№	File	Description
1.	**`/boot/vmlinuz`**	the Linux kernel file (note: file naming conventions may include release information)
2.	**`/dev/hda`**	the device file for the 1st IDE hard drive on the system
3.	**`/dev/hdc`** **`/dev/cdrom`**	commonly, the IDE CDROM drive device file which often is a soft link to **`/dev/cdrom`** – the real CDROM driver file
4.	**`/dev/null`**	the device which contains nothing; sometimes it is used to send output to this device to make it go away forever
5.	**`/dev/sda`**	the device file for the 1st SATA or SCSI hard drive on the system
6.	**`/etc/aliases`**	contains aliases used by Sendmail and other mail transport agents
7.	**`/etc/bashrc`**	contains global defaults and aliases used by the bash shell
8.	**`/etc/crontab`**	the shell script to run commands periodically, it also invokes hourly, daily, weekly, and monthly scripts
9.	**`/etc/exports`**	contains a list of file systems which can get made available to other systems on the network via NFS
10.	**`/etc/fstab`**	the file system table; it contains the description of what disk devices are available at what mount points
11.	**`/etc/group`** **`/etc/gshadow`**	holds information regarding group accounts; secure group account information is kept in **`/etc/gshadow`**
12.	**`/etc/grub.conf`**	the grub boot loader configuration file
13.	**`/etc/host.conf`**	resolver configuration file; it contains the configuration information specific to the resolver library

14.	**/etc/hosts**	contains host names and their corresponding IP addresses used for name resolution whenever a DNS server is unavailable
15.	**/etc/hosts.allow**	contains a list of hosts allowed to access services on this computer
16.	**/etc/hosts.deny**	contains a list of hosts forbidden to access services on this computer
17.	**/etc/inittab**	describes how the INIT process should set up the system in various run-levels
18.	**/etc/issue**	contains the pre-login message, often overwritten by the **/etc/rc.d/rc.local** script in Red Hat and some other rpm-based Linux distributions
19.	**/etc/logrotate.conf**	stores configuration settings for rotation of system logs
20.	**/etc/modules.conf**	holds options for configurable system modules
21.	**/etc/motd**	this is the “message of the day” file which is printed upon login; it can be overwritten by **/etc/rc.d/rc.local** in Red Hat on start-up.
22.	**/etc/mtab**	status information for currently mounted devices and partitions
23.	**/etc/passwd** **/etc/shadow**	holds information regarding registered user accounts; password parameters and encrypted passwords themselves are kept in **/etc/shadow** for better security
24.	**/etc/sysconfig/network-scripts/ifcfg-eth?**	the configuration file for the network interface called **eth?**, where **?** stands for its number
25.	**/etc/profile**	contains global defaults for the bash shell
26.	**/etc/resolv.conf**	contains a list of domain name servers used by the local machine for name resolution
27.	**/etc/securetty**	contains a list of terminals where **root** can login
28.	**/proc/cpuinfo**	contains the CPU related information
29.	**/proc/filesystems**	holds the information regarding Linux file systems
30.	**/proc/interrupts**	stores the interrupts which are currently being used
31.	**/proc/ioports**	contains a list of the I/O addresses used by devices connected to the server
32.	**/proc/meminfo**	contains memory usage information for both physical memory and swap
33.	**/proc/modules**	holds currently loaded kernel modules
34.	**/proc/mounts**	holds currently mounted file systems
35.	**/proc/stat**	contains various statistics about the system, such as the number of page faults since the system was last booted
36.	**/proc/swaps**	holds the information on the utilization of the swap file
37.	**/proc/version**	contains the information about the version of the OS
38.	**/var/log/lastlog**	stores the information about the last boot process
39.	**/var/log/messages**	contains messages produced by the **syslog**/**rsyslog** service during the boot process
40.	**/var/log/wtmp**	is a binary data file that holds the login time and duration for the user who is currently on the system

Table A-3 Important Linux Directories

№	Directory	Description
1.	**`/bin/`**	contains all binaries needed for the boot process and for running the system in a single-user mode, including essential commands such as **`cd`**, **`ls`**, etc.
2.	**`/boot/`**	holds files used during the boot process
3.	**`/dev/`**	contains device files for all hardware devices on the system
4.	**`/etc/`**	contains files used by application subsystems such as mail, the Oracle database, etc.
5.	**`/etc/init.d/`**	contains various service start-up scripts
6.	**`/etc/profile.d/`**	holds application setup scripts run by **`/etc/profile`** upon login
7.	**`/etc/rc.d/`**	holds subdirectories which contain run-level specific scripts
8.	**`/etc/rc.d/init.d/`**	contains run-level initialization scripts
9.	**`/etc/rc.d/rc?.d/`**	contains soft links to scripts which are located in **`/etc/rc.d/init.d/`** for services to be started and stopped at the indicated run-level (there '?' stands for a number corresponding to the default run-level)
10.	**`/etc/skel/`**	contains files which will be copied in the new user's home directory, when a new user account is created
11.	**`/etc/X11/`**	contains subdirectories and configuration files for the X Window system
12.	**`/home/`**	contains user home directories
13.	**`/lib/`**	contains some shared library directories, files, and links
14.	**`/mnt/`**	it is the typical mount point for the user-mountable devices such as USB pen drives and CDROM
15.	**`/proc/`**	represents a virtual file system that provides system statistics; it doesn't contain real files but provides an interface to runtime system information
16.	**`/root/`**	it is a home directory for the root user
17.	**`/sbin/`**	contains system executable files that represent commands used by the privileged user for system administrative functions
18.	**`/tmp/`**	it is a standard location for temporary files created by applications and users
19.	**`/usr/`**	contains subdirectories with source code, programs, libraries, etc.
20.	**`/usr/bin/`**	contains commands available to normal users
21.	**`/usr/bin/X11/`**	contains X Window system binaries
22.	**`/usr/include/`**	holds include-files used in C programs
23.	**`/usr/share/`**	contains shared directories for man files, info files, etc.
24.	**`/usr/lib/`**	contains library files searched by the linker when programs are compiled
25.	**`/usr/local/bin/`**	contains common executable application files local to the system
26.	**`/usr/sbin/`**	contains commands used by the privileged user for system administrative functions
27.	**`/var/`**	contains administrative files such as log files, locks, spool files, and temporary files used by various utilities

