Exercise 1 4/4

Exercise sheet 1 - Answers

Exercise 2

a)

Virtual Box

Debian VM: With Virtual Box, no software of driver installation was necessary for the Debian VM to have Internet access. I did a rough check of the Internet connection by opening YouTube in Firefox that was part of the installed Debian distro.

Install Steps:

1. Download VirtualBox executable from official website
2. Download Debian ISO from official website
3. In Virtualbox create a new VM by clicking New.
4. Enter a name for the new VM and st the type to Linux and the version to Debian (64-bit).
5. Set memory limit to 2GB
6. Create a new virtual harddrive for the VM to use
7. Select VM and click start
8. Select the Debian iso int he apparing window
9. Wait dor Debian to boot and follow installer

No special commands or menus required.

Alpine VM: For the Alpine VM there too weren't any special steps required to get Internet access. The only thing I had to ensure is that DHCP was enabled during the `alpine-setup` script. I tested the Internet connection by using the `ping` command to ping www.google.com.

Install Steps:

1. Download VirtualBox executable from official website
2. Download Alpine ISO for VM from official website
3. In Virtualbox create a new VM by clicking New.
4. Enter a name for the new VM and st the type to Linux and the version to Other (64-bit).
5. Set memory limit to 1GB
6. Create a new virtual harddrive for the VM to use
7. Select VM and click start
8. Select the Debian iso int he apparing window
9. Wait dor Alpine to boot and follow installer
10. Create a new user with `# adduser <YourUsername>`
11. Install doas so you can allow to use root privileges with `# adduser # apk add doas`
12. To enable user to use root privileges add it to the Wheel group `# adduser <YourUsername> wheel`

No other special commands required.

QEMU/KVM

They both have access to Internet. Going through the installation automatically established an Internet connection (same as for the Virtual Box installation), forwarding the default host connection to the guest. Connection checked on both VMs with `ping` command.

Debian install QEMU/KVM

1. Download Debian ISO from official website.
2. In Virt-Manager click on the "Create a new virtual machine" button.
3. Select "local install media" (to choose to boot from the ISO).
4. Select memory and CPU's to allocate for the VM, than also the virtual disk space.
5. The machine boots into the ISO.
6. Choose "Graphical Install" option.
7. Select language, location and keyboard.
8. Config network and domain.
9. Set ip password for root.
10. Set up a user and his password.
11. Select clock time and partition disks ("Guided - use entire disk).
12. Confirm and install base system.
13. Configure the package manager with the best domain for the mirrors.
14. Select the DE and the Display Manager
15. After installation click "continue" and the system will reboot.

Alpine install QEMU/KVM

1. Download the iso from website.
2. In Virt-Manager click on the "Create a new virtual machine" button.
3. Select "local install media".
4. Select memory and CPU's to allocate for the VM, than also the virtual disk space.
5. Than the machine boots into the ISO and you can login as root
6. Run the *setup-alpine* skript:
    1. Select keyboard layout
    2. Select hostname
    3. Select internet interface and other internet options (accepted defaults)
    4. Select root password
    5. Select timezone
    6. Select mirrors with "f" and select automatically best for your location
    7. Select ssh server
    8. Select disk and mode to install. Traditional disk install with "sys". Base install will be done.
    9. Reboot the machine with *reboot*.
7. Login again as root
8. Create a new user: *adduser -g "Ruben Hutter" ruben*

9. Select password and confirm it
10. Add user to wheel group: *adduser ruben wheel*
11. Install important packages: *apk add doas vim* (you can also follow the wiki to install sudo instead)
12. Make wheel group persistent for user: *vim /etc/doas.conf* and append "permit persist :wheel"
13. Enable community repository: *vim /etc/apk/repositories* and remove the comment of the correct alpine version community repo
14. Refresh the repositories: *apk update*
15. Change default shell installing libuser package and then *lchsh ruben* and enter the path to the shell (e.g`/bin/bash`)

---

b)

Virtual Box

1. Select the machine you want to duplicate
2. In the menu bar click on **Machine** and then on **Clone**
3. A window which leads you through the cloning process is opened. There enter the name of the clone and press *Next*.
4. On the 2nd page set the *Clone Type* to *Full Clone* to ensure that the whole VM including the hard disk file gets cloned to prevent any dependencies of the clone to the original VM.
5. Click Clone

QEMU/KVM

Just run command `sudo virt-clone --original <VM NAME> --auto-clone` to clone the machine. The clone was directly ready to run. In the background the .qcow2 file of the VM was duplicated, and the name was automatically completed with *"-clone"* thanks to the `--auto-clone` flag.

2/2

Exercise 3

a)

you get root-privileges

sudo: The command `sudo <command>` runs a single command with the permissions of the root. The user is prompted to enter the root password before the command is run. Sudo stands for substitute user do. ✓

sudoer-list: Is a file (/etc/sudoers) that contains a list of user or/and groups and sets different rights to them. It offers the administrator a tool to control who can to what. Every time a user calls `su` or `sudo`, his username is checked against this list. ✓

su: The command `su <USERNAME>` prompts the user to enter the password of the user specified as the argument and then starts a shell instance with the *you become that user

permissions of the specified user. It basically lets a user execute commands with the privileges of another user.* *su* stands for *substitute user* or *switch user* ✓

---

b) Only installations that are on user level (executable, and non, files are installed only for the user) doesn't need sudo privileges. Normally if you install a package on Linux it is system wide, so you need to execute them with `sudo <CMD>`. ✓

---

c) The package manager let's you install, remove, upgrade and more generally manage packages / programs. It's like an "app store" (with everything for free), a common place where to download your applications so that you don't have to browse the web and search for some executable. It depends on the distro, but the packages are located in the different repositories, and therefore there are different repository maintainer. ✓

---

d) I would use `touch <NAME OF FILE>` command. You can also use `cat > <NAME OF FILE>` and than terminate cat command without writing anything to it. ✓

---

e) It concatenates files to the standard output (from there you can also redirect it into an other command/file). If you don't know, just enter `man cat` in terminal ;) ✓

---

f) Piping is when you pass an output of a command into an other command as input, or redirect an output to a file (or vice versa). For example: `ls | wc -l` or `echo "hello world!" >> hello.txt` ✓

---

g) I can directly start them in the background, with *&* at the end of the command, or send them later. With `jobs` you get a list of all current jobs running, and with `fg <JOB ID>` you bring in *foreground* the task by attaching it to the shell. ✓

---

h) The command `file <FILENAME>` can be used to display the files type. ✓

---

i) The `touch <FILENAME>` command creates a new empty file if the file is not already present, or changes the documents timestamps. ✓

---

j) It exists the *man*, the manual pages that are accessible from any terminal, that gives a complete overview of the searched parameter. There is also the

`--help` flag (in some cases also `-h` works) that can be added to a command. With the flag, an essential overview is printed directly into the shell.

✓ 2/2

Total: 8/8