Table Of Contents

Installation of Linux OS using VM

If you want to use linux without making changes to you current windows system, you can use `VM` (virtual manager) to achieve this goal. There are several software which can be used to run *guest operating system*. Some of them are:

- Virtual Box
- VMWare
- QEMU

We'll focus on Virtual Box here.

`VirtualBox` is free and open source virualization software from Oracle.

Requirments

- Good Internet Connection to download ISO(most of them are around size of **2GB**)
- Minimum RAM of around 4GB for windows(should have 8GB) although it'll depends on which distro you're trying to install
- Host system with atleast 8 - 10GB of disk space.
- Should enable hyper visualization from your **BIOS(UEFI)** setting

Steps

Here are the steps to install linux OS on vm

Download and Install VirtualBox

Go to the website of Oracle VirtualBox and get the latest stable version according to your host system.

Download the Linux ISO

Next, you need to download the ISO file of the linux distribution which you want from it's official website.

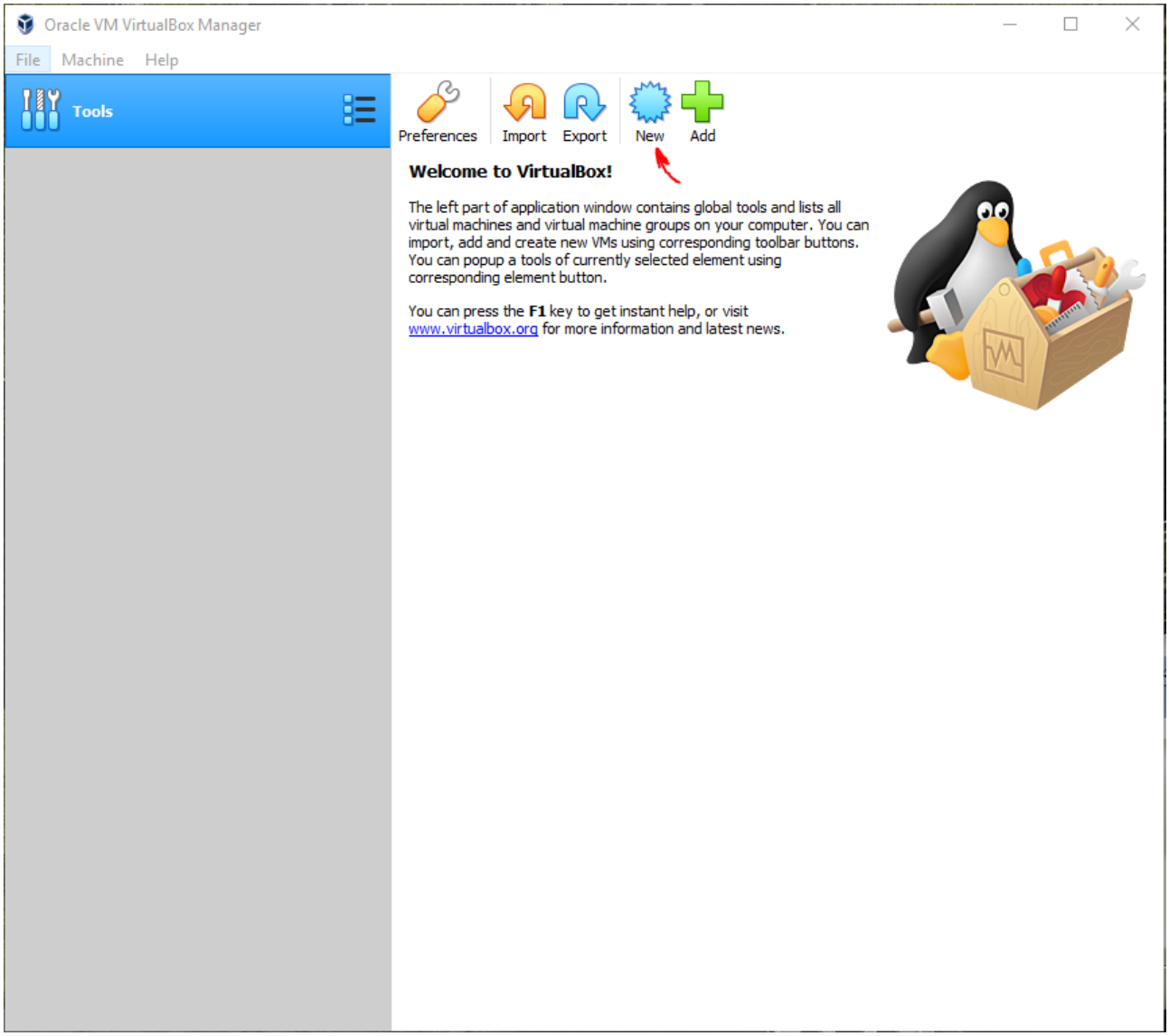
There are several types of linux/unix based distros out there, some are:

- Debian based
  - Debian
  - Ubutnu
  - Mint
  - Elementary
- Arch based
  - Arch
  - Manjaro
  - Void
  - Antergos
- NixOS
- Fedora
- RedHat
- OpenSuse
- Gentoo
- BSD
  - FreeBSD
  - OpenBSD etc.

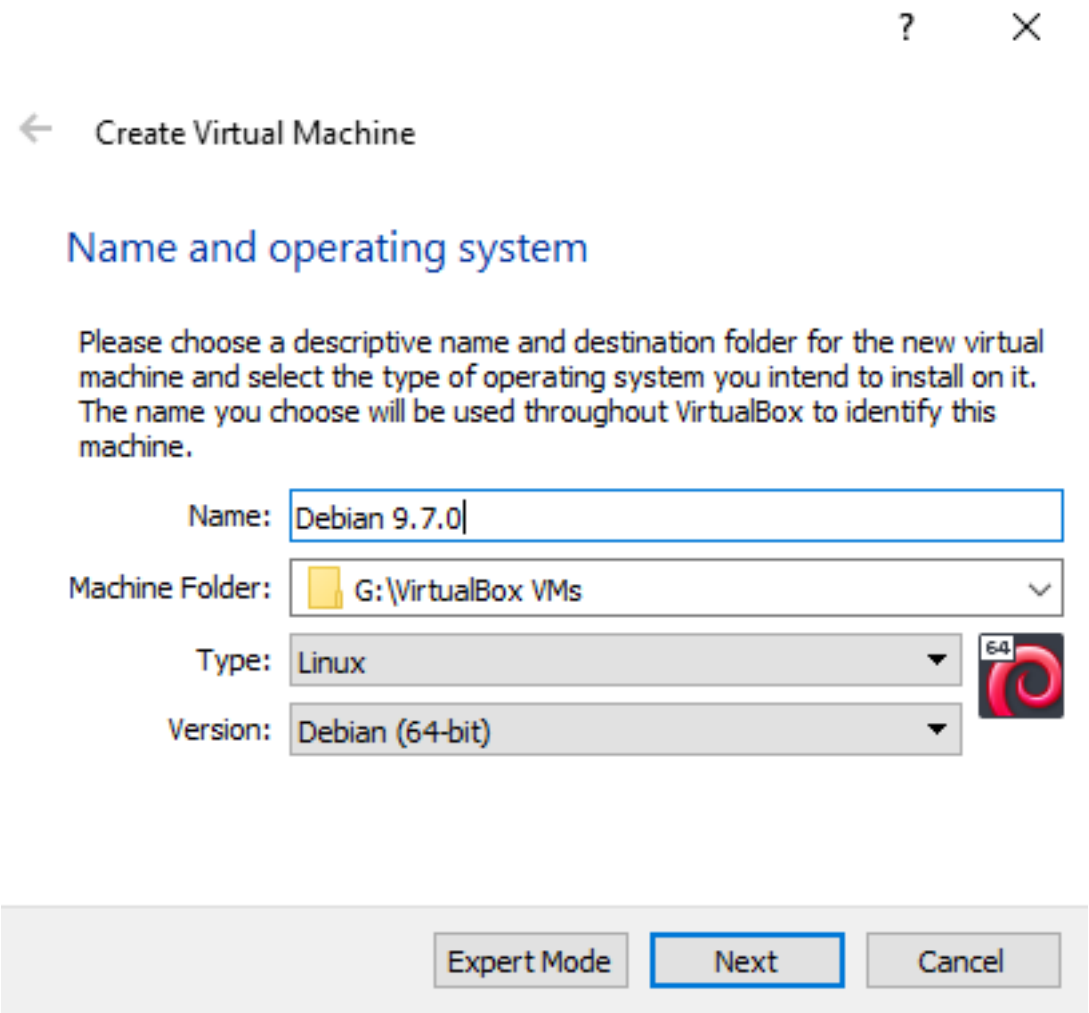
I'm using `Debian` in here as example. Get it from the following link `https://www.debian.org/`