Table A-4 FTP and Samba Console Commands

`?`	request help or the information about the FTP commands	
`ascii`	set the mode of file transfer to ASCII (this is the default and transmits seven bits per character)	
`binary`	set the mode of file transfer to binary (the binary mode transmits all eight bits per byte and thus provides less chance of a transmission error and must be used to transmit files other than ASCII files)	
`bye`	exit the FTP environment (same as `quit`)	
`cd`	change directory on the remote machine	
`close`	terminate a connection established with the remote computer	
	`close john`	close the current FTP connection with `john` (but the current user will be left within the FTP environment)
`delete`	delete (remove) a file in the current remote directory	
`get`	copy a file from the remote machine to the local machine	
	`get ABC DEF`	copy the file **ABC** from the current remote directory to a file named `DEF` in the current local directory.
	`get ABC`	copy the file **ABC** from the current remote directory to a file with the same name **ABC** in the current local directory
`help`	request a list of all available FTP commands	
`lcd`	change the directory on the current (local) machine	
`ls`	list the names of the files in the current remote directory	
`mkdir`	make a new directory within the current remote directory	
`mget`	copy multiple files from the remote machine to the local machine; a user will be prompted for a `y/n` answer before transferring each file	
	`mget *`	copy all the files from the current remote directory to the current local directory, using the same filenames
`mput`	copy multiple files from the local machine to the remote machine; a user will be prompted for a `y/n` answer before transferring each file	
`open`	to open a connection with another computer	
	`open john`	open a new FTP connection with **john** (a user must enter a username and password for **john**'s account or it will be an anonymous connection)
`put`	copy one file from the local machine to the remote machine	
`pwd`	find out the pathname of the current directory on the remote machine	
`quit`	exit the FTP environment (same as **bye**)	
`rmdir`	remove (delete) a directory in the current remote directory	

Appendix B. Service Profiles

№	Service	Daemon Name	Default Port(s)	Packages (*.rpm)	Configuration File(s)	Log File(s)	GUI
1.	**NFS** (Network File System)	**nfs**	2049	*nfs-utils*; *nfs-utils-lib*; *nfs4-acl-tools*	/etc/exports	/var/log/messages	system-config-nfs
2.	**Samba**	**smb**	137, 138, 139	*samba*; *samba-client*; *samba-common*	/etc/samba/smb.conf	/var/log/samba/%m.log	system-config-samba
3.	**FTP** (File Transfer Protocol)	**vsftpd**	20, 21	*vsftpd* (server); *ftp* (client)	/etc/vsftpd/vsftpd.conf /etc/vsftpd/ftpusers /etc/vsftpd/user_list	/var/log/vsftpd.log	-
4.	**SSH** (Secure Shell)	**sshd**	22	*openssh*; *openssh-client*; *openssh-server*; *openssh-askpass*	/etc/ssh/* $HOME/.ssh/*	/var/log/secure	-
5.	**SQUID**	**squid**	3128	*squid*	/etc/squid/*	/var/log/squid/*	-
6.	**Iptables**	**iptables** **iptables6**	-	*tcpwrappers*	/etc/sysconfig/iptables	/var/log/messages	system-config-firewall
7.	**DHCP** (Dynamic Host Configuration Protocol)	**dhcpd**	67 (server) 68 (client)	*dhcp*; *dhcp-client*	/etc/dhcpd.conf	/var/log/messages	-
8.	**Apache**	**httpd**	80 (http) 443 (https)	*httpd*; *httpd-devel*	/etc/httpd/conf/httpd.conf	/var/log/httpd/*	system-config-httpd
9.	**MySQL**	**mysqld**	3306	MySQL+PHP: *mysql*; *perl-DBD-MySQL*; *mysql-server*; *php-pdo*; *php-mysql*	/etc/my.cnf /etc/php.ini	/var/log/mysqld.log	PHPMyAdmin (third party software)

10.	**DNS** (Domain Name System)	**named** **rndc** 53 (DNS) 953 (rndc)	*bind*; *bind-utils*; *bind-libs*;	/etc/named.conf /var/log/messages system-config-bind
11.	**Sendmail**	**sendmail** 25 (SMTP) 110 (POP3) 143 (IMAP)	*sendmail*; *sendmail.cf*; *m4*	/etc/mail/sendmail.mc /var/log/maillog -
12.	**NTP** (Network Time Protocol)	**ntpd** 123	*ntp*	/etc/ntp.conf /var/log/messages -
13.	**CUPS** (Common Units Printing Service)	**cupsd** 631	*cupsys*; *cupsys-bsd*; *cupsys-client*; *foomatic-bin*	/etc/cups/cups.conf /etc/cups/printers.conf /var/log/messages system-config-printer
14.	**NIS** (Network Information Server)	**ypserv** **ypbind** dynamically assigned by rpcbind (portmap)	For server only: *ypserv*; For server and client: *ypbind*; *yp-tools*; *rpcbind* (or *portmap*); *nscd*	For server: /etc/yp.conf /etc/nsswitch.conf /var/yp/sequrenets /etc/sysconfig/network /var/yp/Makefile For client: /etc/yp.conf /etc/nsswitch.conf /var/log/messages For client: system-config-authentification
15.	**Kerberos**	**kerberos** 88	*krb5_libs*; *krb5_server*; *krb5_workstation*	/etc/krb5.conf /var/krb5/kdc.log /var/log/kadmind.log gnome-kerberos

Appendix C. Test Questions[1]

Pre-test

1. Linux is ...

   ○ a family of UNIX-like operating systems which use the Linux kernel
   ○ a family of UNIX-like operating systems which use the Minix kernel
   ○ a certain UNIX-like operating system which has many descendants known as Linux distributions (or flavors)
   ○ the first extension of UNIX (the original Linux is not used today but it has many actual descendants known as Linux distributions)

2. Make the right correspondence to the consequence **GNU – Linux – MINIX**:

   ○ Torvalds–Tanenbaum–Stallman ○ Torvalds–Stallman–Tanenbaum
   ○ Stallman–Torvalds–Tanenbaum ○ Stallman–Tanenbaum–Torvalds
   ○ Tanenbaum–Stallman–Torvalds ○ Tanenbaum–Torvalds–Stallman

3. What is SELinux?

   ○ An rpm-based Linux distribution
   ○ A debian-based Linux distribution
   ○ A secure shell (like bash or korn shell)
   ○ A set of modifications to provide a mechanism for supporting access control security policy

4. From the string **Red Hat–Fedora–openSUSE–Mageya–CentOS–Linspire** select the redundant object:

   ○ Red Hat ○ Fedora ○ openSUSE ○ Mageya ○ CentOS ○ Linspire

5. From the string **Debian–Ubuntu–Xubuntu–Knoppix–CentOS–Linspire** select the redundant object:

   ○ Debian ○ Ubuntu ○ Xubuntu ○ Knoppix ○ CentOS ○ Linspire

6. Which of the following statements is/are not true:

   □ BackTrack is Debian-based □ Knoppix is Debian-based
   □ CentOS is Debian-based □ Pentoo is Gentoo-based
   □ openSUSE is rpm-based □ Mandriva is rpm-based

7. Place the following pairs in the order by depending on the need (by default) to connect to the repository (first – repository is required; then – repository is not necessary, the software resource can be installed from a package):

   ○ rpm/dpkg, yum/apt ○ rpm/yum, dpkg/apt ○ dpkg/apt, rpm/yum
   ○ dpkg/yum, rpm/apt ○ yum/apt, dpkg/rpm ○ apt/rpm, yum/dpkg

