

Homework

Part 1: Virtualization – 0) Introduction

master@d08e360 (20220915-171922)

P. Mainini / E. Benoist / C. Fuhrer / L. Ith

BTI1341 / Fall 2022-23

1 Setup Your Environment

In order to follow the course, try out demos and work on exercises, a (virtual) machine running some GNU/Linux system is *required*.¹ Debian GNU/Linux is recommended, but basically every distribution should work.

★ This task is optional if you already have a running GNU/Linux system which you can use for this course.

1.1 Installing Debian GNU/Linux

We will now install Debian GNU/Linux in VirtualBox. First, download the required files:

- [vir] for running virtual machines
- [debb] to install Debian (choose small CD version for “amd64”)

After downloading, first install VirtualBox and start it. Then, configure a new virtual machine for the Debian installation:

1. Create a new virtual machine
2. Configure the following settings:
 - Name: “**bti1341**” (or some other name of your choice)
 - Type: “**Linux**”
 - Version: “**Debian (64 bits)**”²

¹Another option might also be the Windows Subsystem for Linux (WSL/WSL2) or macOS, however we provide no support for those.

²If you can only select the 32 bit version, you might have to enable virtualization (VT) in the BIOS/UEFI settings of your machine.

- Memory: At least 2 GB (2048 MB) recommended – or more
 - Virtual Hard Disk: Recommended are 10-15 GB – or more as well (use the default type “VDI”, with dynamically allocated space)
3. Start the new virtual machine and select the Debian ISO you have downloaded as boot medium
 4. Follow the steps in the Debian installer to install the system
 - Choose your keyboard layout *correctly!* This is very important as a wrong layout may for instance cause problems entering passwords etc.
 - Let the installer erase the whole disk (this will NOT erase your real disk, just the new virtual one)
 - Choose the minimal installation, we will install additional packages when required
 - Refer to the Debian documentation ([deba]) for further details
 5. Add your user to the “**sudo**” group
You should normally not work as user “root”,³ but use the utility “sudo” when you need to perform administrative tasks. For this, your normal user must be in the “sudo” group:
 - a) Login to the system and open a *shell* (console/terminal)
 - b) Switch to user “**root**” using command “**su -**”
 - c) Add your user to the group with “**adduser <user> sudo**” (where “<user>” is the login name of your normal user)
 - d) Completely log out of the system and login again
 6. Optional: install the VirtualBox guest additions: in the VirtualBox menu, select “**Install Guest Addition CD**” and refer to VirtualBox documentation for further instructions. With the additions installed, you can use copy and paste between your host and the virtual machine, mount shared folders etc.

1.2 Install the Required Software Packages

The shell command used to install software packages in Debian is:

```
sudo apt install <package1> <package2> ...
```

Use it to install the following packages:

- Packages required for the course: **build-essential**, **manpages-dev**, **python**, **python3**, **python-tk**, **mdadm**, **xxd**, **git**
- A text editor of your choice, e.g. **vim**, **emacs**, **gedit** etc.

³The “**root**” user is the system administrator account in UNIX-like systems.

1.2.1 System Upgrades

From now on, you should regularly update your system:

```
sudo apt update && sudo apt full-upgrade
```

1.2.2 git Configuration

In order to properly annotate your commits, `git` needs to be set up correctly. As your normal user, use the following commands to configure your identity (be sure to use your own name!):

```
git config --global user.name "Lastname Firstname"
git config --global user.email "firstname.lastname@students.bfh.ch"
```

▲ If you are already using `git` on the same machine (and user), this overrides your global configuration! In this case, it might be better to use the “`--local`” option and set the configuration *per-repository*. Type “`git help config`” to find out more.

2 Refresh C Programming Skills ★

If you haven’t done any C programming for a while or don’t remember how to use `gcc`, `make` and friends, it is *strongly* recommended that you invest some time for a refreshment.

Work through the *Lab Tutorial* ([ADADa] from OSTEP at your own pace:

It is best to try out the examples and play around with different command line options, read the man pages etc.⁴

3 Run the OSTEP Demos on Your Own

After having set up your environment and possibly refreshed your programming skills, it is best to ensure that everything works correctly. For this, try to run the demonstrations, which were shown during the lecture, on your own:

1. Clone the git-repository containing the code examples from OSTEP ([ost]).
2. Compile the demonstrations and run them, use the OSTEP introduction ([ADADb]) as reference
3. Read the source code and try to understand it

4 Operating System History

Finally, read some bits about operating system history: Section 2.6 (p.12-16) from the OSTEP introduction ([ADADb])

⁴Note: Due to a bug in Debian 10, the man pages for `gcc` are currently not installable, please refer to the online docs in [gcc] instead.

References

- [ADADa] Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, *OSTEP, Appendix F, Laboratory: Tutorial*, <http://pages.cs.wisc.edu/~remzi/OSTEP/lab-tutorial.pdf>.
- [ADADb] ———, *OSTEP, Chapter 2, Introduction to Operating Systems*, <http://pages.cs.wisc.edu/~remzi/OSTEP/intro.pdf>.
- [deba] *Debian GNU/Linux, online documentation*, <https://www.debian.org/doc/>.
- [debb] *Debian GNU/Linux, small CD ISO image*, <https://www.debian.org/distrib/netinst>.
- [gcc] *gcc 8.3.0 documentation*, <https://gcc.gnu.org/onlinedocs/gcc-8.3.0/gcc/>.
- [ost] *GitHub.com, remzi-arpacidusseau/ostep-code: Code from various chapters in OSTEP (http://www.ostep.org)*, <https://github.com/remzi-arpacidusseau/ostep-code>.
- [vir] *VirtualBox Downloads*, <https://www.virtualbox.org/wiki/Downloads>.