

20HT – 1DV512 – Operating Systems Group Assignment 1

[[1]](#endnote-1)

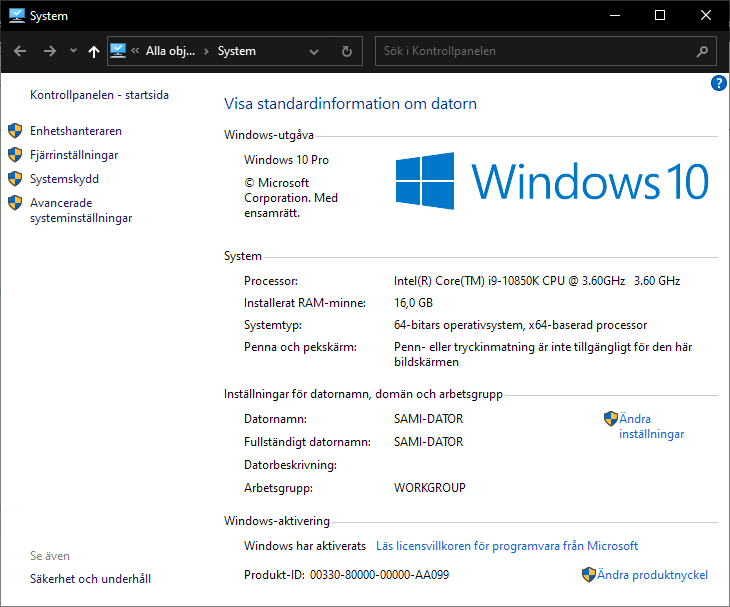
Student: Sami Mwanje

ID/mail: mm223kk@student.lnu.se

Assignment date: 2020-12-06

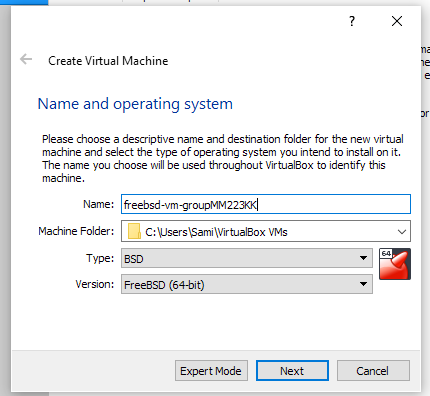
Hand in date: 2021-06-18

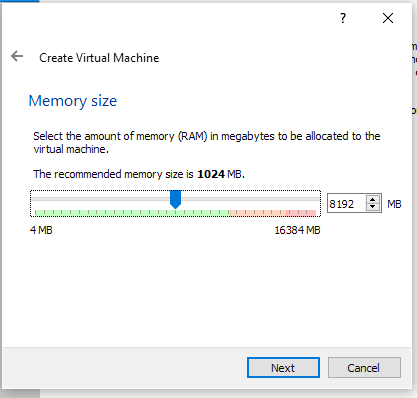
Task 1:

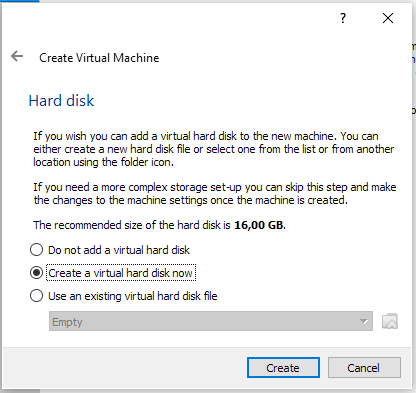
****

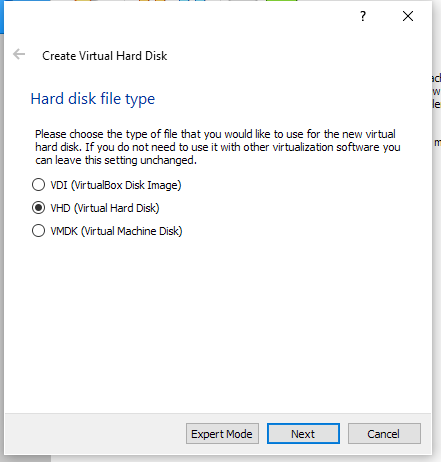
I choose to use VirtualBox for the FreeBSD VM. This was the simplest to install on my machine. Here is the information about the host that the VM will run on. Intel Core i9-10850k CPU @ 3.60 GHZ. 10 cores, 20 threads. Windows 10 Pro 64-bit.

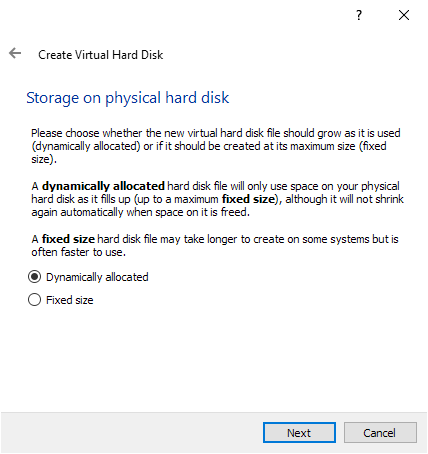
Task 2:

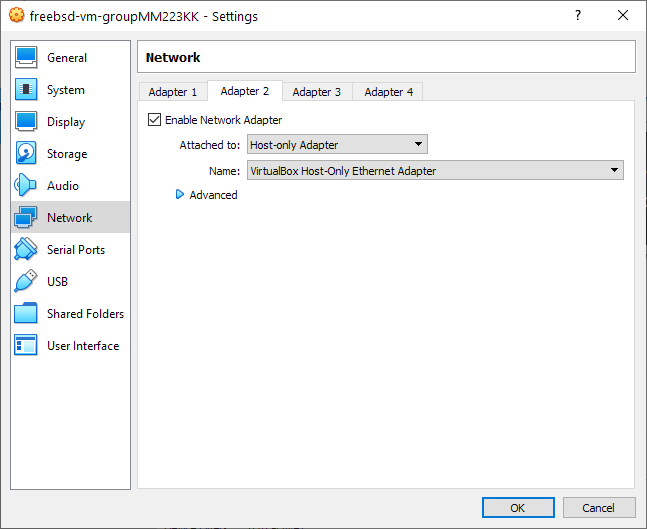


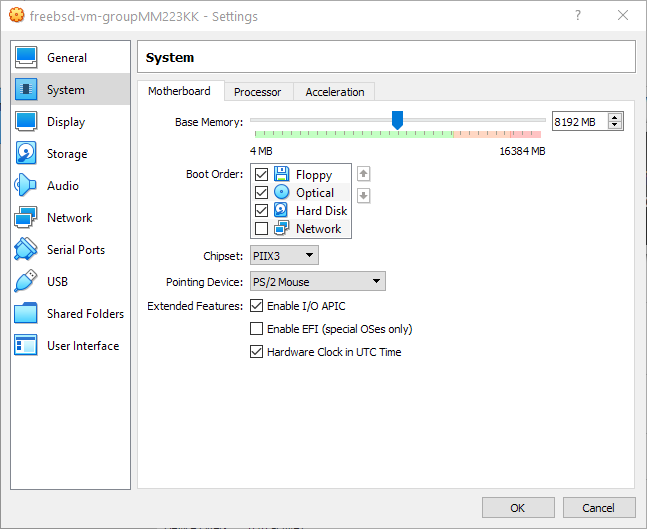


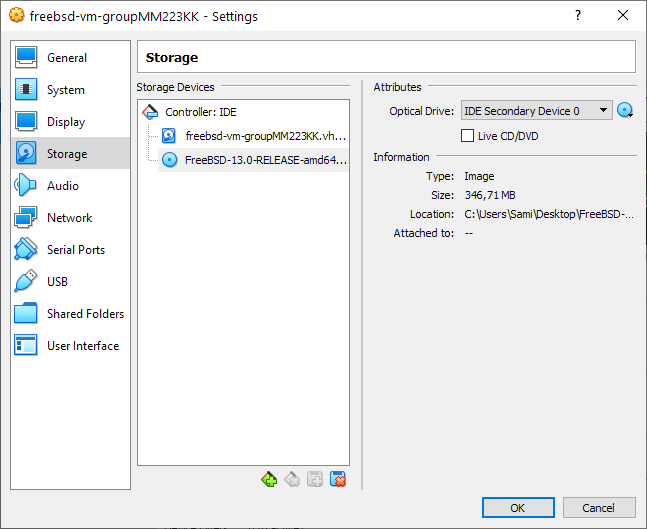


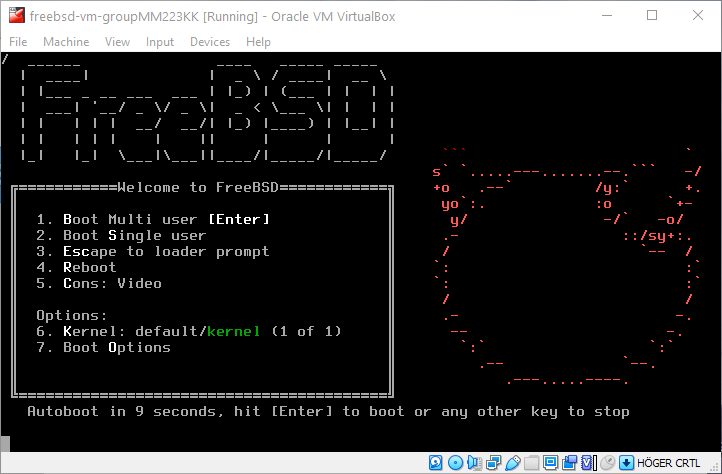


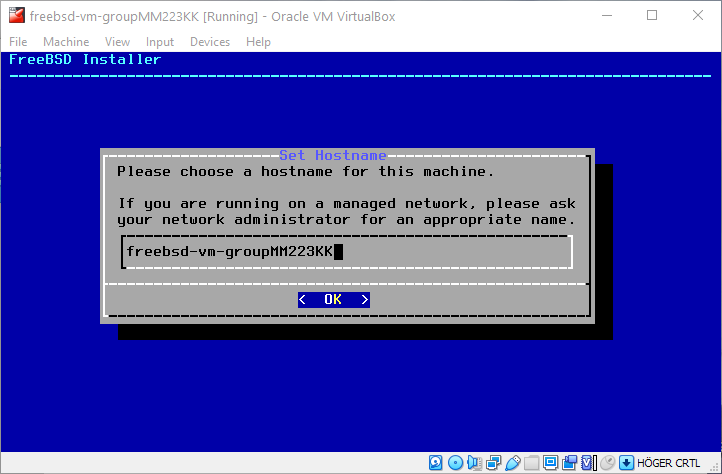


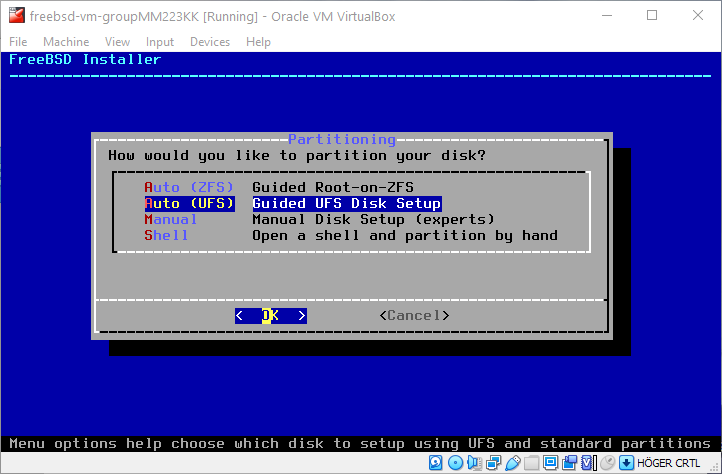


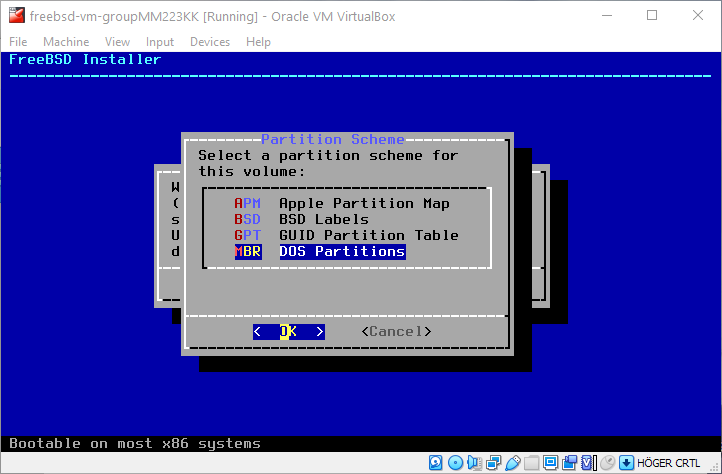


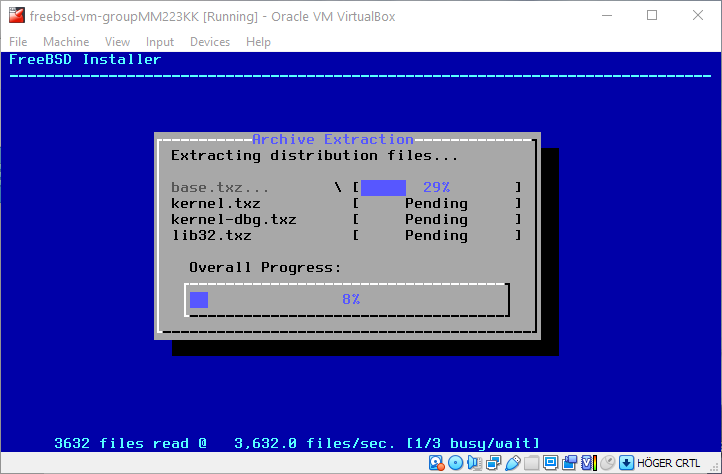


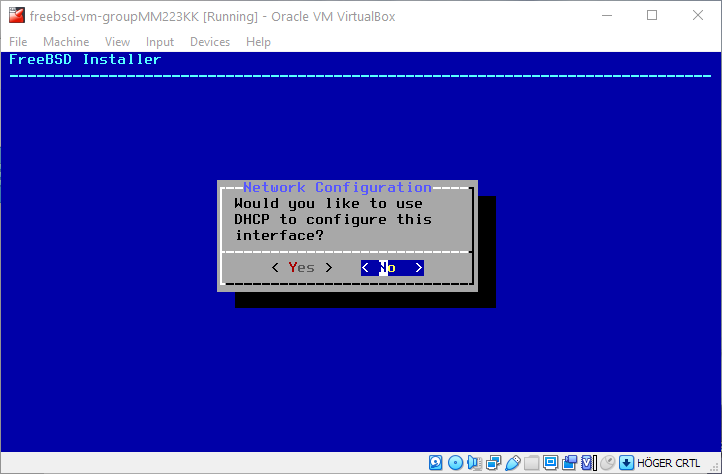


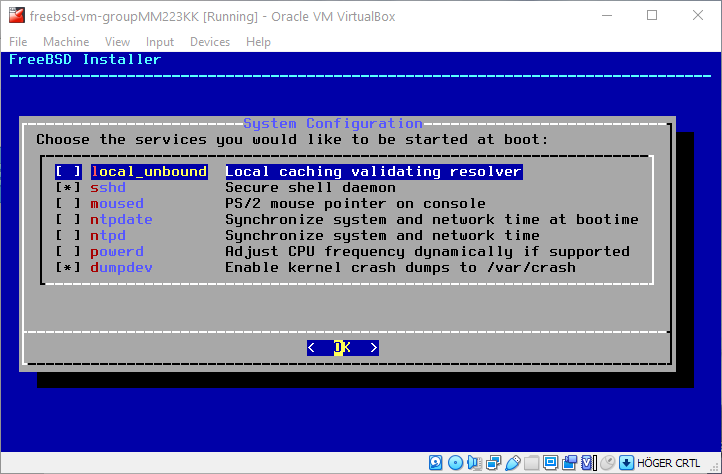


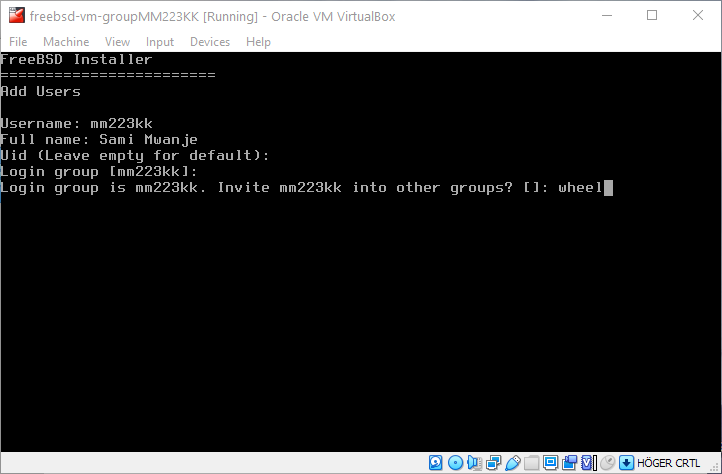


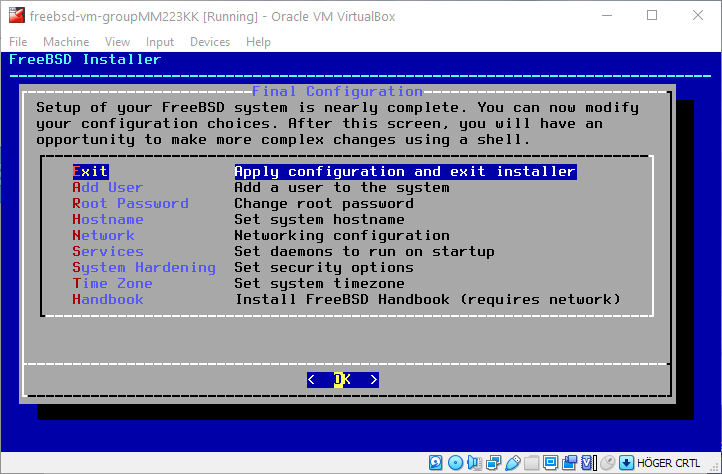






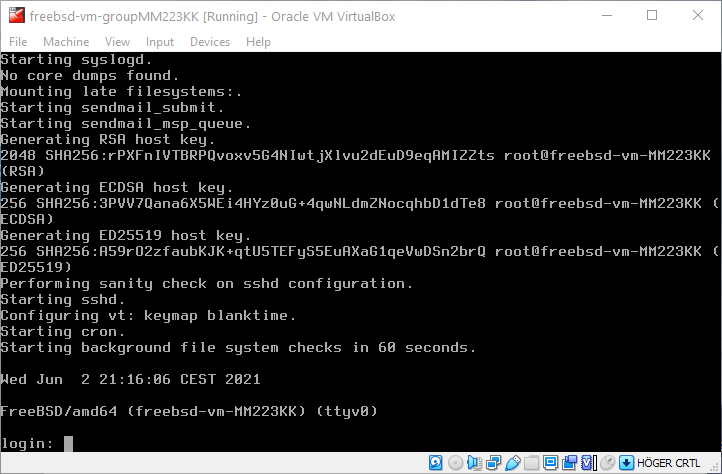


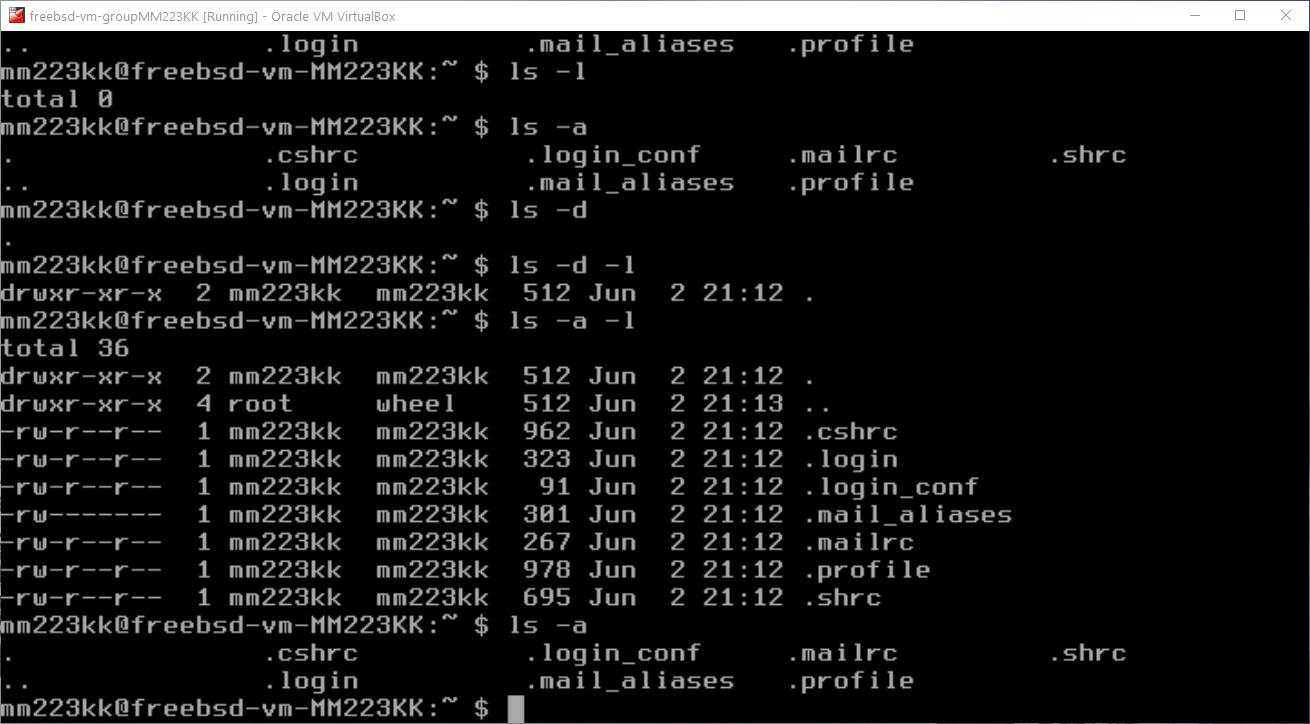




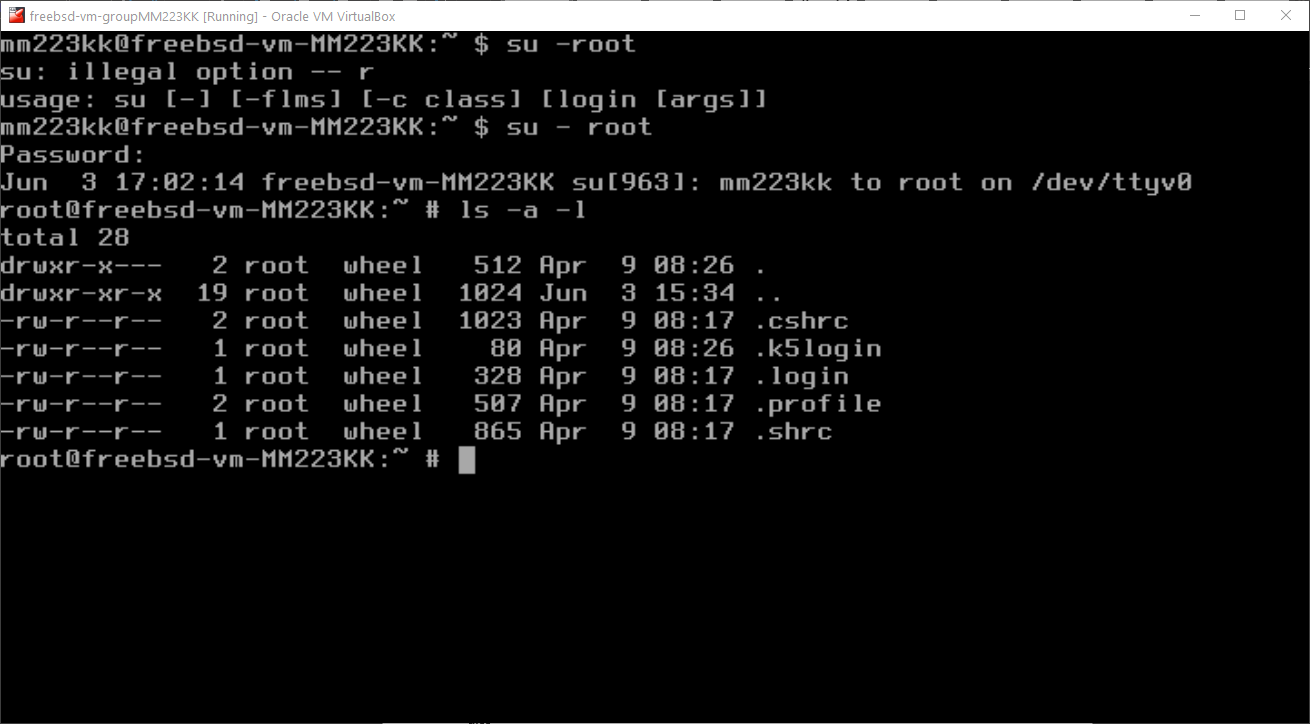
I choose to install it using **8192** of **memory**, which is maybe not needed. I had some problems with the internet setup on the “**bootonly**” version, so I installed it using the “**1disc**” version which went smoothly. The internet was then setup later on in the VM. Since I am working alone, I created two users. **Admin** and **mm223kk**.

**Task 3.1:**

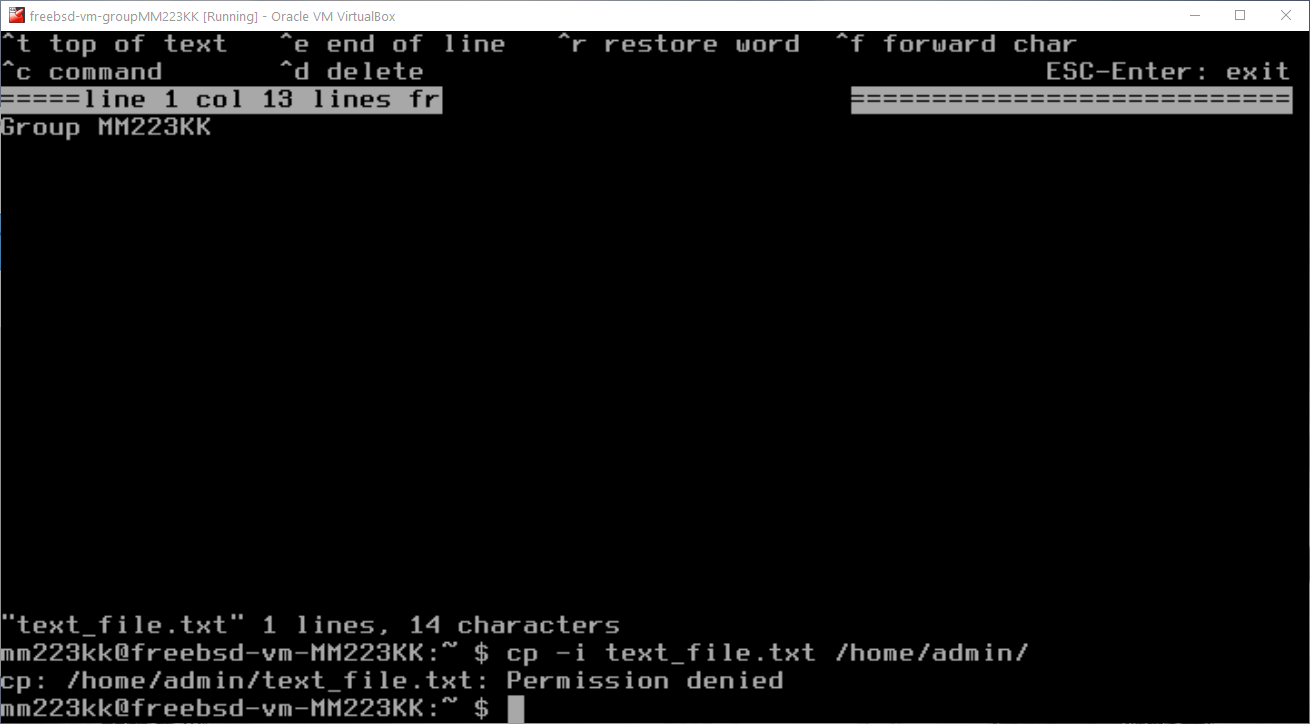
 Signing into the account “mm223kk”.



Listing the files in the home directory, using “**ls -a -l**”. “**ls -a -l**” will list complete file contents of the current directory including the hidden files and file permissions.



The command ”**su - root”** is used to change from one user to another. Here I am changing from “mm223kk” to the user root. After that the “**ls -a -l**” is used once again to display the complete list of the current home directory. We can see 28 files in root and 36 in mm223kk.