8. Place the following pairs separately depending on the package manager type:

   ○ rpm/dpkg, yum/apt ○ rpm/yum, dpkg/apt
   ○ dpkg/yum, rpm/apt ○ yum/apt, dpkg/rpm

[1] The symbol ○ means that the question has the only true statement as the answer while the symbol □ signifies, that the question can have *one or more* true statements in its answer

9. Linux distributions can be installed onto:

□ personal computers □ mainframes □ supercomputers □ laptops
□ tablet computers □ mobile phones □ video and game consoles

10. The command `shutdown -r +2 poweroff` will produce one of the following:

○ the system will be rebooted in 2 minutes ○ a syntax error
○ the system will be powered off in 2 minutes
○ the system will be powered off in 2 minutes using recursive method of closing applications
○ the system will be rebooted in 2 minutes and the message `poweroff` will be sent to all connected users

11. The command `sh runme.sh` has been executed successfully as well as the command `sh /etc/init.d/rc.d/rc3.d/S99runme` (which is the soft link for the file `runme.sh`). But while rebooting the system the script `runme.sh` has not been executed. What may be possible reasons, if to reboot the system the command `init 3` was used?

□ instead of the soft link a hard link should be used
□ the script was created by a non-`root` user
□ the script does not have executable permissions
□ the `S99runme` file must end with the extension `.sh`
□ the default run-level is set to 5

12. To uninstall the package (say, the full package name is ***`xxx`***) from the system running *CentOS* operating system the following command should be used:

○ `rpm -i xxx` ○ `rpm -u xxx` ○ `apt-get clean xxx`
○ `dpkg -u xxx` ○ `dpkg - xxx` ○ `apt-get -u remove xxx`
○ the true command is not listed

13. To install the package (say, the full package name is ***`xxx`***) from the DVD onto the system running *Red Hat* operating system the following command should be used:

○ `rpm --install xxx` ○ `rpm --import xxx` ○ `apt-get install xxx`
○ `apt-get -i source xxx` ○ `yum install xxx` ○ `yum import xxx`
○ the true command is not listed

14. A tree-like structure of directories is known as:

○ filesystem ○ swap space ○ mount point table (MPT)
○ virtual filesystem ○ filesystem hierarchy standard (FHS)

15. The size of virtual memory is equal to:

○ the size of RAM ○ the size of swap space
○ the size of physical memory plus the size of swap space
○ the size of physical memory minus the size of swap space

16. The command to view a complete log of the installation process is:

○ `vi /var/log/install.log` ○ `nano /etc/install.log`
○ `tail /tmp/install.log` ○ `cat /root/install.log`

17. A swap space is used:

○ for data storage
○ to increase the amount of available memory
○ both to increase the amount of memory available and for data storage

18. What forms of swap space does Linux have?

○ the only form – a swap partition
○ the only form – a swap file
○ two forms – a swap partition and a swap file

19. Select the right statement:

○ It is possible to have on the system SEVERAL swap partitions or NO swap partitions at all
○ The ONLY ONE swap partition MUST exist on every system
○ At least ONE swap partition MUST exist on every system
○ Every system MUST have MORE than one swap partition

20. The right stage consequence of Linux boot process is as follows:

○ BIOS→MBR→Boot Loader→Kernel Loader→Kernel→
→init process→Start-up scripts
○ BIOS→MBR→Boot Loader→Kernel→init process→Start-up scripts
○ BIOS→MBR→Kernel Loader→init process→Start-up scripts
○ BIOS→MBR→Boot Loader→Kernel→Start-up scripts→init process
○ BIOS→MBR→Kernel→init process→Start-up scripts
○ BIOS→MBR→Kernel→Start-up scripts→init process

21. During the Linux boot process the power-on self test is performed:

○ exactly before the BIOS stage ○ within the BIOS stage
○ exactly after MBR stage ○ within the Boot Loader stage
○ exactly after Boot Loader stage

22. The “anatomy” of MBR (Master Boot Record) is as follows:

○ In bits: 446 (for boot loader), 64 (for partition table), 2 (for signature).
○ In bytes: 446 (for boot loader), 64 (for partition table), 2 (for signature).
○ In bits: 446 (for partition table), 64 (for boot loader), 2 (for signature).
○ In bytes: 446 (for partition table), 64 (for boot loader), 2 (for signature).
○ In bits: 446 (for boot loader), 32 (for partition table), 2 (for signature).
○ In bytes: 446 (for boot loader), 32 (for partition table), 2 (for signature).
○ In bits: 446 (for partition table), 32 (for boot loader), 2 (for signature).
○ In bytes: 446 (for partition table), 32 (for boot loader), 2 (for signature).

23. What statement(s) about MBR (Master Boot Record) is/are not true?

□ MBR is the very first sector of a computer's hard disk
□ MBR is the very first sector of each partition of a computer's hard disk
□ There may be many boot records, but MBR is always alone
□ MBR contains instructions how to load GRUB using a pre-selected OS

24. Select the wrong statements:

□ GRUB and LILO are both kernel loaders □ GRUB is not a boot loader
□ GRUB and LILO are both boot loaders □ GRUB is not a kernel loader
□ LILO is not a boot loader □ LILO is not a kernel loader
□ GRUB is a kernel loader and LILO is a boot loader
□ GRUB is a boot loader and LILO is a kernel loader

25. Modern Linux-based systems are designed in such a way that during the boot process:

○ GRUB is loaded after LILO ○ GRUB is loaded before LILO
○ GRUB is loaded instead of LILO ○ LILO is loaded instead of GRUB

26. Configuration files are usually stored in:

○ **`/bin`** ○ **`/boot`** ○ **`/etc`** ○ **`/lib`** ○ **`/root`** ○ **`/var`**

27. The default run-level is specified in:

○ **`/etc/fstab`** ○ **`/etc/grub.conf`** ○ **`/etc/inittab`** ○ **`/etc/boot.conf`**

28. To check the current run-level the following command should be used:

○ **`runlevel`** ○ **`pwd`** ○ **`whereis`** ○ **`locate`**
○ **`init`** (simply **`init`**, without parameters, not like **`init 3`** or **`init 5`**)

29. The directory **`/etc/rc.d/rc0.d/`** contains:

○ soft links to the scripts which will be executed before system is shutting down
○ scripts themselves which will be executed before system is shutting down
○ soft links to the scripts which will be executed before system is rebooting
○ the scripts themselves which will be executed before system is rebooting

30. If any user types the command **`ls -l /etc/rc.d/rc1.d/`** then the first part of each output string will look like:

○ `-rwxrwxrwx` ○ `-rwxrwxrwxt` ○ `lrwxrwxrwx` ○ `lrwxrwxrwxt`
○ `drwxrwxrwx` ○ `drwxrwxrwxt` ○ `crwxrwxrwx` ○ `crwxrwxrwxt`

Final Test

1. Suppose you are in the run-level 3. Which command must be used to get the scripts `/etc/rc.d/rc5.d/S*` executed if the default run-level is set to 3?
   ○ `reboot` ○ `shutdown now` ○ `shutdown -r now` ○ `shutdown +5`
   ○ `init 5` ○ `startx` ○ either `init 5` or `startx`

2. Two commands `cd /etc/rc.d/rc2.d/` and then `ls k*` were executed. Then:
   ○ the first command went successfully, the second one has risen an error
   ○ both commands went successfully, but no output has been displayed
   ○ both commands went successfully, some output lines have been displayed

3. Place the numbers of run-levels in the order corresponding to the sequence:
   - "Network-less mode",
   - "Reserved mode",
   - "Single user mode",
   - "Full multiuser GUI mode",
   - "Full multiuser console mode",
   - "Reboot",
   - "Poweroff".