Whatever distribution you downloaded it's highly recommended to check its checksum(md5, sha256) etc.

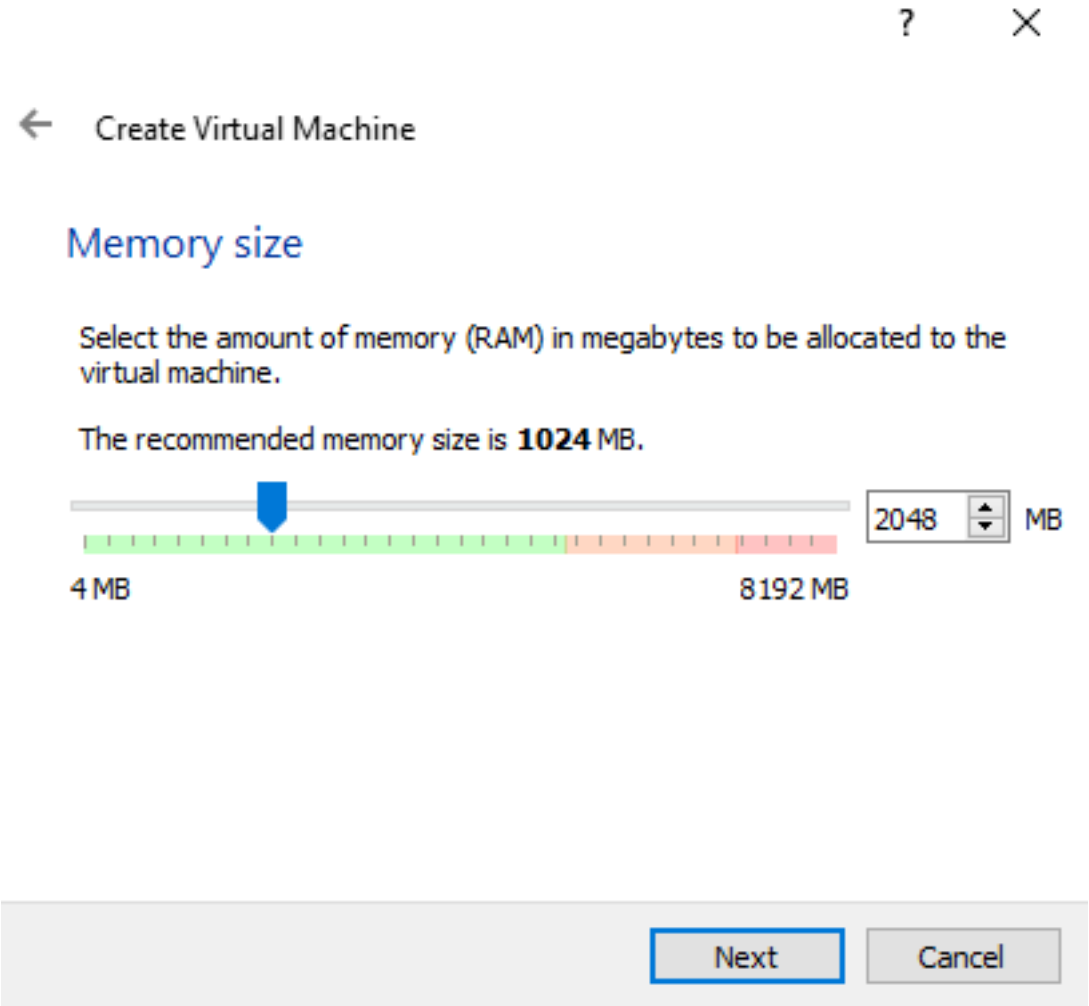
Virtual Machine Creation



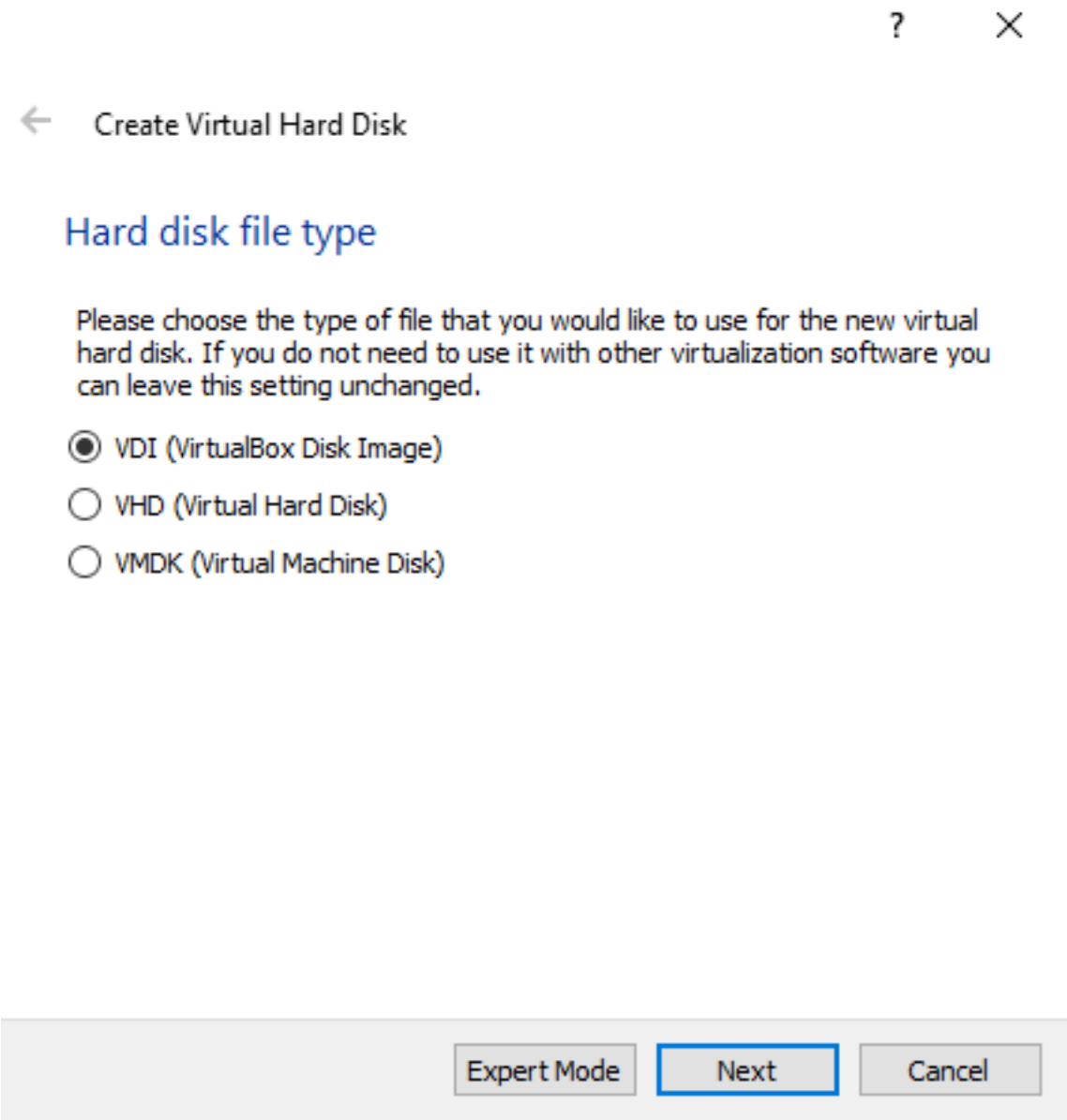
