2021 1dv512 Group Assignment 1

Group 30
Robin Svensson (rs223dj)
Saleh Shalabi (ss225bx)
Mattias Ståhlgren (ms225kw)

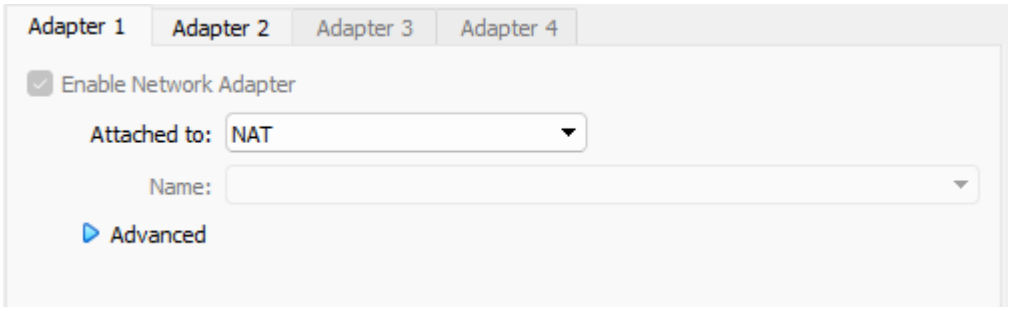
Task 1:

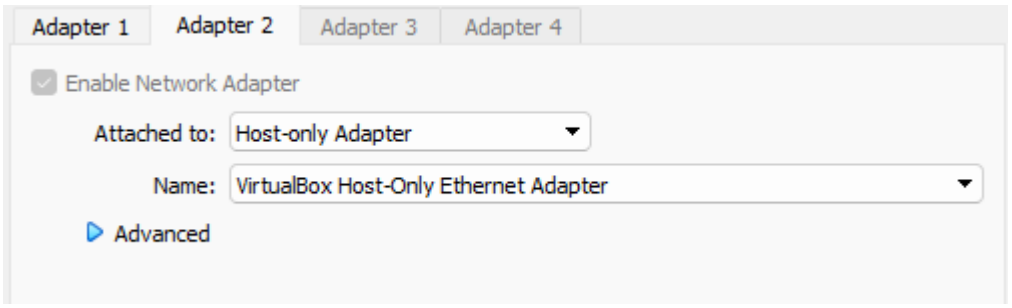
We decided to run with the Oracle VM VirtualBox Manager program for our virtual machine, with the host OS being Windows 11 (x86) using an intel processor aswell as intel graphics card.

Task 2:

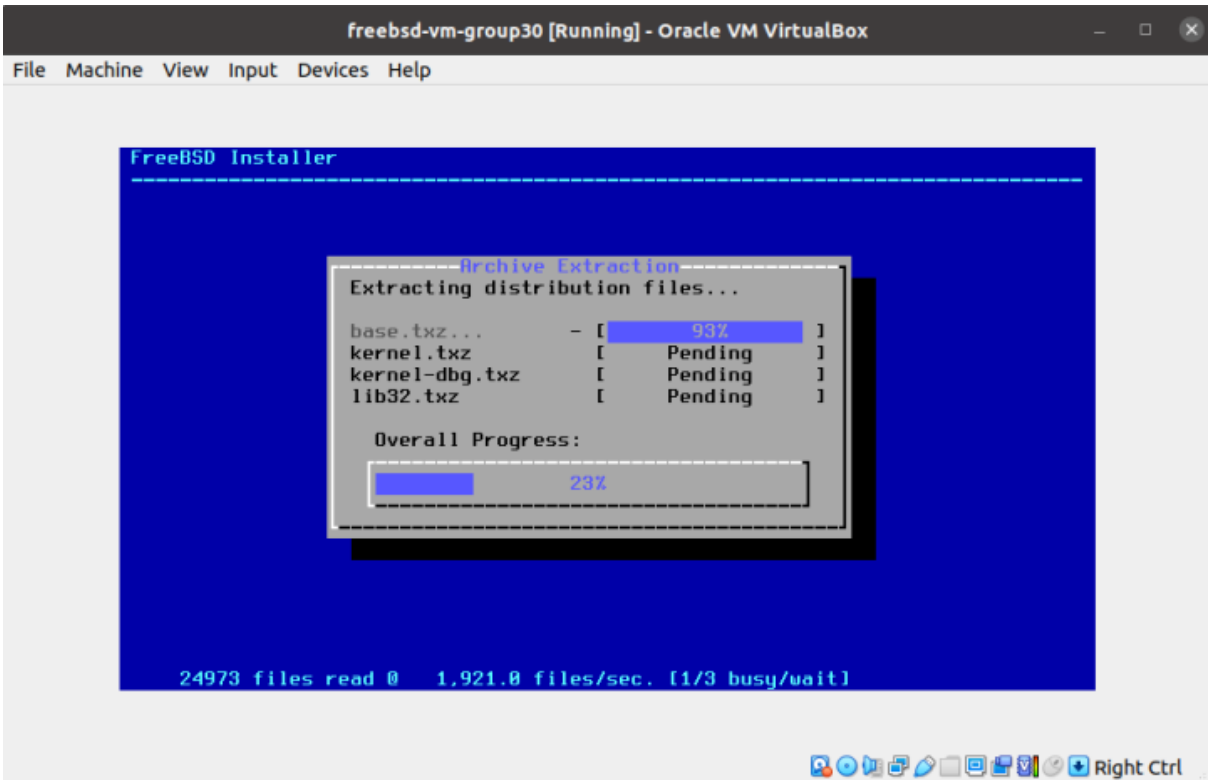
As for our FreeBSD installer we used the FreeBSD-13.0-RELEASE-amd64-bootonly version from their website.