   ○ 0 1 2 3 4 5 6 ○ 2 4 1 5 3 6 0 ○ 2 4 1 5 3 0 6
   ○ 4 2 1 5 3 6 0 ○ 4 2 1 5 3 0 6 ○ 2 4 1 3 5 6 0
   ○ 2 4 1 3 5 0 6 ○ 4 2 1 3 5 6 0 ○ 4 2 1 3 5 0 6

4. Which command should be executed to activate scripts contained in `/etc/rc.d/rc6.d/` ?
   □ `startx` □ `runlevel 6` □ `reboot`
   □ `halt` □ `poweroff` □ `shutdown now`

5. If a soft link to `startx` is created in the `/etc/rc.d/rc.3/` directory then the commands `init 5` and `startx` will become identically equal.
   ○ True ○ False

6. To reboot the system the following commands can be used:
   □ `shutdown –r` □ `reboot` □ `init 6` □ `telinit 6`
   □ `init 0` □ `telinit 0` □ `restart` □ `halt -r`

7. To power off the system the following commands can be used:
   □ `shutdown` □ `halt –p` □ `halt` □ `stopx` □ `init 0` □ `init 6`
   □ `poweroff` □ `telinit 0` □ `telinit 6` □ `shutdown –c` □ `shutdown -k`

8. The `init` process has:
   ○ `PID 0` and no `PPID` ○ `PID 1` and `PPID 0`
   ○ `PID 1` and no `PPID` ○ `PID 0` and `PPID 1`

9. Typing in the terminal the command **system-config-users &** will:

○ cause a syntax error

○ open the User Manager window but the terminal window will not be accessible until the opened window is closed

○ open the User Manager window and the terminal window will be accessible for typing other commands

10. The presence of the **&** sign at the end of a command means that the corresponding command will be executed:

○ in the background mode

○ in the foreground mode

○ with the higher priority for execution

○ with the lower priority for execution

11. Type the number of the redundant command from the listed below:

1. **kill -s KILL 2149**
2. **kill -KILL 2149**
3. **kill -SIGKILL 2149**
4. **kill -9 2149**
5. **kill 2149**

The correct answer is _______.

12. The execution of the command **pkill -9 -u john** by **root** will cause the following behavior:

○ the user account called **john** will be removed

○ each **john**'s process will be terminated, and the user **john** will be forcibly logged out

○ if the process with PID=9 has been created by the user **john**, then the mentioned process will be terminated

○ the process with PID=9 will be terminated on behalf of the user **john**, not on **root**'s behalf

13. What command is to be used to manage states of services at current boot?

○ **chkconfig** ○ **service** ○ **status** ○ **runlevel**

14. To configure services to be started automatically on different run-levels the following command should be used:

○ **chkconfig** ○ **service** ○ **status** ○ **runlevel**

15. The proper consequence of installing and setting up any service will be as follows:

○ **rpm -qa** …; **rpm -i** …; **chkconfig** … **on**; **cat** ; **service** … **start**

○ **rpm -qa** …; **rpm -i** …; **service** … **start**; **nano** …; **chkconfig** … **on**

○ **rpm -qa** …; **rpm -i** …; **chkconfig** … **on**; **service** … **start**; **gedit** …

○ **rpm -qa** …; **rpm -i** …; **vi** …; **service** … **start**; **chkconfig** … **on**

○ **rpm -qa** …; **rpm -e** …; **chkconfig** … **on**; **vim** …; **service** … **start**

16. To see whether the NFS service is running or not the following command should be used (choose the wrong answer):

○ **`service nfs status`** ○ **`/etc/init.d/nfs status`** ○ **`chkonfig --list nfs`**

17. To see the run-levels the NFS service will be started automatically on, the following command should be used (choose the right answer):

○ **`service nfs status`** ○ **`/etc/init.d/nfs status`**
○ **`chkonfig --list nfs`** ○ **`chkonfig --level nfs on`**

18. The command **`vmstat -f`** will:

○ flush the virtual memory
○ display the number of forks since boot
○ display the statistics of the virtual memory usage (full view)

19. The command **`fg %1`** will provide the following:

○ the first job will be brought to the foreground
○ the user called **`fg`** will be granted **`root`** privileges
○ the file called **`fg`** is a bash script which takes a value `1` as a parameter and the mentioned command will execute it

20. The command **`ls -l /home/john/abc`** has been executed and the following output has been displayed:

```
drw-rw-rw-t. 32 john staff 4096 Mar 16 00:26 /home/john/abc
```

Choose the right statement:

○ all users can read and modify files (in the case those files have proper permissions) within the directory **`abc`** but only **`root`** can delete them
○ all users can read and modify files (in the case those files have proper permissions) within the directory **`abc`** but only the users **`root`** and **`john`** can delete them
○ all users can read and modify files (in the case those files have proper permissions) within the directory **`abc`** but only the users **`root`** and **`john`** as well as members of the group staff can delete them
○ all users can read, modify and delete files (in the case those files have proper permissions) within the directory **`abc`**
○ the object **`abc`** is not a directory, it is a device file

21. The command **`ls -l /home/john/abc`** has been executed and the following output has been displayed:

```
-rwxrw-r--. 1 john staff 4096 Mar 16 00:26 /home/john/abc
```

How could it get reached?

○ **`chmod 421 /home/john/abc`** ○ **`chmod 764 /home/john/abc`**
○ **`chmod 752 /home/john/abc`** ○ **`chmod 642 /home/john/abc`**
○ **`chmod 762 /home/john/abc`** ○ **`chmod 761 /home/john/abc`**

22. The command **`ls -l /home/john/abc`** has been executed and the following output has been displayed:

```
-rwxrw-r-x. 1 john staff 4096 Mar 16 00:26 /home/john/abc
```

How could it get reached?

○ **`chmod 421 /home/john/abc`** ○ **`chmod 754 /home/john/abc`**
○ **`chmod 765 /home/john/abc`** ○ **`chmod 642 /home/john/abc`**
○ **`chmod 762 /home/john/abc`** ○ **`chmod 764 /home/john/abc`**

23. The command **`ls -l /home/john/abc`** has been executed and the following output has been displayed:

```
----------. 1 john staff 4096 Mar 16 00:26 /home/john/abc
```

How could it get reached?