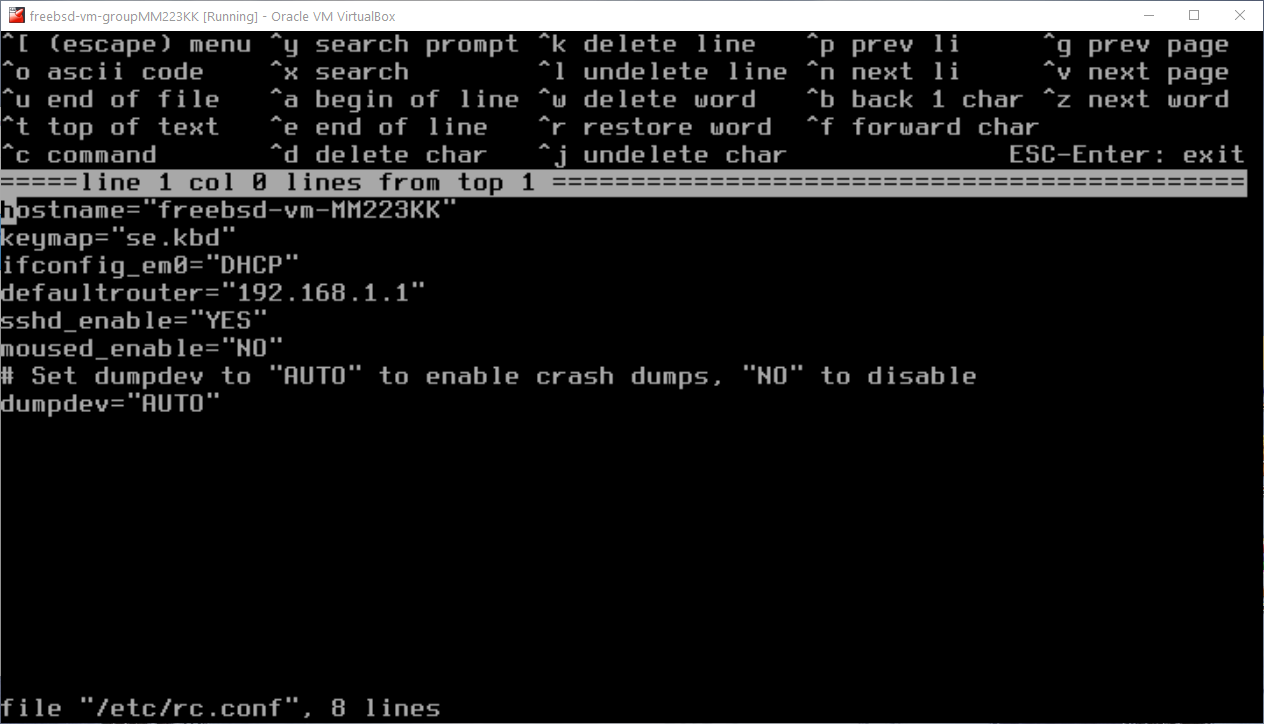


Here I am creating a new text file using the command “**ee**”. The file will have the content “**Group MM223KK**”. After the content has been added the file gets the name “**text\_file.txt**”.

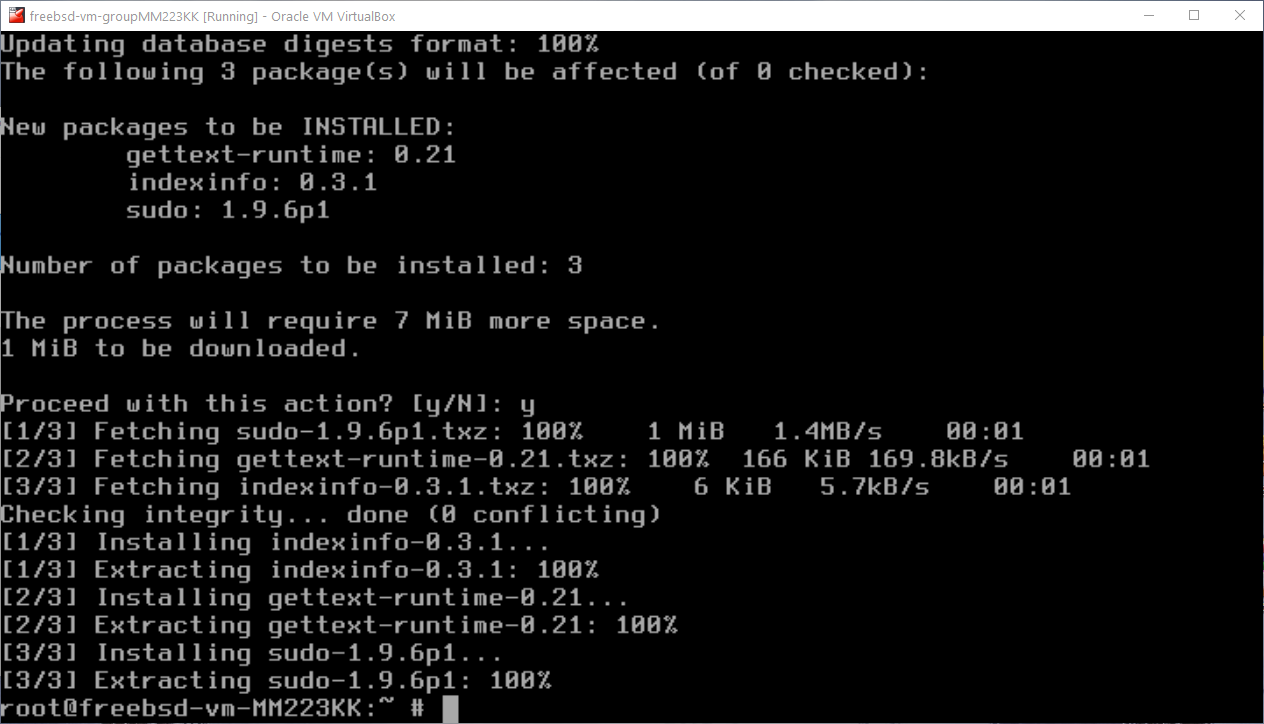
An attempt to copy the file from the current directory to the user “**admin**” is denied. This is done with the command “**cp -i text\_file.txt /home/admin/**”.

The command is denied because the user “mm223kk” do not have the permissions to write a file to another user. It may have worked if the user was calling the command through the root.

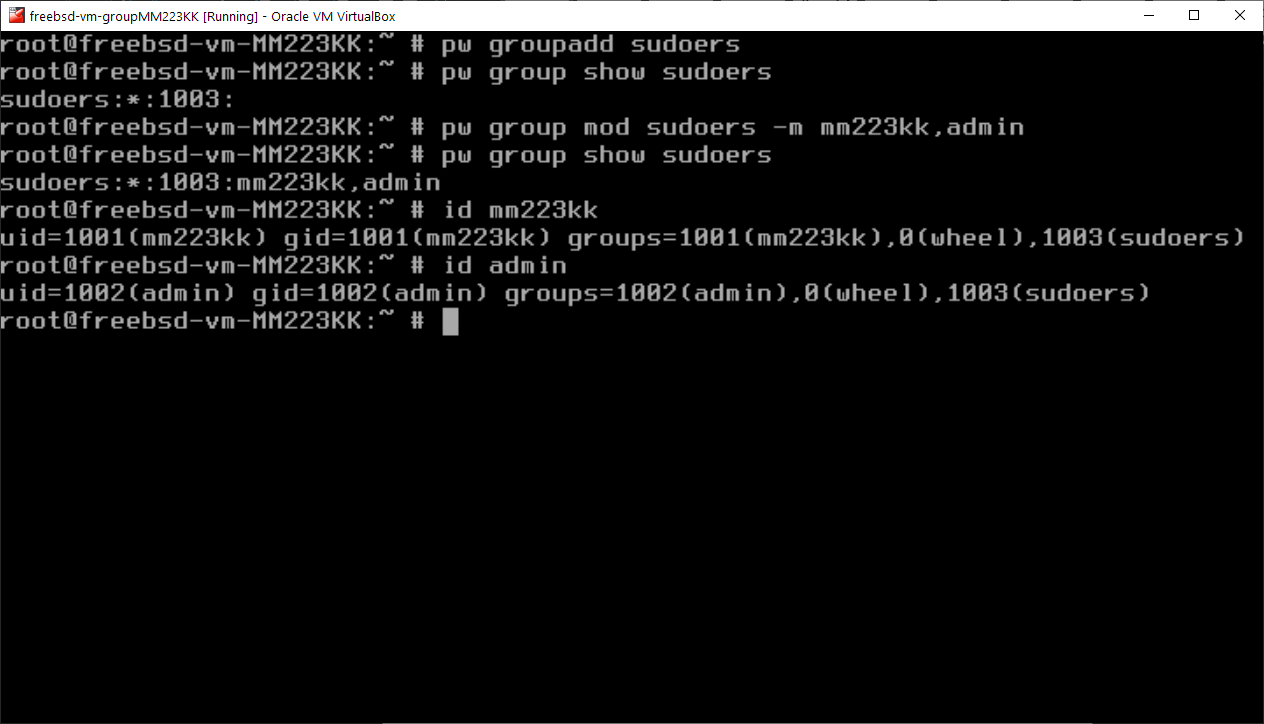
**3.2:**



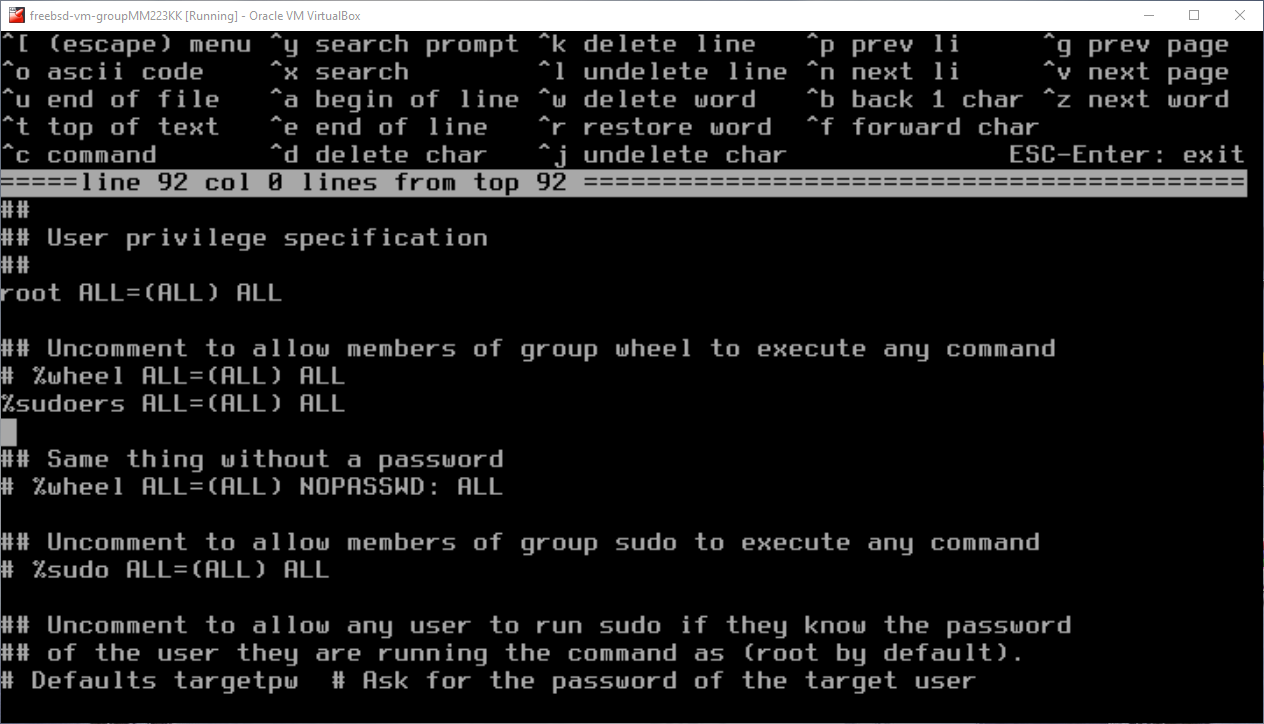
Did not get the network to work on the first try so I had to edit some stuff in the “**rc.conf**”. This was done by signing into root and using the command “**ee /etc/rc.conf**” After that the line **ifconfig\_em0=”DHCP**” was edited and the “**defaultrouter=** ” was set to my home router address.