Our configuration of FreeBSD was to allow 4gb of ram, 2 cpu cores allowed for performances and 16gb of storage space. Also setup one interface for the hostonly network, this then applied to the VM aswell as the NAT.

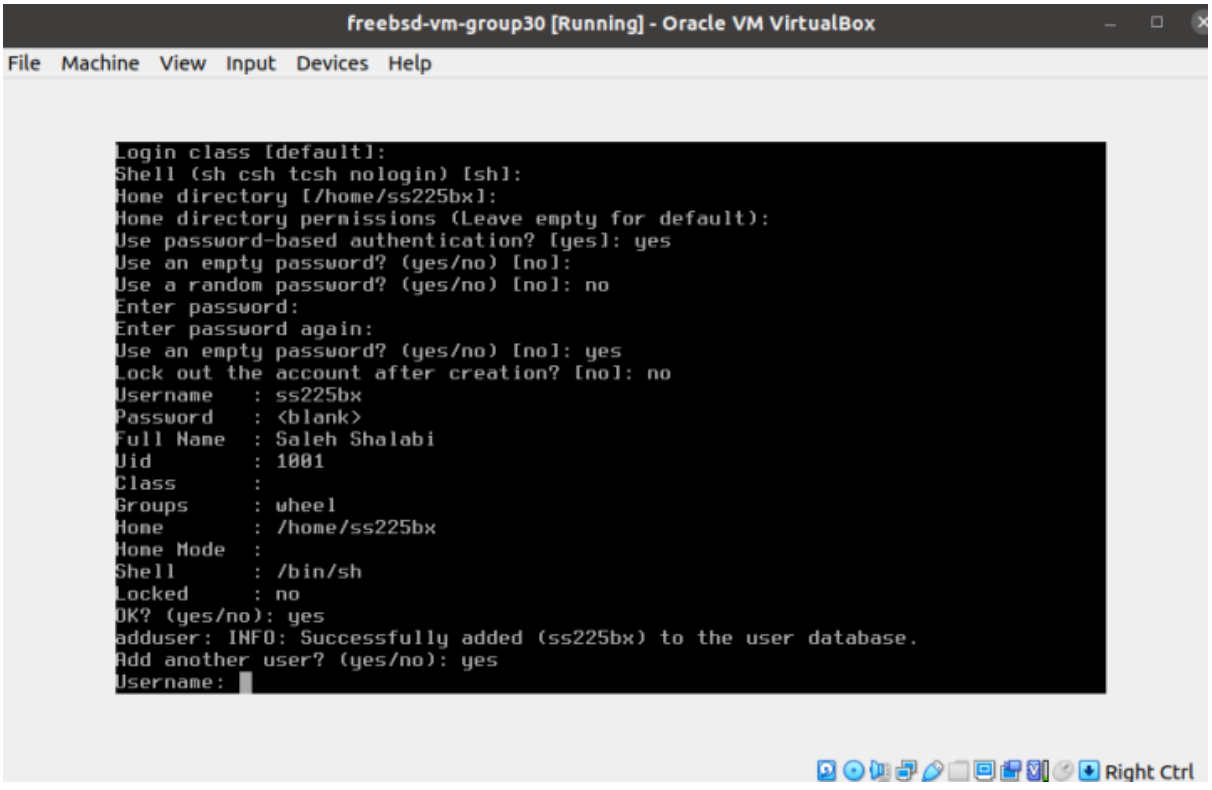


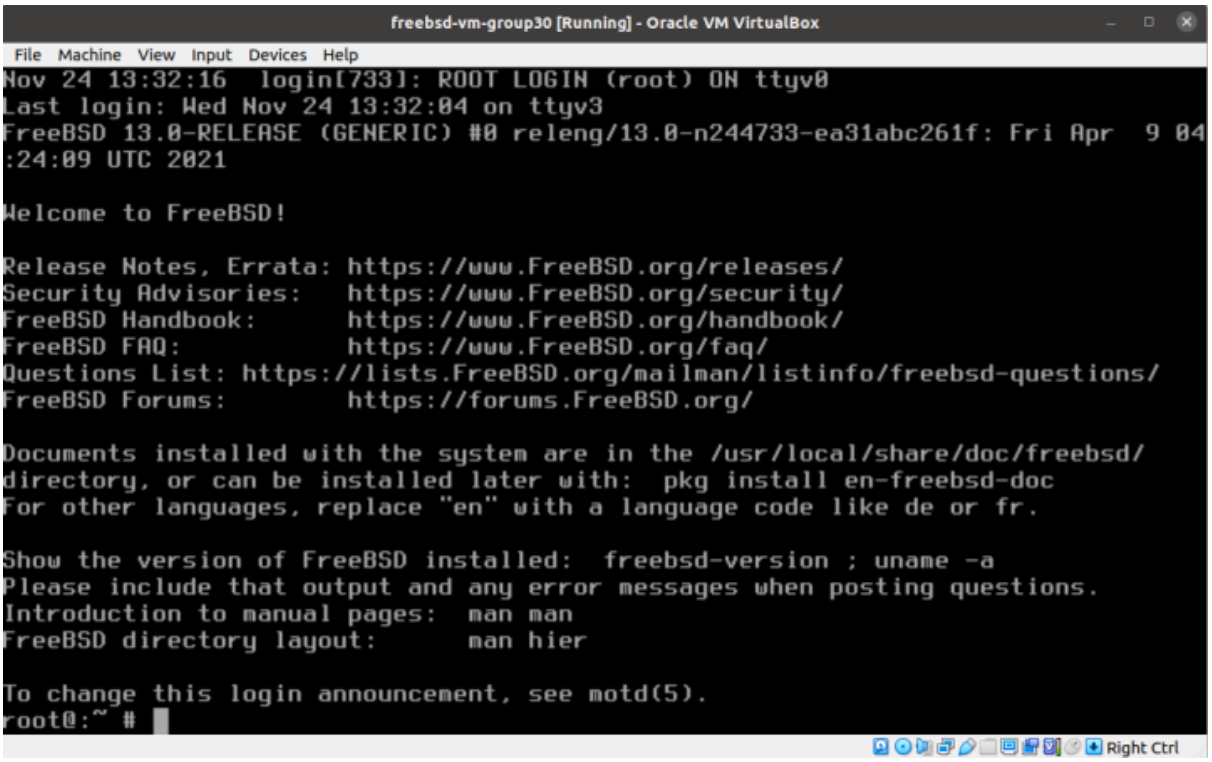


As for the installation of the OS, we needed a couple of attempts to get everything sorted. But the end result was as planned.



This was the user template that we used for our users, where only the main one was part of wheel.





Once we had finished the installation we managed to get access to the OS by logging into our root account.