○ **`su -; chmod u-rwx abc; chmod g-rwxt abc; chmod o-rwxt abc`**
○ **`su; chmod 0000 ~/abc`** ○ **`su - john; chmod 0 ~/abc`**
○ **`su staff; chmod a-rwx /home/john/abc`** ○ all answers are true

24. The command **`ls -l /etc/rc.d/rc3.d/`** has been executed and the lines with the same starting part have been displayed (dots in the string represent information which is different for different lines):

```
lrwxrwxrwx. 1 root root ………………………..
```

Continue the statement: ‘All contained objects are …’

○ regular files ○ device files ○ link files ○ directories ○ pipes

25. The user **`john`** has executed the command:

```
su -; echo 'alias "c=clear"' > .bashrc.
```

What does it mean?

○ The command **`c`** (as well as the command **`clear`**) will be executed successfully if they are typed by **`root`** only

○ The command **`c`** (as well as the command **`clear`**) will be executed successfully if they are typed by **`john`** only

○ The command **`c`** (as well as the command **`clear`**) will be executed successfully if they are typed by both **`root`** and **`john`**

○ The command **`c`** (as well as the command **`clear`**) will be executed successfully if they are typed by any user

○ The command **`c`** (but not the command **`clear`**) will be executed successfully if they are typed by **`root`** only

○ The command **`c`** (but not the command **`clear`**) will be executed successfully if they are typed by **`john`** only

○ The command **`c`** (but not the command **`clear`**) will be executed successfully if they are typed by both **`root`** and **`john`**

○ The command **`c`** (but not the command **`clear`**) will be executed successfully if they are typed by any user

○ After reboot the message `alias c=clear` will be displayed

26. If `root` runs the command `su - john; pwd` then what will be the output?

○ `/home/root` ○ `/` ○ `/root` ○ `~` ○ `/home/john`
○ The `root` user will be prompted to change `john`’s password

27. What will be the result of the `pwd` command if having this kind of prompt:

```
[root@debian ~]$
```

○ `/home/root` ○ `/` ○ `/root` ○ `/home/john` ○ `debian`

28. What does `./xxx` mean in general?

○ The file `xxx` is hidden
○ This is a soft link to the file `xxx`
○ This is the boot configuration file located in `/etc/init.d/` directory
○ This is a command which will try to execute the file `xxx`
○ The object `xxx` represents a system start-up script
○ The file `xxx` has executable permissions

29. What statement is true about `/.xxx` ?

○ The file `xxx` is hidden; it is located in the current directory (not necessarily in the top directory `/`);
○ The file `xxx` is hidden and it is located in the top directory `/`
○ This command will try to execute the file `xxx`
○ The object `xxx` represents a system start-up script
○ The file `xxx` has executable permissions

30. What will the command `cat ./xxx` do?

○ It will cause a syntax error
○ The content of the hidden file `xxx` will be displayed
○ The file `xxx` is not hidden, it is located in the current directory and the `cat` command will display the content of `xxx`
○ The file `xxx` has executable permissions and the `cat` command will display its content

31. Which of the following command strings will be executed by `Cron` service every Monday?

○ `1 * * * * echo “Hello, world!”` ○ `* 1 * * * echo “Hello, world!”`
○ `* * 1 * * echo “Hello, world!”` ○ `* * * 1 * echo “Hello, world!”`
○ `* * * * 1 echo “Hello, world!”` ○ The true command is not listed

32. The service which generates log strings of system events is called:

○ `auditd` ○ `rsyslogd` ○ `crond` ○ `named` ○ `bind`

33. The service which generates log strings of user specified events is called:

○ `auditd` ○ `rsyslogd` ○ `crond` ○ `named` ○ `bind`

34. Which command will set an audit watch on a particular file?
○ **auditctl** ○ **aureport** ○ **ausearch** ○ **autrace** ○ **audispd**

35. The **root** shell can be completely disabled.
○ True ○ False

36. The command **usermod -G wheel john** will:
○ grant all administrative privileges to the user called **john**
○ grant to the user called **john** the ability to use the **su** command
○ grant to the user called **john** the ability to use the **sudo** command
○ grant to the user called **john** the ability to use both **su** and **sudo** commands
○ simply add a user called **john** to a group called **wheel**

37. To display the available disk space the following command is used:
○ **ls** ○ **du** ○ **df** ○ **dd** ○ **fsck** ○ **fdisk** ○ **sfdisk**

38. What does the command **du -b *** do?
○ It counts the size of all files located in the current directory (in bytes)
○ It counts the size of all files located in the current directory (in blocks)
○ It displays the free space on every partition (in bytes)
○ It displays the free space on every partition (in blocks)

39. To prevent **root** from logging into the system remotely via SSH the following file must be edited:
○ **/etc/sudoers** ○ **/etc/ssh/sshd_config** ○ **/etc/securetty**

40. The **visudo** command allows to:
○ edit a read-only file **/etc/sudoers**
○ edit a read-only file **/etc/securetty**
○ run any commands which require administrative privileges
○ edit the file **/etc/sudoers** as well as the command **vi /etc/sudoers**
○ edit the file **/etc/securetty** as well as the command **vi /etc/securetty**

41. The purpose of **/etc/securetty** file is:
○ to disable **root** logins to certain devices (e.g. to the console)
○ to allow users listed there to use the **su** command
○ to deny users listed there to use the **su** command
○ to allow users listed there to use the **sudo** command
○ to deny users listed there to use the **sudo** command
○ to prevent users from logging into the system via SSH

42. The line `root:::root` is present in:
○ **/etc/passwd** ○ **/etc/shadow** ○ **/etc/group** ○ **/etc/gshadow**

43. The line `root:x:0:root` is present in:
○ **/etc/passwd** ○ **/etc/shadow** ○ **/etc/group** ○ **/etc/gshadow**

44. The line `root:x:0:0:root:/root:/bin/bash` is present in:
○ **`/etc/passwd`** ○ **`/etc/shadow`** ○ **`/etc/group`** ○ **`/etc/gshadow`**

45. The line `general:!!:genboss:john,michael` may be present in:
○ **`/etc/passwd`** ○ **`/etc/shadow`** ○ **`/etc/group`** ○ **`/etc/gshadow`**

46. If **`smbperson`** is the user account created with the help of two commands
`useradd smbperson` and **`smbpasswd`**
then the line like `smbperson:!!:15041:0:99999:7:::` is mentioned in:
○ **`/etc/passwd`** ○ **`/etc/shadow`** ○ **`/etc/group`** ○ **`/etc/gshadow`**

47. If the **`root`** user has executed the command **`userdel john`** then:
○ only **`root`** can access the directory **`/home/john`**
○ any user can access the directory **`/home/john`**
○ the directory **`/home/john`** will not exist