After the network setup the command “**pkg install sudo**” could now be used to install the sudo package. The installation went on smoothly without any errors.

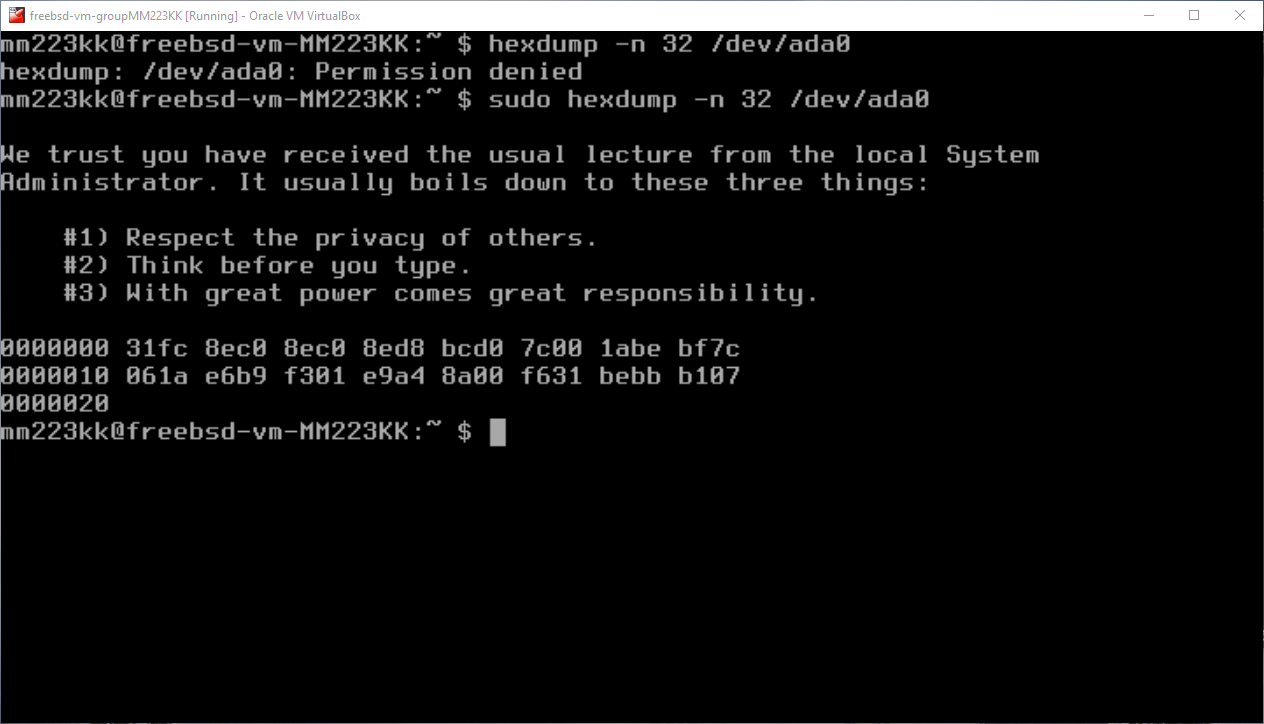


First, I created a new group called “**sudoers**” using “**pw group add sudoers**”, and the I listed the members in the group using “**pw group show sudoers**”. The group was first empty, so I added the user mm223kk and admin to the group with full privileges using “**pw group mod sudoers -m mm223kk,admin”.**



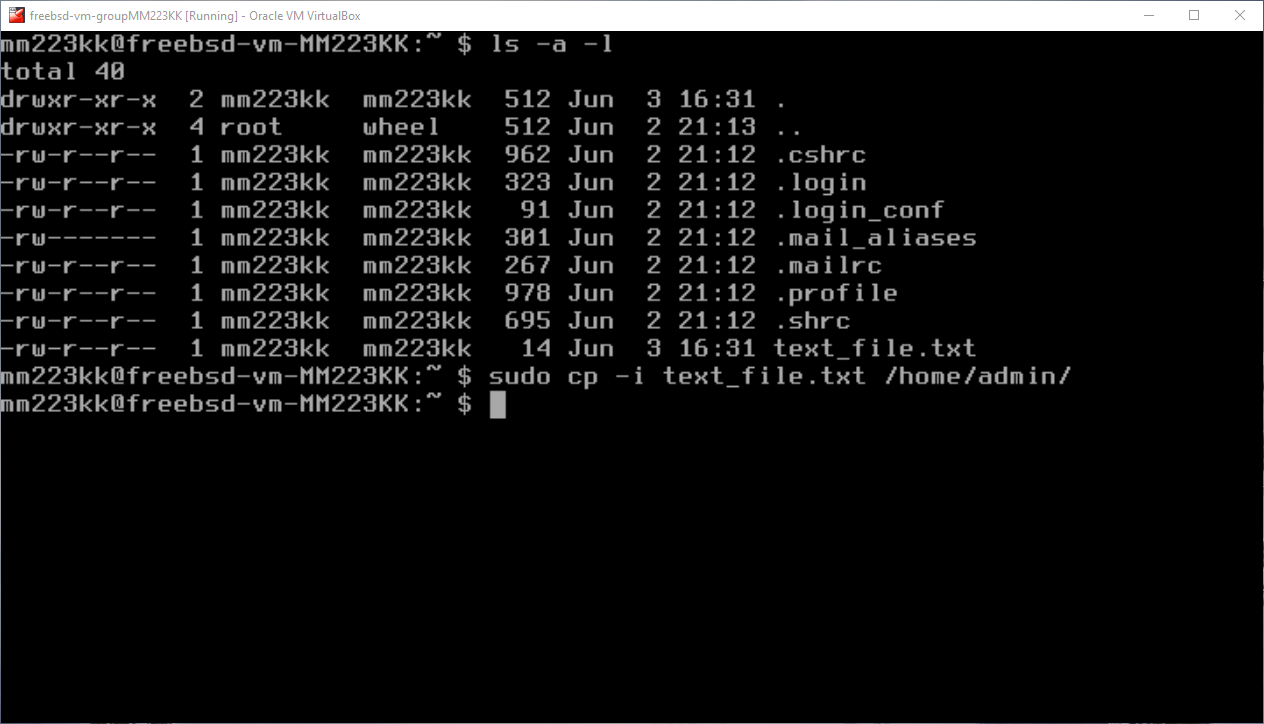
The last thing was to run the command “**ee /usr/local/etc/sudoers/”** and create a new line with “**%sudoers ALL=(ALL) ALL**”. This allowed all members of the group sudoers to execute any command.

**3.3:**

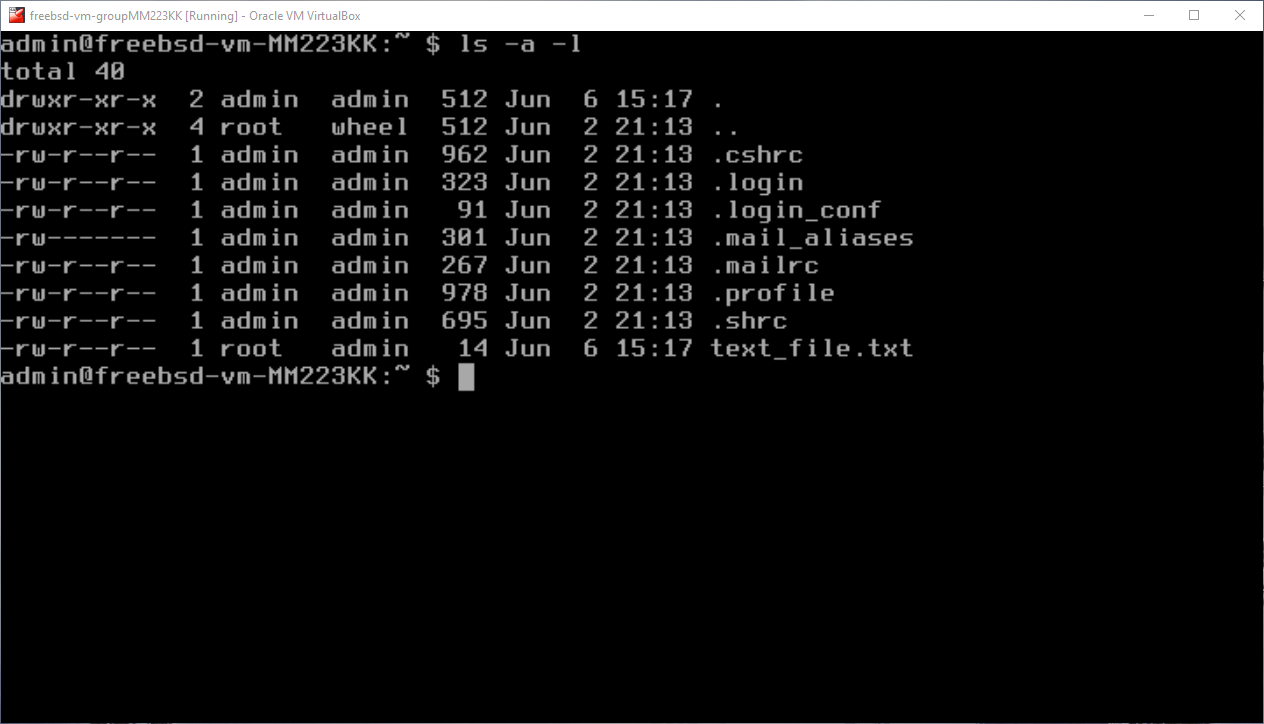


First, I run the command **“hexdump -n 32 /dev/ada0”.** This is called without any privileges directly from the “mm223kk” user. Running the same command with **“sudo”** takes use of the privileges that the user was granted. The **“hexdump”** command is used to display the specified files in hex. The **“-n 32**” decides how many bytes that are going to be printed out in this case 32. The “**sudo”** command allow the user to use a certain command as a supervisor or another user, therefore **“sudo hexdump -n 32 /dev/ada0 “** gives no permission denied message.