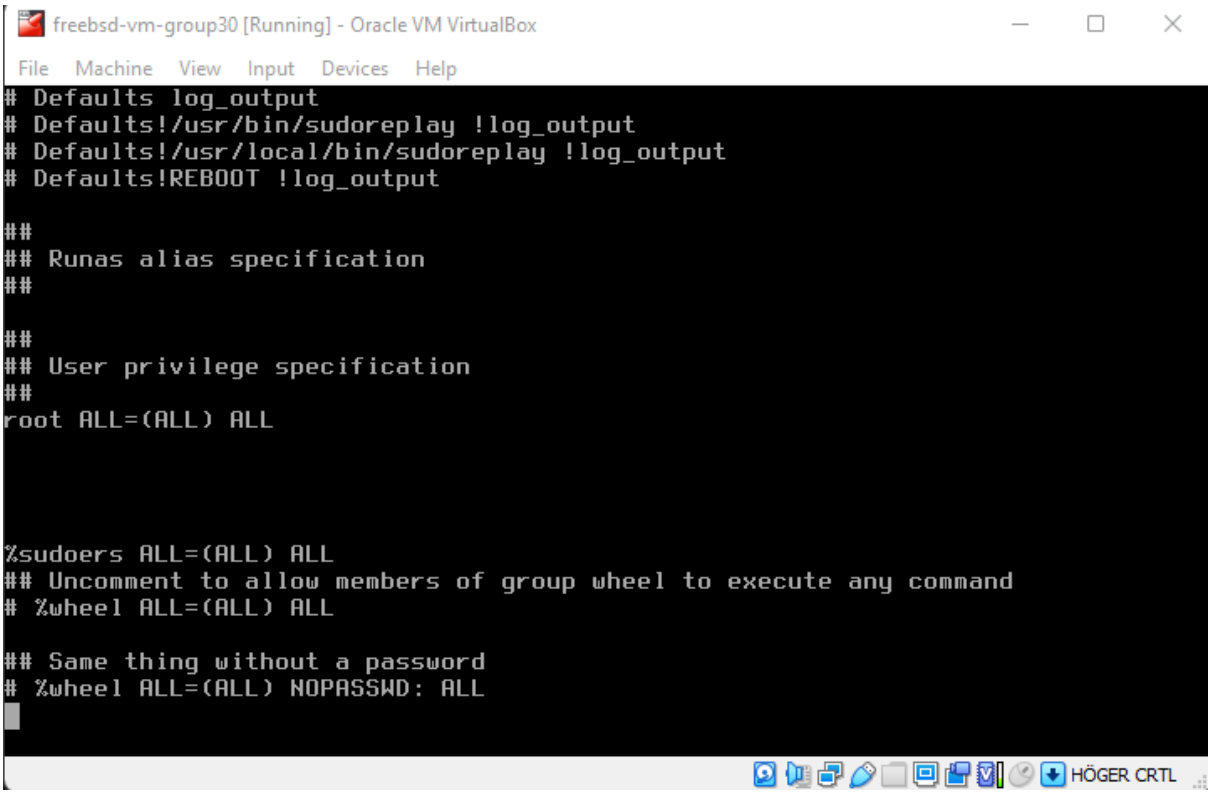
Task 3

As we had installed the OS we used the id the command to see the information about the user that was currently logged in, ([1] is representing the initial login but containing the sudoers group aswell) .

```
rs223dj@freebsd-vm-group30 /home/rs223dj/dv512-group30%: id
uid=1001(rs223dj) gid=0(wheel) groups=0(wheel),1004(sudoers)
```

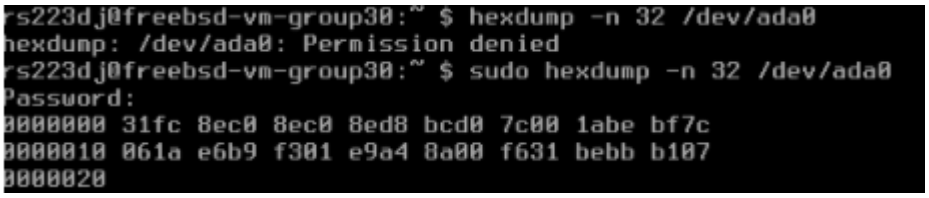
1. id command result

We tried to copy a .txt file from our home directory to another users directory but was shown a error message saying that we didn't have sufficient privileges to do such an action. Which led to us having to install the sudo package by logging into the superuser (root) and doing an pkg install command. After that we created the group seen in [1] called sudoers and added the [2] necessary privileges to allow the users to use superuser commands.



2. *Group sudoers privileges.*

To check whether this worked for us we ran the [3] hexdump command with the arguments -n 32 /dev/ada0, this as an normal user (rs223dj) gav us an error message regarding lacking privileges to create this hexdump. But when we ran the sudo command prior together with the command it gave us the hex values of that file that we asked for with a length of 32.



