Exercise 3
Linux installation

Krzysztof Palka

April 26, 2013

Abstract

This report presents process of installation and configuration of Linux system on virtual machine.

1 Introduction

I will use VirtualBox program as virtual machine and Fedora 32-bit Linux distribution. VirtualBox packages are are available at `https://www.virtualbox.org/wiki/Downloads` and List of Fedora packages here: `http://mirrors.fedoraproject.org/publiclist/Fedora/18/`. I'm downloading i386 distribution which stands for 32 bit version.

2 Preparation of virtual machine

I'm installing VirtualBox and creating new machine with following parameters

Name Fedora

Memory 768 MB

Disk file type VDI

Disk storage details Dynamically allocated

Disk size 10 GB

I'm setting downloaded iso file as start disk and starting machine.

3 Installation

I'm choosing *Install Fedora*. During verification of installation media I'm pressing *Esc*, to skip this proces. I'm choosing default language and going to partitioning section by clicking on disk icon. As *Partition type* I'm choosing *Standard Partition* and selecting *Let me customize the partitioning of the disks instead*. I'm adding new partition—root—by typing */* as *Mount Point* and *2GB* as *Desired capacity*. In *customize* section I'm choosing:

Device Type Standard Partition

File System ext4

And applying changes. In analogous way I'm adding following partitions:

/boot 500 MB

/home 500 MB

/tmp 500 MB

/usr 4 GB

/var 2 GB

And finally I'm adding swap partition for remaining space by typing *swap* in *Mount Point* and leaving *Desired capacity* field empty. I'm clicking *Finish Partitioning* and then *Software selection*. As environment I'm choosing *Minimal Install*, clicking *Done* and *Begin Installation*. I'm setting root password and finishing configuration and clicking *reboot*. After reboot of VM I'm unchecking Fedora installation iso file (Devices → CD/DVD) and rebooting it again.

When system boot I'm logging in as *root* and typing following commands:

```
1 cat /proc/mounts
2 tune2fs -c 10 -i 2m -e remount-ro /dev/sda1
3 tune2fs -c 10 -i 2m -e remount-ro /dev/sda2
4 tune2fs -c 10 -i 2m -e remount-ro /dev/sda3
5 tune2fs -c 10 -i 2m -e remount-ro /dev/sda4
6 tune2fs -c 10 -i 2m -e remount-ro /dev/sda5
7 tune2fs -c 10 -i 2m -e remount-ro /dev/sda6
8 tune2fs -c 10 -i 2m -e remount-ro /dev/sda7
9 tune2fs -l /dev/sda1
```

cat /proc/mounts shows current mount options. tune2fs is used to adjust tunable filesystem parameters on filesystems. I used following arguments:

-c 10 Adjust the maximal mounts count between two filesystem checks to 10.

-i 2m Adjust the maximal time between two filesystem checks to 2 months.

-e remount-ro Change the behavior of the kernel code when errors are detected. Remount filesystem read-only.

/dev/sda*N* Device name.

-l List the contents of the filesystem superblock (To check).

4 Basic Configuration

According to informations placed in file /etc/inittab system uses 'tagets' instead of runlevels, so I'm changing default runlevel by taping ln -s /lib/systemd/system/<target name>.target /etc/systemd/system/default.target, where <target name> could be multi-user for runlevels 3 or graphical for runlevel 5.

```
10 ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/
       default.target
```

To check all groups on system

```
11 cat /etc/group
```

To add new group

```
12 groupadd students
```

To create new user, and add him to groups: [username], wheel (for possibility to use sudo), users and students.

```
13 adduser -U -G wheel,users,students krzysiek
14 passwd krzysiek
15 id krzysiek # to check
```