**3.4:**



Trying to copy the “**text\_file.txt**” file to **/home/admin/** using **sudo** from mm223kk. As stated earlier, mm223kk now has the privileges to run the command **sudo** and can now write files to other users. On the image below we can see that the copy was successful.

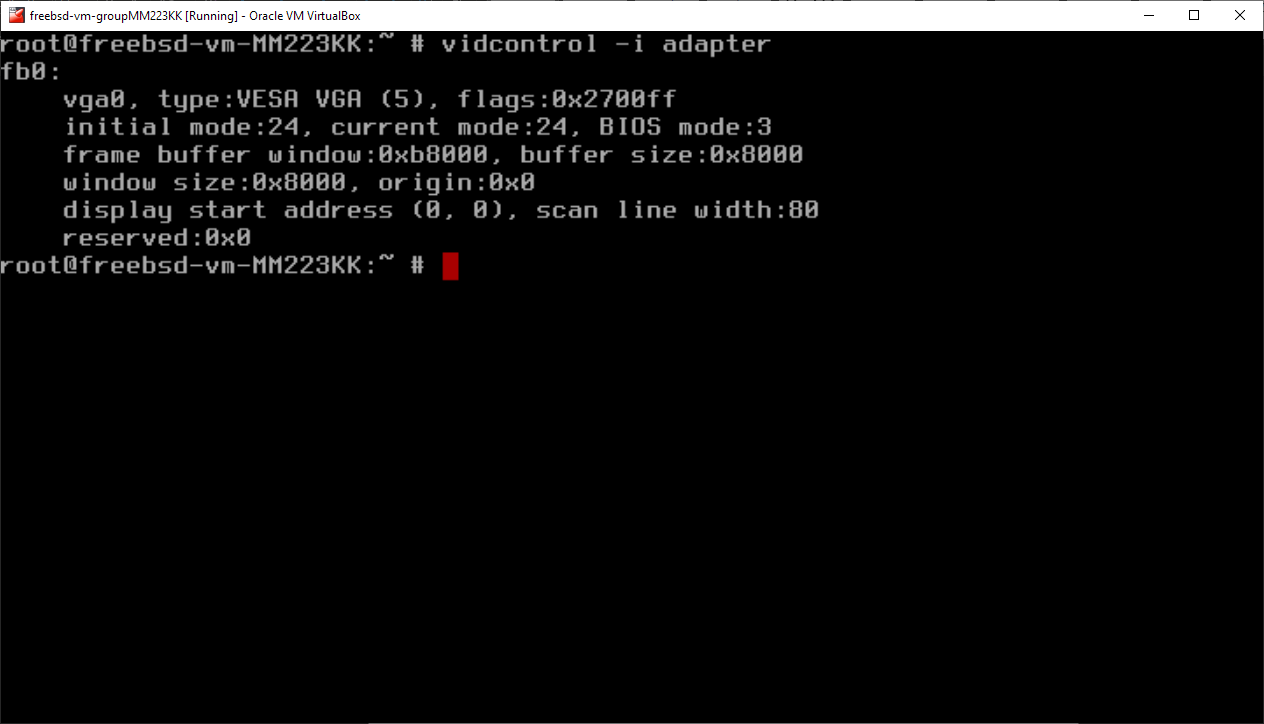


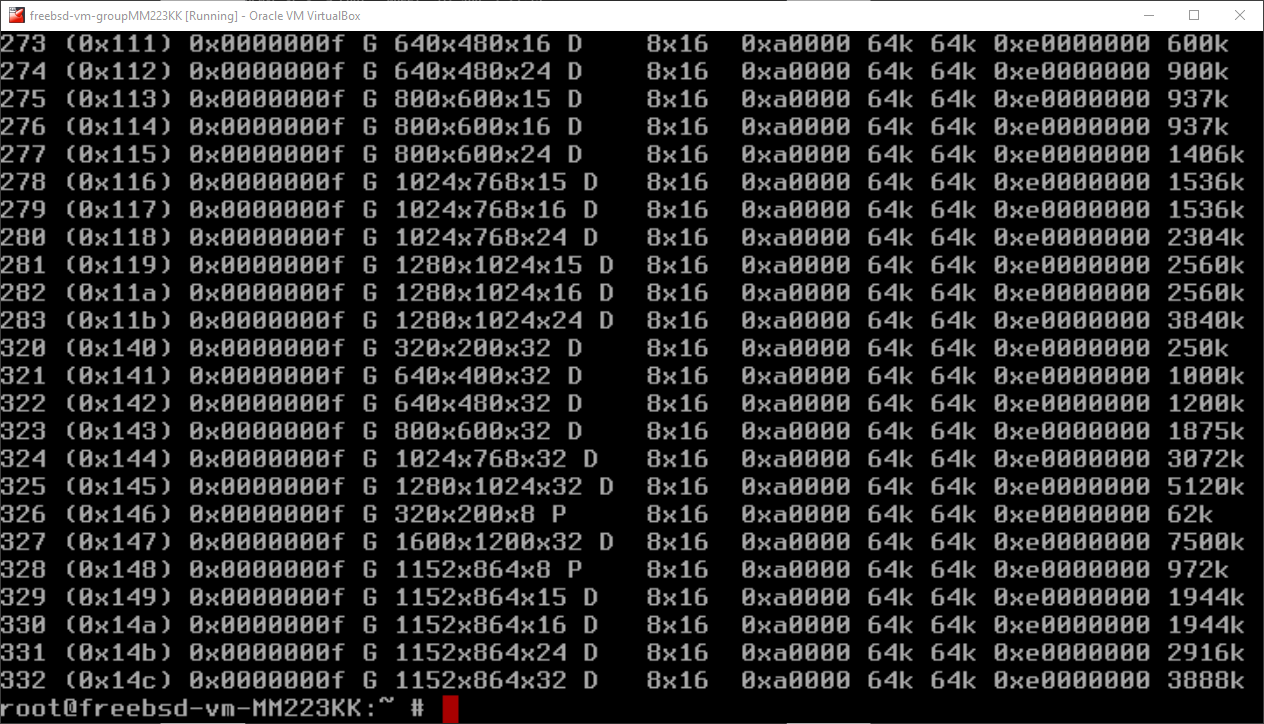
3.5:



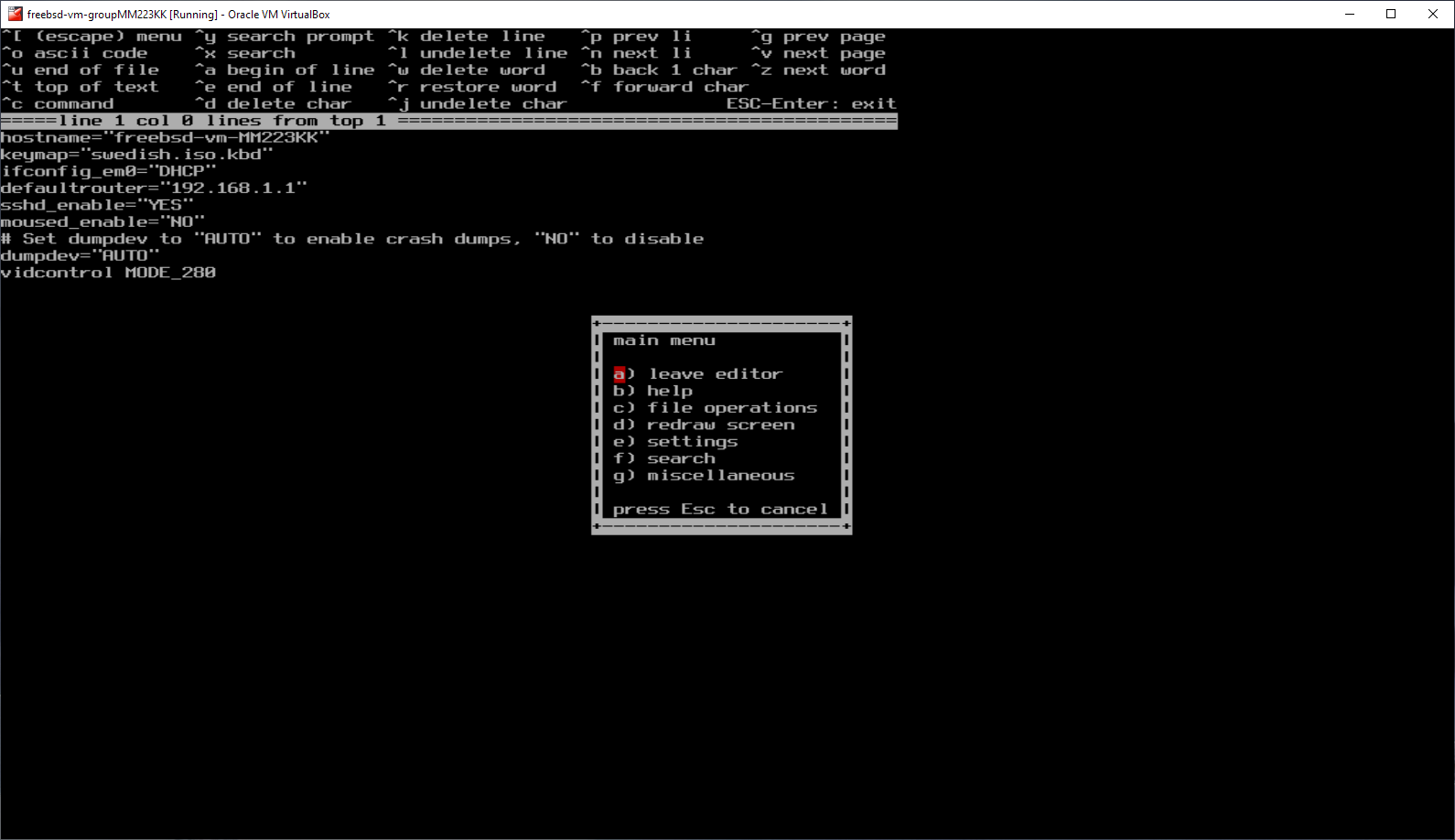
I first got an error about “**Inappropriate ioctl for device.”** So, I had to go to the “**/boot/loader.conf”** and create a new line that stated “**kern.vty=sc**”.