48. What will do the following command: **`gpasswd -a john walley`** ?
○ Create a user called **`john`** and add him to a group called **`walley`**
○ Add a user called **`john`** to a group called **`walley`**
○ Set up a password ‘`john`’ for a group called **`walley`**
○ Set up a password ‘`walley`’ for a group called **`john`**
○ Set up a password ‘`walley`’ for a user called **`john`**

49. What will be the result of the following command:

```
useradd -g math -G bio,phys -s /bin/tcsh -c "John Connor" -m john
```

○ Groups **`bio`** and **`phys`** are the primary ones for the user **`john`**,
`/bin/tcsh` is the default shell for the user **`john`**,
`john` is the member of the group **`math`**
○ The group **`math`** is the primary one for the user **`john`**,
`/bin/tcsh` is the home directory of the user **`john`**,
`john` is the member of groups **`bio`** and **`phys`**
○ Groups **`bio`** and **`phys`** are the primary ones for the user **`john`**,
`/bin/tcsh` is the home directory of **`john`**,
`john` is the member of the group **`math`**
○ The group **`math`** is the primary group for the user **`john`**,
`/bin/tcsh` is the default shell for **`john`**,
`john` is the member of groups **`bio`** and **`phys`**
○ The group **`math`** is the primary one for the user **`john`**,
`/bin/tcsh` is the home directory of **`john`**,
`john` is the member of the group **`math`** as well as users **`bio`** and **`phys`**

50. It isn’t possible to assign more than one IP address to one Ethernet card.
○ True ○ False

51. When editing the file **`/etc/udev/rules.d/70-persistent-net.rules`** the following can be achieved:

- □ the list of all attached devices can be edited
- □ the default number of available virtual consoles can be changed
- □ names of the network interface cards can be changed
- □ the rules referred to Internet access can be set up or changed
- □ the rules referred to access within the certain LAN can be set up or changed

52. The command `ls /etc/sysconfig/network-scripts/ifcfg-eth0:0` will:

- ○ cause a syntax error
- ○ display the content of the specified file if it exists
- ○ display the only string `/etc/sysconfig/network-scripts/ifcfg-eth0:0` if the file exists
- ○ display the detailed string of the file **`/etc/sysconfig/network-scripts/ifcfg-eth0:0`** if this file exists
- ○ create a soft link to the specified file

53. Which command(s) will provide an error when being executed even by **`root`** if the system has two NICs called `eth0` and `eth1` and they both are down?

- □ **`ifconfig`**
- □ **`ifconfig lo down`**
- □ **`ifconfig eth1:0 192.168.10.12  mask 255.255.255.0 up`**
- □ **`ifconfig eth0:0 192.168.10.12 netmask 255.255.0.0`**
- □ **`ifconfig eth0 192.168.10.12 mask 255.255.255.0 up`**
- □ **`ifconfig eth1 192.168.10.12 broadcast 255.255.255.0 up`**
- □ **`ifconfig eth0:192.168.10.12 192.168.10.14 up`**

54. Having the network interface `eth0` the line `GATEWAY=xxx.xxx.xxx.xxx` should be placed into:

- ○ **`/etc/hosts`**
- ○ **`/etc/resolve.conf`**
- ○ **`/etc/sysconfig/network-scripts/ifcfg-eth0`**
- ○ **`/etc/sysconfig/network`**

55. Having the network interface `eth0` the line `DNS1=xxx.xxx.xxx.xxx` should be placed into:

- ○ **`/etc/hosts`**
- ○ **`/etc/resolve.conf`**
- ○ **`/etc/sysconfig/network-scripts/ifcfg-eth0`**
- ○ **`/etc/sysconfig/network`**

56. The directory **`/tmp/nfs`** should be shared within the whole network via NFS for read and write remote access. What actions are necessary?

- □ Executing the command **`chmod o+rw /tmp/nfs`**
- □ The string `/tmp/nfs *(rw)` should be placed into **`/etc/exports`**
- □ Executing the command **`exportfs –vu`**
- □ Mounting **`/tmp/nfs`** directory to the empty directory on the client machine

57. The directory **`/tmp/nfs`** should be shared within the whole network via NFS for read-only remote access. What actions of listed below are necessary?

- □ Executing the command **`chmod o+r /tmp/nfs`**
- □ The string `/tmp/nfs *(ro)` should be placed into **`/etc/exports`**
- □ Executing the command **`exportfs –a`**
- □ Mounting **`/tmp/nfs`** to the empty directory on the server machine

58. The directory **`/tmp/nfs`** is shared via NFS from the host `192.168.10.1` to the host `192.168.10.21`. What action is required to get the access to **`/tmp/nfs`**?

- ○ Executing the following command on the host `192.168.10.21`:
  `mount -t nfs 192.168.10.1:/path_to_shared_resource /mnt/nfs`
- ○ Executing the following command on the host `192.168.10.1`:
  `mount -t nfs 192.168.10.21:/path_to_shared_resource /mnt/nfs`
- ○ Executing the following command on the host `192.168.10.21`:
  `mount -t nfs 192.168.10.21:/path_to_shared_resource /mnt/nfs`
- ○ Executing the following command on the host `192.168.10.1`:
  `mount -t nfs 192.168.10.1:/path_to_shared_resource /mnt/nfs`

59. The directory **`/tmp/nfs`** is shared via NFS from the host `20.20.10.1` to hosts `20.20.0.0/255.255.0.0`. Which attempts to use **`/tmp/nfs`** will be successful?

- □ Mapping **`/tmp/nfs`** as a network drive from `20.20.0.21` if it is running Windows 7.
- □ Mapping **`/tmp/nfs`** as a network drive from `20.20.10.21` if it is running Windows Server 2008.
- □ Mounting **`/tmp/nfs`** on `20.20.0.25` if it is running CentOS.
- □ Mounting **`/tmp/nfs`** on `20.20.10.21` if it is running Red Hat.

60. A directory **`/tmp/public`** has been exported from `192.168.10.1` (the NFS server) and mounted into **`/mnt/nfs`** on `192.168.10.21` (the NFS client). Everything worked well until the client system got rebooted. Adding a line `192.168.10.1:/tmp/public /mnt/nfs nfs defaults 0 0` to the file **`/etc/fstab`** on the *client machine* did not help: again the same problem appeared after rebooting the client machine. What may be possible reasons?