3. *hexdump with sudo and without.*

We then tried to copy over the files that we tried prior but using the sudo command between the home folder and the other user folder. This now worked as we now had superuser permissions [4].

```
rs223dj@freebsd-vm-group30:/usr/home/rs223dj $ sudo cp robin.txt ../ss225bx
rs223dj@freebsd-vm-group30:/usr/home/rs223dj $ cd ..
rs223dj@freebsd-vm-group30:/usr/home $ cd ss225bx
rs223dj@freebsd-vm-group30:/usr/home/ss225bx $ sudo ls
.cshrc          .login_conf     .mailrc         .shrc
.login          .mail_aliases   .profile        robin.txt
```

4. Copying robin.txt to ss225bx

Once this was done we tried to see if we could get a higher resolution in our FreeBSD VM, but this was a big problem depending on which type of graphics card you were currently using, as for the PC used in the document we had to add a line of code into our /etc/rc.conf to allow the adapter to be recognizeable by the system. After that we found how to fix the resolution to a higher one (1600x1200) [5].

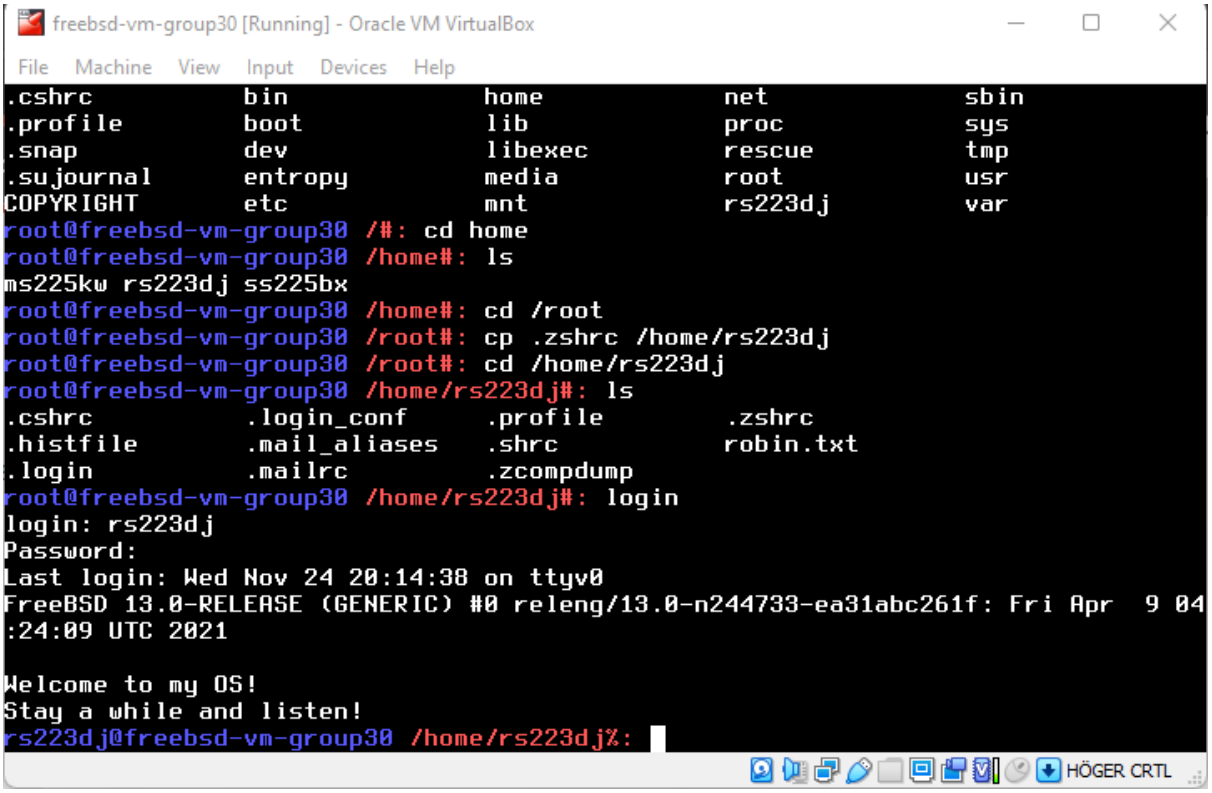
```
rs223dj@freebsd-vm-group30 /home/rs223dj%: vidcontrol show


 0 black       8                0 BACKGROUND     8 BACKGROUND
 1 blue        9 lightblue      1 BACKGROUND     9 BACKGROUND
 2 green      10 lightgreen     2 BACKGROUND    10 BACKGROUND
 3 cyan       11 lightcyan      3 BACKGROUND    11 BACKGROUND
 4 red        12 lightred       4 BACKGROUND    12 BACKGROUND
 5 magenta    13 lightmagenta   5 BACKGROUND    13 BACKGROUND
 6 brown      14 yellow         6 BACKGROUND    14 BACKGROUND
 7 white      15 lightwhite     7               15 BACKGROUND

rs223dj@freebsd-vm-group30 /home/rs223dj%: vidcontrol -i adapter
fb0:
    vga0, type:VESA VGA (5), flags:0x1700ff
    initial mode:24, current mode:327, BIOS mode:3
    frame buffer window:0xe0000000, buffer size:0x753000
    window size:0x753000, origin:0x0
    display start address (0, 0), scan line width:6400
    reserved:0xe0000000
rs223dj@freebsd-vm-group30 /home/rs223dj%:
```