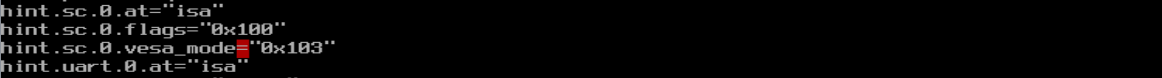
Running the “**vidcontrol -i adapter”** command displayed this:





Here I used “**vidcontrol -i mode”** to decide which mode I preferred. After some tries with editing “**vidcontrol MODE\_XXX**” in “**ee etc/rc.conf**” I chose 280.



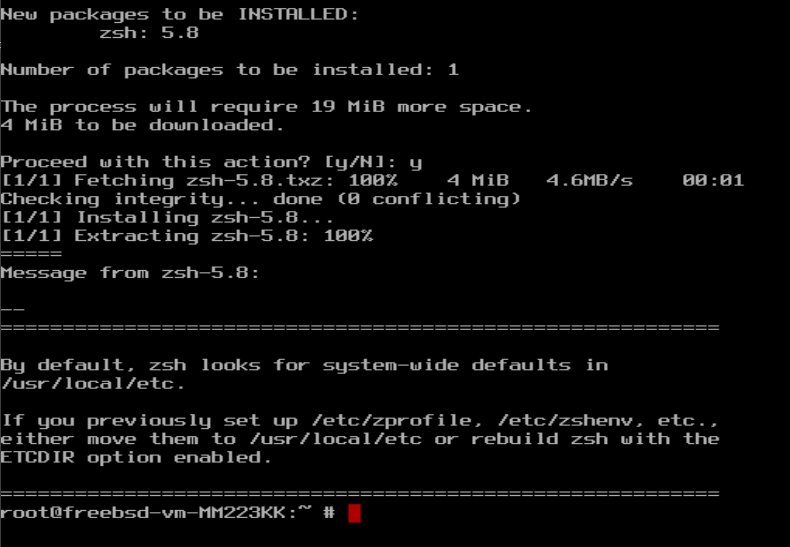


I also added the line “ **hint.sc.0.vesa\_mode=0x103** “ into “**ee /boot/device.hints**”. In order to fully fix the syncons.

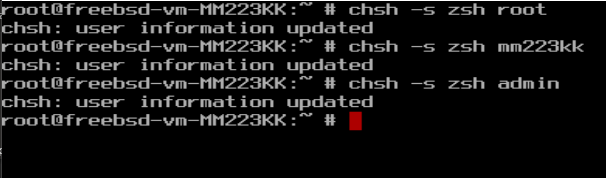
****

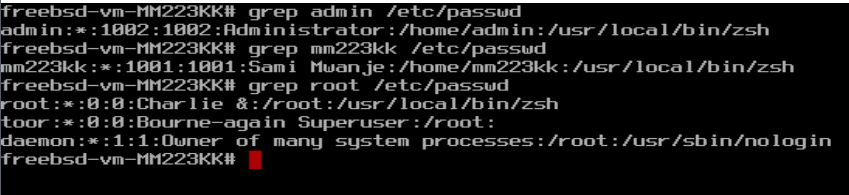
A run of “**vidctrontol -i adapter**” and “**vidcontrol show”.**

**3.6:**

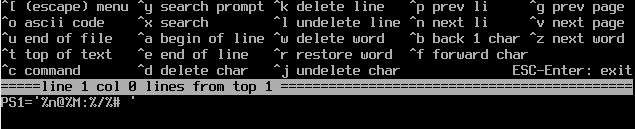


The first thing I did was installing the new **zsh shell** by running the command “**pkg install zsh**”. As can be seen the installation went smoothly.



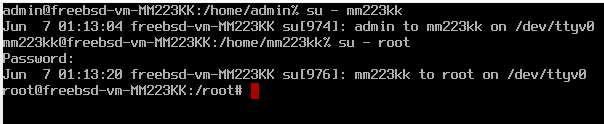


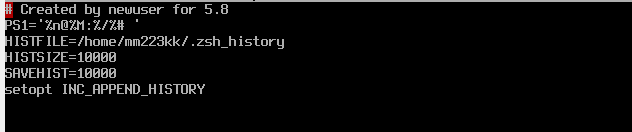
I installed the new shell for all the users using “ **chsh -s zsh “username”** “. The zsh was also set for all users with grep.



First, I had to edit the file “.**zshrc**” for all users. This file was located in home. And could be edited with the command “**ee .zshrc**”. In this file I created a new line **PS1=’%n@%M:%/%# ’.** “**%n”** displays the current user. “**%M”** the host name.

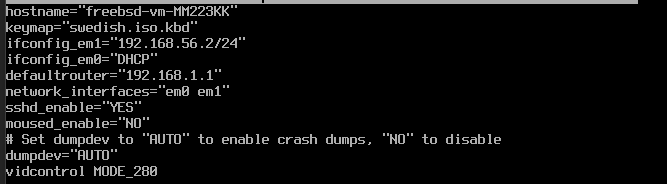
“**%/”** prints the current location/path. **%#** checks if the current user has root privileges and displays “**#**”. If the user **does not** have root privileges **“%”** will be displayed. Here is a test on the account:



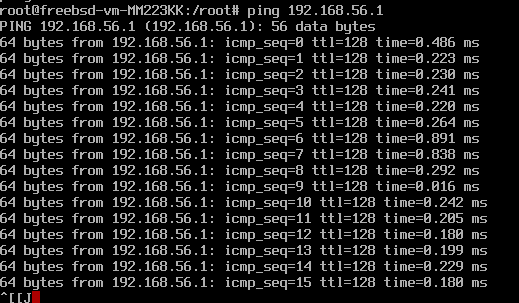


Finally, the history settings were set. These settings were found on the internet and were recommended. This was done for all users. “**HISTFILE = /home/username/.zsh\_history**” will be the location for the history files.

**3.7:**

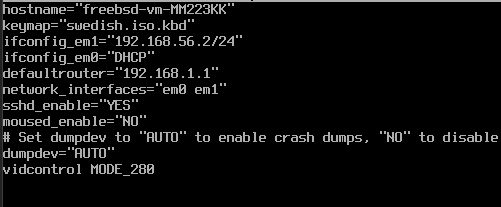


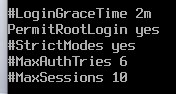
I configured the host-only card by adding two new lines“**ifconfig\_em1= 192.168.56.2/24**” and “ **network\_interfaces=”em0 em1**” “. The address was found by using windows **cmd** and **ipconfig** which displayed **192.168.56.1** so I changedit to **192.168.56.1.**



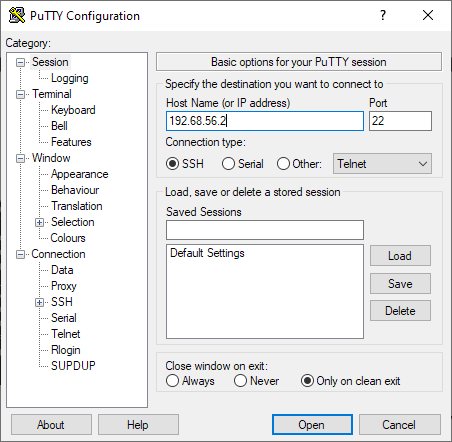
The “ping 192.168.56.1” displayed an established connection to the vm-static-ip-address.

**3.8:**

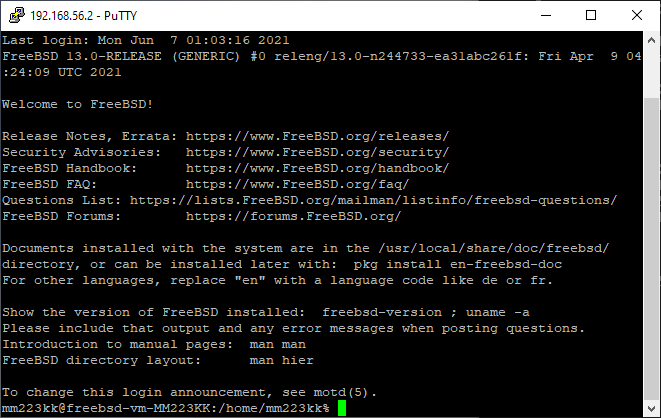
****

****

As it can be seen in the “**ee etc/rc.conf**” file the “**sshd\_enable**” is set to yes. The **PermitRootLogin** option is also set to yes in “ **ee /etc/ssh/sshd\_config** ”. Now it is up to **PutTy**.

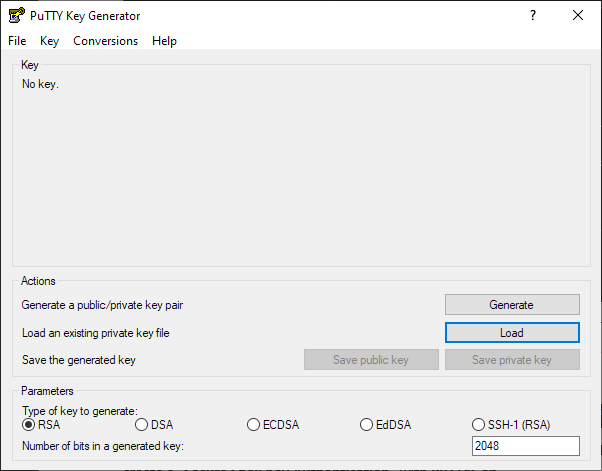


The Host IP address is set to the same address as the one that was set in **ifconfig\_em1= 192.168.56.2/24**



The PutTy terminal now pops up and asks for login information. After that the information has been entered, I can now have access to the VM using PutTy.