- □ There is a syntax error in the line that was added to the file **`/etc/fstab`**
- □ The client machine has lost its IP address after reboot
- □ This line should be added to **`/etc/mtab`** file, not to **`/etc/fstab`**
- □ This line should be added to **`/etc/inittab`** file, not to **`/etc/fstab`**
- □ This line should be added to **`/etc/fstab`** file on the server machine
- □ To get the directory mounted by the means of **`/etc/fstab`** file we should be always logged into the client system as a root user
- □ The **`iptables`** service is stopped

61. In the `/etc/exports` file there is the line

```
/home/nato 192.168.0.0/255.255.255.0(ro)
```

This means that the directory `/home/nato` is shared with:

○ the only host `192.168.0.0` (which belongs to the class C network)
○ the range of hosts having IP addresses from `192.168.0.0` to `255.255.255.0`
○ all hosts within the class C network `192.168.0.xxx`
○ all hosts within the class B network `192.168.xxx.xxx`

62. To get certain directories shared with Linux-based systems only the following file should be edited:

○ `/etc/hosts` ○ `/etc/inittab` ○ `/etc/samba/smb.conf`
○ `/etc/resolv.conf` ○ `/etc/exports` ○ the true answer is not listed

63. To get certain directories shared with both Linux-based and Windows-based systems the following file should be edited:

○ `/etc/hosts` ○ `/etc/inittab` ○ `/etc/samba/smb.conf`
○ `/etc/resolv.conf` ○ `/etc/exports` ○ the true answer is not listed

64. To check whether the web server service is installed on the system running Red Hat Enterprise Linux operating system the following command should be used:

○ `chkconfig --list ypserv` ○ `chkconfig --level http`
○ `rpm -qa | grep httpd` ○ `rpm -qa | grep apache2`
○ `service apache status` ○ `service squid status¶`

65. To correct work of the NFS service requires the following ports to be open (sort them in the direct chronological order by the time to get open):

○ 111, 2049 ○ 2049, 111 ○ 2049, 111, 53
○ 2049, 53, 111 ○ 111, 953, 2049 ○ 111, 2049, 953

66. In the active FTP type:

○ an FTP server initiates a data transfer connection back to the client and uses the port 20 as the source port and a high port as the destination port
○ an FTP server initiates a data transfer connection back to the client and uses the port 21 as the source port and a high port as the destination port
○ an FTP server initiates a data transfer connection back to the client and uses high ports as the source and destination ports
○ a data transfer connection back to the client is initiated by the FTP client and data is transmitted to the FTP client through the port 20
○ a data transfer connection back to the client is initiated by the FTP client and data is transmitted to the FTP client through the port 20
○ a data transfer connection back to the client is initiated by the FTP client and data is transmitted to the FTP client through the port 20

67. In the passive FTP type:

○ an FTP server initiates a data transfer connection back to the client and uses the port 20 as the source port and a high port as the destination port
○ an FTP server initiates a data transfer connection back to the client and uses the port 21 as the source port and a high port as the destination port
○ an FTP server initiates a data transfer connection back to the client and uses high ports as the source and destination ports
○ a data transfer connection back to the client is initiated by the FTP client and data is transmitted to the FTP client through the port 20
○ a data transfer connection back to the client is initiated by the FTP client and data is transmitted to the FTP client through the port 21
○ a data transfer connection back to the client is initiated by the FTP client and data is transmitted to the FTP client through one of the high ports

68. If using the **`vsftpd`** service then the default FTP directory of the user called **`ftpuser`** will be the following one:

○ **`/var/ftp/pub`** ○ **`/home`** ○ **`/home/ftpuser`** ○ **`/tmp/public`** ○ **`/tmp`**

69. If in the **`vsftpd.conf`** it is set `userlist_enable=YES` and `userlist_deny=NO` but **`ftpuser`** is mentioned only in **`/etc/user_list`** then:

○ **`ftpuser`** can connect via FTP ○ **`ftpuser`** cannot connect via FTP

70. If in the **`vsftpd.conf`** it is set `userlist_enable=YES` and `userlist_deny=YES` but **`ftpuser`** is mentioned only in **`/etc/user_list`** then:

○ **`ftpuser`** can connect via FTP ○ **`ftpuser`** cannot connect via FTP

71. If in the **`vsftpd.conf`** it is set `userlist_enable=NO` and **`ftpuser`** is mentioned only in **`/etc/user_list`** then:

○ **`ftpuser`** can connect via FTP ○ **`ftpuser`** cannot connect via FTP

72. Is it possible to connect to the remote system via SSH without typing passwords?

○ Yes, as an anonymous user
○ Yes, but both parties must exchange their cryptographic keys
○ No, the authentication is always required when connecting via SSH

73. Is FTP-connection possible without typing passwords from the client side?

○ Yes, as an anonymous user but with empty password
○ Yes, but both parties must exchange their cryptographic keys
○ No, the authentication is always required when connecting via FTP

74. To access resources shared via Samba through the terminal window the FTP commands (like **`put`** or **`get`**) can be used.

○ True ○ False

75. There are two machines with the IP addresses `192.168.10.1` and `192.168.10.21`. On the first machine the command **`ping 192.168.10.21`** went successfully, but on the second machine the command **`ping 192.168.10.1`** did not. What may be the possible reasons?

□ These two machines have different network masks
□ The firewall of the `192.168.10.1` host blocks incoming **`icmp`** packages
□ The firewall of the `192.168.10.21` host blocks incoming **`icmp`** packages
□ The firewall of the `192.168.10.21` host blocks outgoing **`icmp`** packages
□ The firewall of the `192.168.10.1` host blocks outgoing **`icmp`** packages

76. The **`vsftpd`** package must be installed onto:

○ the FTP-server and it is also required on the FTP-client system
○ the FTP-server system only
○ the FTP-client system only
○ the FTP-server and it may be installed (but not necessarily) on the FTP-client system
○ the FTP-client and it may be installed (but not necessarily) on the FTP-server system

77. The **`ftp`** package must be installed onto:

○ the FTP-server and it is also required on the FTP-client system
○ the FTP-server system only
○ the FTP-client system only
○ the FTP-server and it may be installed (but not necessarily) on the FTP-client system
○ the FTP-client and it may be installed (but not necessarily) on the FTP-server system

78. Running the machine with the IP address `192.168.10.21` the user **`john`** executed the command **`ftp john@192.168.10.1`** and got an FTP-access. Choose the right statement:

○ There is a user account called **`john`** on `192.168.10.21` machine and on the host `192.168.10.1` the same account may exist (but not necessary)
○ There is a user account called **`john`** on `192.168.10.21` machine and on the host `192.168.10.1` the same account necessarily exists
○ There is a user account called **`john`** on `192.168.10.1` machine and on the host `192.168.10.21` the same account may exist (but not necessary)
○ There is a user account called **`john`** on `192.168.10.1` machine and on the host `192.168.10.21` the same account necessarily exists

79. The default directory for anonymous FTP-users is:

○ **`/home/anonymous`** ○ **`/home/nobody`** ○ **`/tmp`** ○ **`/tmp/public`**
○ **`/tmp/ftp`** ○ **`/var/pub`** ○ **`/var/ftp/pub`**

80. The default directory for the authorized FTP-user **`john`** is:

○ **`/home/john`** ○ **`/home/nobody`** ○ **`/tmp/public/john`**
○ **`/tmp/ftp/john`** ○ **`/var/pub/john`** ○ **`/var/ftp/pub/john`**

81. The Samba service can be used on the system to make it:

□ a file server □ a print server □ DNS server
□ a primary/backup domain controller server