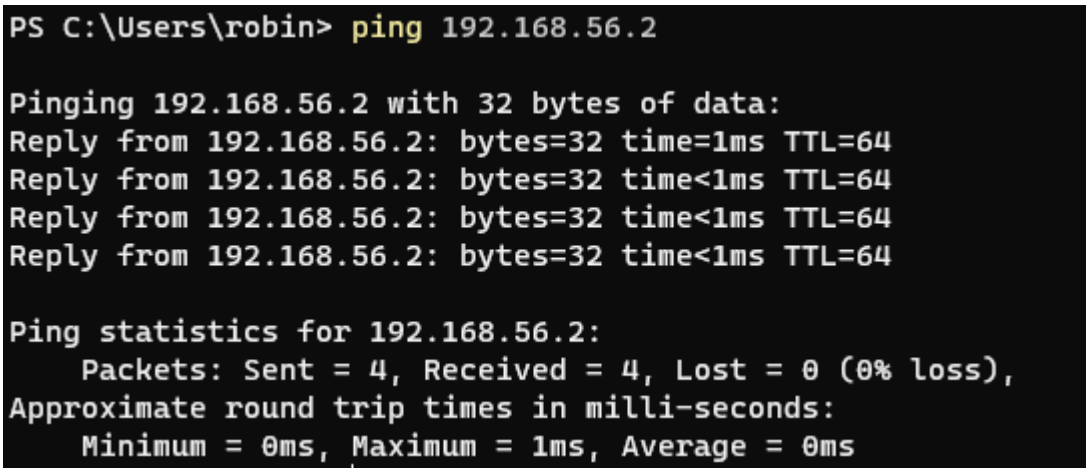
5. vidcontrol show and -i adapter

We then switched the shell used by our users to zsh by installing the package then heading into chsh and switching the shell to /usr/local/bin/zsh. After which we created a .zshrc file inside of our home folder and created a custom prompt as shown in [6].