To configure DHCP Client

```
16 echo 'NETWORKING=yes' > /etc/sysconfig/network
17 echo 'DEVICE=p2p1' > /etc/sysconfig/network-scripts/ifcfg-p2p1
18 echo 'BOOTPROTO=dhcp' >> /etc/sysconfig/network-scripts/ifcfg-
       p2p1
19 echo 'ONBOOT=yes' >> /etc/sysconfig/network-scripts/ifcfg-p2p1
```

To add users tytus, romek and atomek

```
useradd -G users tytus # add user to group users
useradd -G users romek
useradd -G users atomek
passwd tytus # set password
passwd romek
passwd atomek
passwd -x 30 tytus # set expiration date of password
passwd -x 30 romek
passwd -x 30 atomek
chage -l romek # check password expiration date
```

To add temporary group *operators* and add user *romek* to this group

```
groupadd operators
gpasswd -a romek operators
groups romek # to check if user romek belongs to group operators
```

To create variable *LOGINTIME* available for every user (*export* makes the variable available to sub-processes)

```
echo "export LOGINTIME=\"$(date -u +'%T')\"" >> /etc/profile
logout #then login, for example as romek
echo $LOGINTIME # to check if variable exist
```

Now, I'm logging of and logging in as root again. To make user tytus able to use shutdown command without giving password using sudo

```
echo "tytus ALL = (ALL) NOPASSWD:/sbin/shutdown" >> /etc/sudoers
logout # then login as tytus
sudo vi # to check if other programs require password, then press
    Ctrl-C before typing password to exit sudo
sudo shutdown -h # to check if shutdown require password
```

I'm starting VM and logging in as root again. To find statically linked programs (almost every is linked dynamically) and compare their sizes

```
find /sbin/ -executable -type f -exec file {} \; | grep "
   statically linked"
stat `which grep`
stat /sbin/ldconfig
```

To synchronize time with external time server when the system is idle

```
yum install at # to run batch
batch # will start at prompt
```

```
45 ntpd -I ntp.itl.waw.pl
```

To update all packages

```
46 yum update
```

To check for any given file from which package it was installed

```
47 rpm -qf `which sh`
```

5 Additional requirements

To install SSH server

```
48 yum install openssh-server
```

As it has been mentioned, this version of system don't uses runlevels anymore and during installation appropriate target (equivalent of runlevels) is created. To generate rsh key

```
49 ssh-keygen -t rsa
```

I didn't change the default folder (/root/.ssh/id_rsa) and set empty phasephras. I copied generated keys to files with authorized key for every user.

```
50 mkdir /home/tytus/.ssh # create appropriate folders for other
       users
51 mkdir /home/romek/.ssh
52 mkdir /home/atomek/.ssh
53 cat /root/.ssh/id_rsa.pub > /root/.ssh/authorized_keys
54 cat /root/.ssh/id_rsa.pub > /home/tytus/.ssh/authorized_keys
55 cat /root/.ssh/id_rsa.pub > /home/romek/.ssh/authorized_keys
56 cat /root/.ssh/id_rsa.pub > /home/atomek/.ssh/authorized_keys
```

To disable requirement of giving password for users of group 'root' or for those using sudo (wheel group)

```
57 echo -e "Match␣Group␣wheel\nPasswordAuthentication␣no" >> /etc/
       ssh/sshd_config
```

To check if it works

```
58 ssh romek@localhost # root user should be able to login without
       giving password
59 logout # to close connection
60 logout # to logout root user, then login for example as romek and
       :
```

```
61 shh root@localhost # now password should be required
```

6 Additional requirements

To install manual downloaded package (in this case nano)

```
62 yum install wget # to download files
63 wget http://www.nano-editor.org/dist/v2.2/RPMS/nano-2.2.6-1.i386.
       rpm
64 rpm -i
65 nano-2.2.6-1.i386.rpm
```

To limit the number of processes that may be run simultaneously by a user belonging to users group to 10.

```
66 echo "@users soft nproc 10" >> /etc/security/limits.conf
```

Automatic logout after 5 minutes of inactivity.

```
67 echo "export TMOUT=300" >> /etc/profile
```