82. The remote access via Samba can be restricted by:

□ number of connections □ users
□ shared source permissions □ hosts

83. In order to get a remote access to resources shared via Samba the command `smb://192.168.10.1/` should be typed in:

□ login shell □ interactive shell □ browser
□ nowhere, because the command is not valid

84. If the DHCP server is used within the network then the line `BOOTPROTO=none` should be placed into **`/etc/sysconfig/network-scripts/ifcfg-eth0`** located on:

○ the DHCP-server system ○ the DHCP-client system
○ the DHCP-server system which necessarily holds a domain
○ the DHCP-client system which necessarily is a domain member
○ this is required on both on the DHCP-server and DHCP-client systems

85. If `eth0` is the name of the installed NIC and the DHCP server is used within the network the line `BOOTPROTO=dhcp` should be placed into:

○ **`/etc/sysconfig/network`** ○ **`/etc/hosts`**
○ **`/etc/sysconfig/network-scripts/ifcfg-eth0`** ○ **`/etc/resolv.conf`**

86. Setting up the Sendmail service what is the right sequence of commands?

○
```
chkconfig sendmail on
m4 /etc/mail/sendmail.cf > /etc/mail/sendmail.mc
vim /etc/mail/sendmail.mc
service sendmail start
mail -v root@travel.biz
```

○
```
service sendmail start
vim /etc/mail/sendmail.mc
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
chkconfig sendmail on
mail -v root@travel.biz
```

○
```
vim /etc/mail/sendmail.cf
m4 /etc/mail/sendmail.cf > /etc/mail/sendmail.mc
chkconfig sendmail on
service sendmail start
mail -v root@ travel.biz
```

○
```
chkconfig sendmail on
vim /etc/mail/sendmail.mc
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
service sendmail start
mail -v root@travel.biz
```

87. The letters `dnl` in the configuration file of the Sendmail service are:
○ the abbreviation ○ used as one-line comment signs ○ a protocol name

88. SMTP, POP3, IMAP have the following port numbers (correspondingly):
○ 110, 25, 143 ○ 110, 25, 443 ○ 25, 110, 143 ○ 25, 110, 443

89. The port number 3128 is used by default by:
○ a web-service ○ a proxy service ○ a DNS service

90. Place the following port numbers in the order that corresponds to a string "**telnet; SSH; FTP**":
○ 20; 21; 22, 23 ○ 21; 20; 22, 23 ○ 22; 23; 21, 22
○ 22; 23; 20, 21 ○ 23; 22; 20, 21 ○ 23; 20; 21, 22

91. What configuration file the following lines should be placed into?

```
search travel.biz
nameserver 192.168.10.1
```

○ **`/etc/hosts`** ○ **`/etc/sysconfig/network`** ○ **`/etc/resolv.conf`**
○ **`/etc/sysconfig/network-scripts/ifcfg-eth0`**

92. The Domain Name System service is called:
○ `bind` ○ `named` ○ `dnsd` ○ `squid` ○ `rndcd`

93. If the Domain Name System service is not installed then to imitate this the following file may be used:
○ **`/etc/hosts`** ○ **`/etc/sysconfig/network`** ○ **`/etc/resolv.conf`**
○ **`/etc/sysconfig/network-scripts/ifcfg-eth0`**
○ There is no right configuration file in the list
○ The DNS service cannot be imitated, it must be installed and configured

94. The **`A`** records in the configuration files of the Domain Name System service stand for:
○ aliases (nicknames) for web-sites
○ mailing mechanism
○ mapping from hostnames to IP addresses
○ reverse mapping from IP addresses to hostnames
○ identifying the servers that are authoritative for a zone

95. The **`PTR`** records in the configuration files of the Domain Name System service stand for:
○ aliases (nicknames) for web-sites
○ mailing mechanism
○ mapping from hostnames to IP addresses
○ reverse mapping from IP addresses to hostnames
○ identifying the servers that are authoritative for a zone

96. The **NS** records in the configuration files of the Domain Name System service stand for:

○ aliases (nicknames) for web-sites
○ mailing mechanism
○ mapping from hostnames to IP addresses
○ reverse mapping from IP addresses to hostnames
○ identifying the servers that are authoritative for a zone

97. The **CNAME** records in the configuration files of the Domain Name System service stand for:

○ aliases (nicknames) for web-sites
○ mailing mechanism
○ mapping from hostnames to IP addresses
○ reverse mapping from IP addresses to hostnames
○ identifying the servers that are authoritative for a zone

98. Each domain zone may have more than exactly one **SOA** record.

○ True ○ False

99. Choose the right statement about the files **~/.bashrc** and **~/.bash_profile**:

○ **~/.bashrc** is responsible for the login shell while **~/.bash_profile** refers to the interactive shell
○ **~/.bashrc** is responsible for the interactive shell while **~/.bash_profile** refers to the login shell
○ **~/.bashrc** is responsible both for the login and interactive shells while **~/.bash_profile** refers to the interactive shell only
○ **~/.bashrc** is responsible both for the login and interactive shells while **~/.bash_profile** refers to the login shell only
○ **~/.bashrc** is responsible for the login shell while **~/.bash_profile** refers to both the login and interactive shells
○ **~/.bashrc** is responsible for the interactive shell while **~/.bash_profile** refers to both the login and interactive shells

100. Choose the right statements about the Kerberos scheme.

Note:

- KAS stands for Kerberos Authentication Server,
- TGS stands for Ticket Granting Server,
- AS stands for Application Server

○ User requests TGS for KAS ticket
TGS gives a ticket for KAS to the client
User requests KAS for Server ticket
KAS gives a ticket for Server to the client
User requests AS for service

○ User requests TGS for Server ticket
TGS gives a ticket for Server to the client
User requests KAS for TGS ticket
KAS gives a ticket for TGS to the client
User requests AS for service

○ User requests KAS for TGS ticket
KAS gives a ticket for TGS to the client
User requests TGS for Server ticket
TGS gives a ticket for Server to the client
User requests AS for service

RECOMMENDED READING

1. Kemp J. Linux System Administration Recipes: A Problem-Solution Approach. – Springer-Verlag New York, 2009. – 260 p.
2. Kroah-Hartman G. Linux Kernel in a Nutshell O'Reilly Media, 2006. – 202 p.
3. Maurice J. Bach. The Design of the UNIX Operating System. – Prentice Hall, 1986. – 471 p.
4. Negus C., Weeks T. Linux Troubleshooting Bible. – John Wiley & Sons, 2004. – 598 p.
5. Olifer N., Olifer V. Computer Networks: Principles, Technologies and Protocols for Network Design. – John Wiley & Sons, 2004. – 973 p.
6. Petersen R. Linux: The Complete Reference, Sixth Edition. – The McGraw Hill Companies, 2008. – 830 p.
7. Silberschatz. A, Galvin P.B., Gagne G. Operating System Concepts. 8th Edition. – John Wiley & Sons, 2009. – 975 p.
8. West M. System Administration. – The Shuttleworth Foundation, 2004. – 174 p.