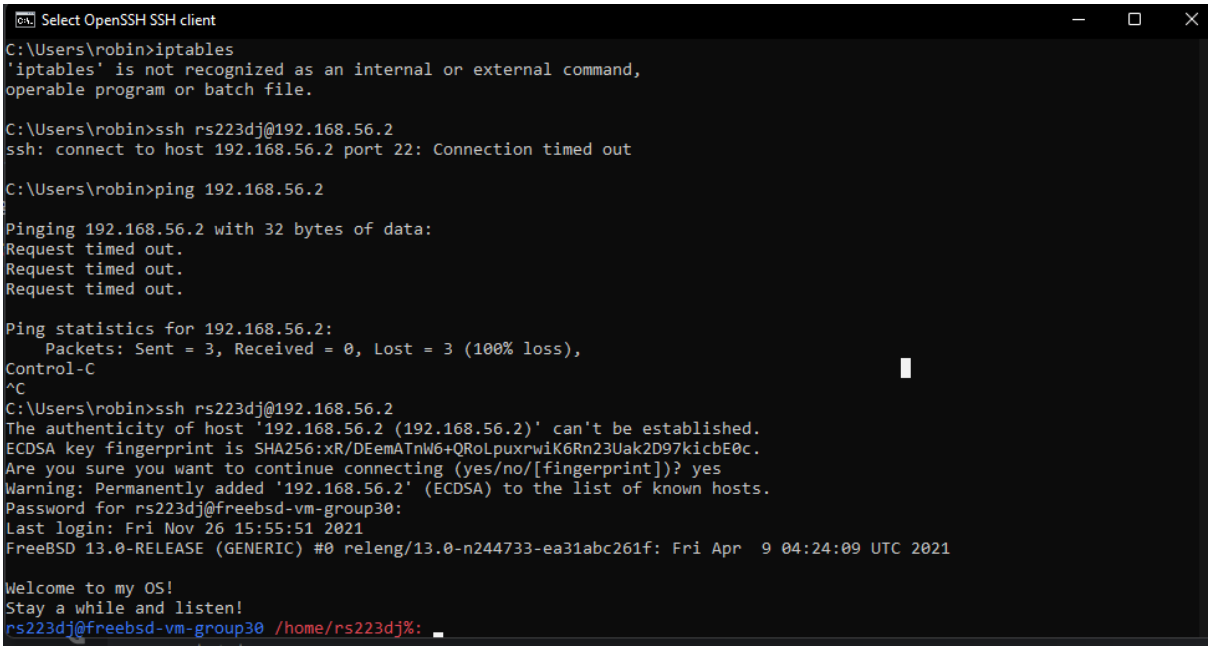


6. zsh shell with proper prompt

Once setup inside the zsh shell we needed to find a way to setup our SSH for the host system to be able to communicate with our VM system. This was done by using our second adapter (host-only) and setting up the static ip of that adapter. The static ip was changed by going into /etc/rc.conf and adding a new line ifconfig_em1=”192.168.56.2” [7] which after that we were able to ping to the system and from it while also being able to connect through our SSH shell in the host system[8].