**3.9:**

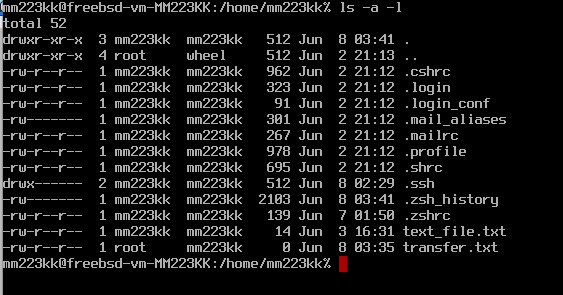


First, I need to create an .ppk file for a “**Secure Shell Key Authentication**” with **PuTTYGen.**

When this was done some settings were needed to be changed in **“/etc/ssh/sshd\_config**” so the authentication did not ask for password every time a transfer was waiting. “**PasswordAuthentication no**”, ”**ChallengeResponseAuthentication no**” and ”**UsePAM no”.**

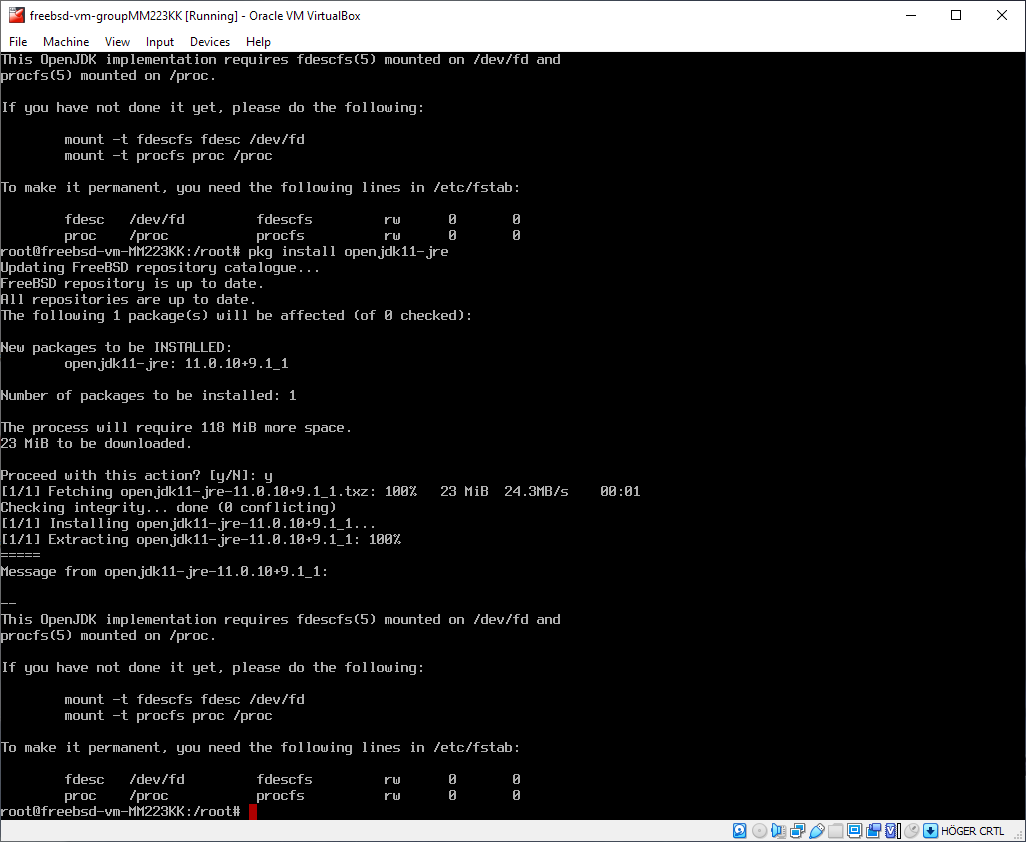


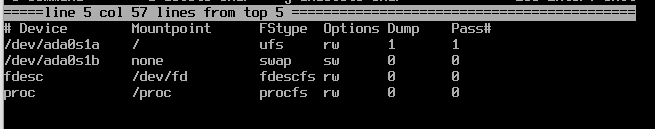
Now I need to download pscp.exe from PuTTY’s webpage and export it to PuTTY directory. From the PuTTY directory I used windows CMD with the line “ **pscp.exe -i C:\Users\Sami\Desktop\transfer\private\_key.ppk C:\Users\Sami\Desktop\transfer\transfer.txt** [**root@192.168.56.2:/home/mm223kk/**](mailto:root@192.168.56.2:/home/mm223kk/) **” pscp.exe**  is to run **pscp**. **-i** is to locate and use the earlier created authentication file and then comes the location to pull from and to.

****

When it comes to the VM location I used the root account and then pointed it to the home directory of mm223kk. Here the file newly transferred file can be seen as **transfer.txt.**

**TASK 4:**

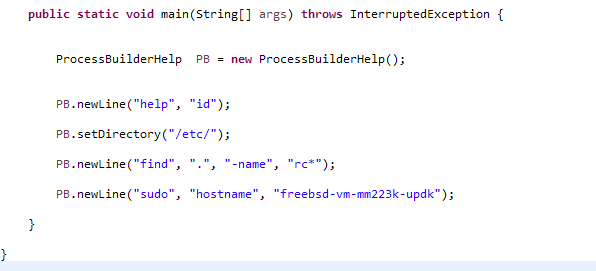
****

****

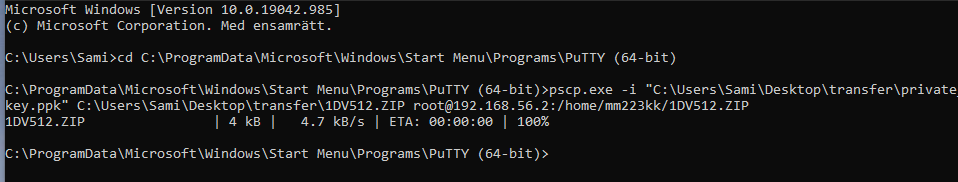
I Installed the “**openjdk11**” using “**pkg install openjdk11**” and “**openjdk11-jre**” using “**pkg install openjdk11-jre**”,the installation went smoothly as it can be seen above.

I also had to add the last two lines in the file “/**etc/fstab**” to make the settings permanent.

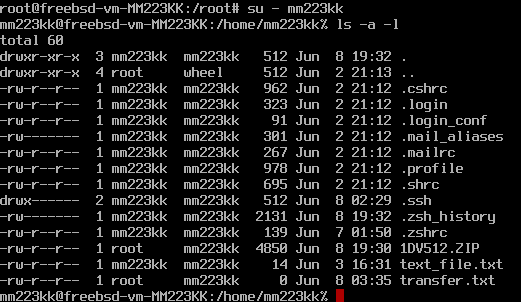
**4.1:**

****

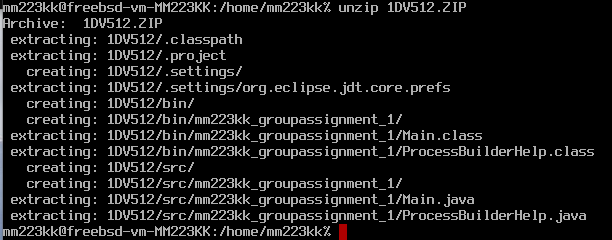
First, I created the java code with a main and a ProcessBuilderHelp class.



The files that I created during the implementation were zipped and then transferred to the VM. The zipping was needed in order to save time and not send the files file by file.

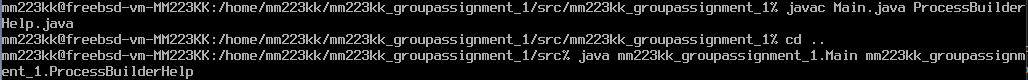


The files can be seen as “**1DV512.ZIP**”.

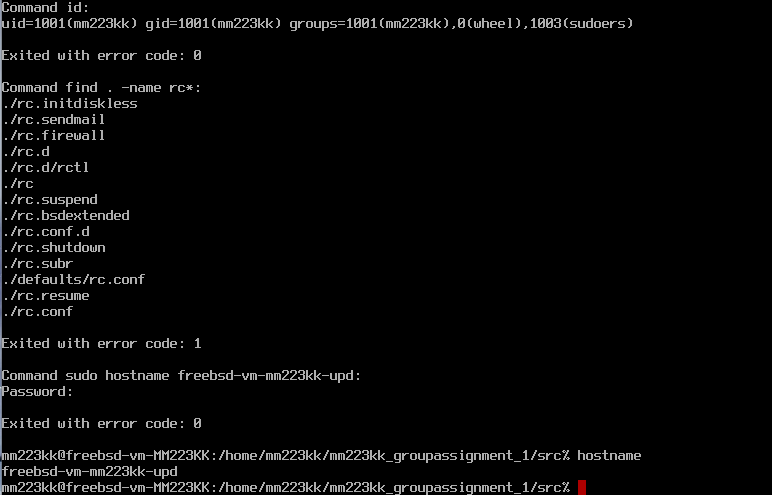




The files were then unzipped and could be found in the location 1DV51.



After this I changed the current path to the location of **Main** and the **processBuilderHelp** class. The files first had to be compiled this was done using the line “**javac Main.java ProcessBuilderHelp.java**”. The compilation went well without any errors after a little misspell.



I had some issue with running the code after the competition. After some googling I found out that I had to go one directory backwards and run the codes using “**java groupassignment\_1.Main groupassignment\_1.ProcessBuilderHelp**” The process went successfully, and the system asked for the password in order to change the VM-host name. The new hostname can be seen here below with a little misspell. Though this is enough to confirm that the code was working and complied successfully.

Due to the fact that I had many courses behind me under this year and the last led to that I missed the first group assignment dates and had to do this one alone. except that the work went smoothly, and I learned a lot of new things while completing the assignment. I am currently hosting some servers from my apartment at home and with this information it comes in handy to create more advanced scripts and controlling the server using PuTTY only. In the future I may switch to Linux-VM-servers due the terminal that is a powerful tool with knowledge and the power to separate each server over various VMs.

1. https://blog.desdelinux.net/sv/sl%C3%A4ppt-freebsd-9-0/ [↑](#endnote-ref-1)