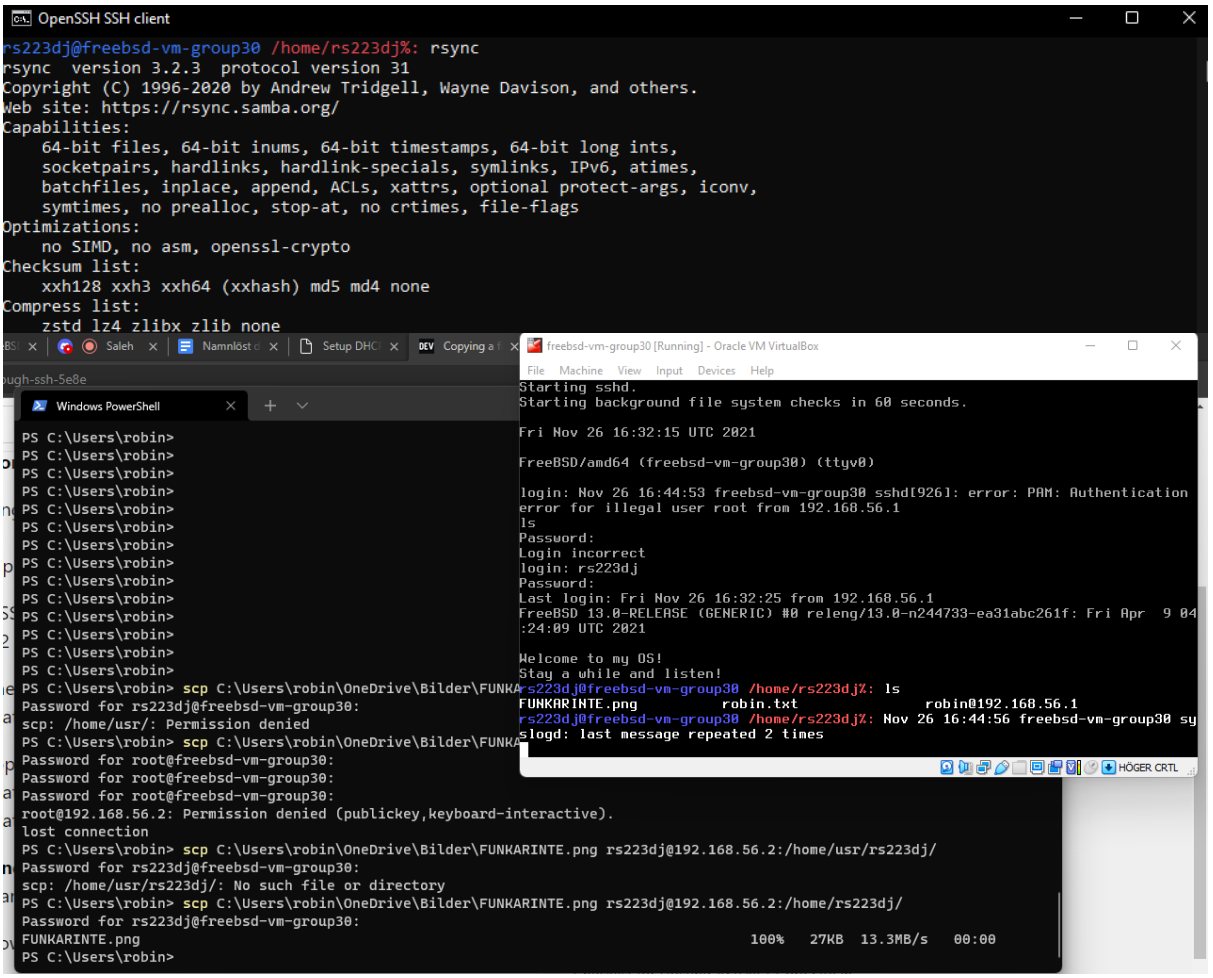


7. Pinging the system from host.



8. SSH connection to VM machine.

To be able to then use some of the files on our host system we needed to install rsync[9.0], this was done by simply using the pkg install rsync command. We then sent over a file to see if it was working correctly[9.1]



9. *(9.0 rsync installed) (9.1 file sent)*

As a final part of the task we installed java and wrote a bit of code to perform a few actions and get the returned value printed out in the command line. Java was installed as prior with the pkg install command[10]

```
rs223dj@freebsd-vm-group30 /home/rs223dj%: java -version
openjdk version "11.0.12" 2021-07-20
OpenJDK Runtime Environment (build 11.0.12+7-1)
OpenJDK 64-Bit Server VM (build 11.0.12+7-1, mixed mode)
rs223dj@freebsd-vm-group30 /home/rs223dj%:
```

10. *Currently installed java version*

We then first created a simple program that ran the id command[11] and then reworked that code to work in a reusable way to allow easy creation of more commands. Where we then used the “find /etc/ -name ‘rc*’ command and printed out the results aswell as its exitcode[11].

```
rs223dj@freebsd-vm-group30 /home/rs223dj/dv512-group30%: java App
the Command used is id
Result:
uid=1001(rs223dj) gid=0(wheel) groups=0(wheel),1004(sudoers)
Exit code: 0
the Command used is find /etc/ -name 'rc*'
Result:
/etc/rc.initdiskless
/etc/rc.sendmail
/etc/rc.firewall
/etc/rc.d
/etc/rc.d/rctl
/etc/rc
/etc/rc.suspend
/etc/rc.bsdextended
/etc/rc.conf.d
/etc/rc.shutdown
/etc/rc.subr
/etc/defaults/rc.conf
/etc/rc.resume
/etc/rc.conf
Exit code: 1
the Command used is hostname freebsd-vm-group30-upd
Result:
Exit code: 1
the Command used is hostname
Result:
freebsd-vm-group30-upd
Exit code: 0
rs223dj@freebsd-vm-group30 /home/rs223dj/dv512-group30%: sudo java App
the Command used is id
Result:
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
Exit code: 0
the Command used is find /etc/ -name 'rc*'
Result:
/etc/rc.initdiskless
/etc/rc.sendmail
/etc/rc.firewall
/etc/rc.d
/etc/rc.d/rctl
/etc/rc
/etc/rc.suspend
/etc/rc.bsdextended
/etc/rc.conf.d
/etc/rc.shutdown
/etc/rc.subr
/etc/defaults/rc.conf
/etc/rc.resume
/etc/rc.conf
Exit code: 0
the Command used is hostname freebsd-vm-group30-upd
Result:
Exit code: 0
the Command used is hostname
Result:
freebsd-vm-group30-upd
Exit code: 0
```

11. *App ran as sudo and not.*

From the program we could see that based on whether or not we used sudo to run the java program we got different exitcodes. We understood this as we didn't actually have permission to view the files if ran without the sudo command but once we ran it as a superuser we got an exitcode of 0 everytime.

Discussion, reflection and comments

We feel as though it has been a very interesting task to complete, but not very clear as to why we did the things we did. There was also alot of bumps in the road as it seems all information about the OS is made using UNIX and we were using Windows, also having different types of graphics cards forced us to install drivers etc which didn't seem clear from the task information.

One thing that has been very informative is how exact you have to be in terms of ur work to make sure everything is working correctly, changing the location of something just a tad bit will make certain stuff not work, not having a " at the end stops ur entire shell from booting etc.

Work distribution

Our work distribution has been good, Robin wrote the document while having discussions with the group, the task were done together on different PC but making sure everyone kept up and understood what was happening at all times. Overall a balanced work distribution in terms of tasks.