Click **New** button to create VM



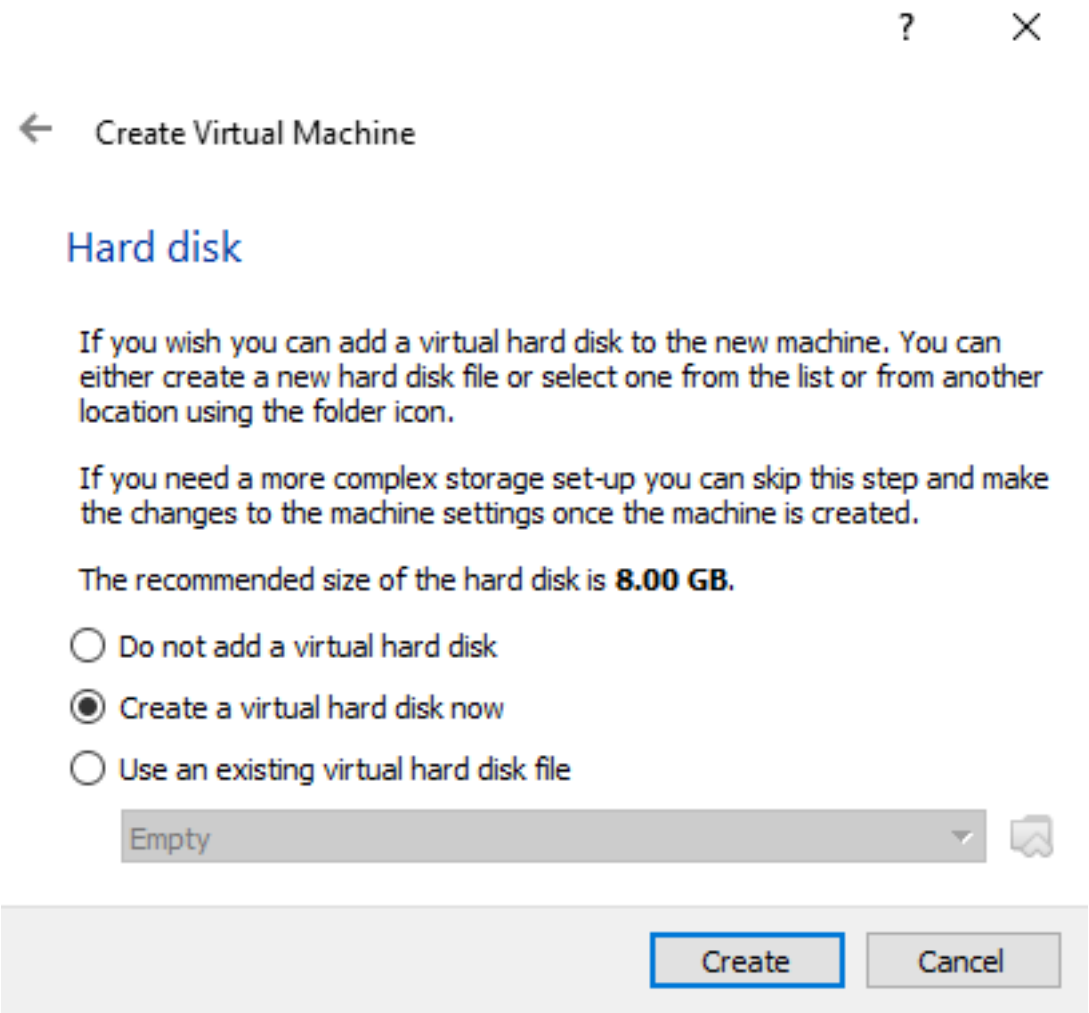
Provide a **name** of your choice to the new VM and a **directory location** to store the related files. Also give type as **Linux** and type of **distro**.



Give a memory size to your VM. Minimum for debian is around 512GB, also you shouldn't allocate more than half of RAM size what the host have.



Choose **Create a virtual hard disk now** option and click **Create**.



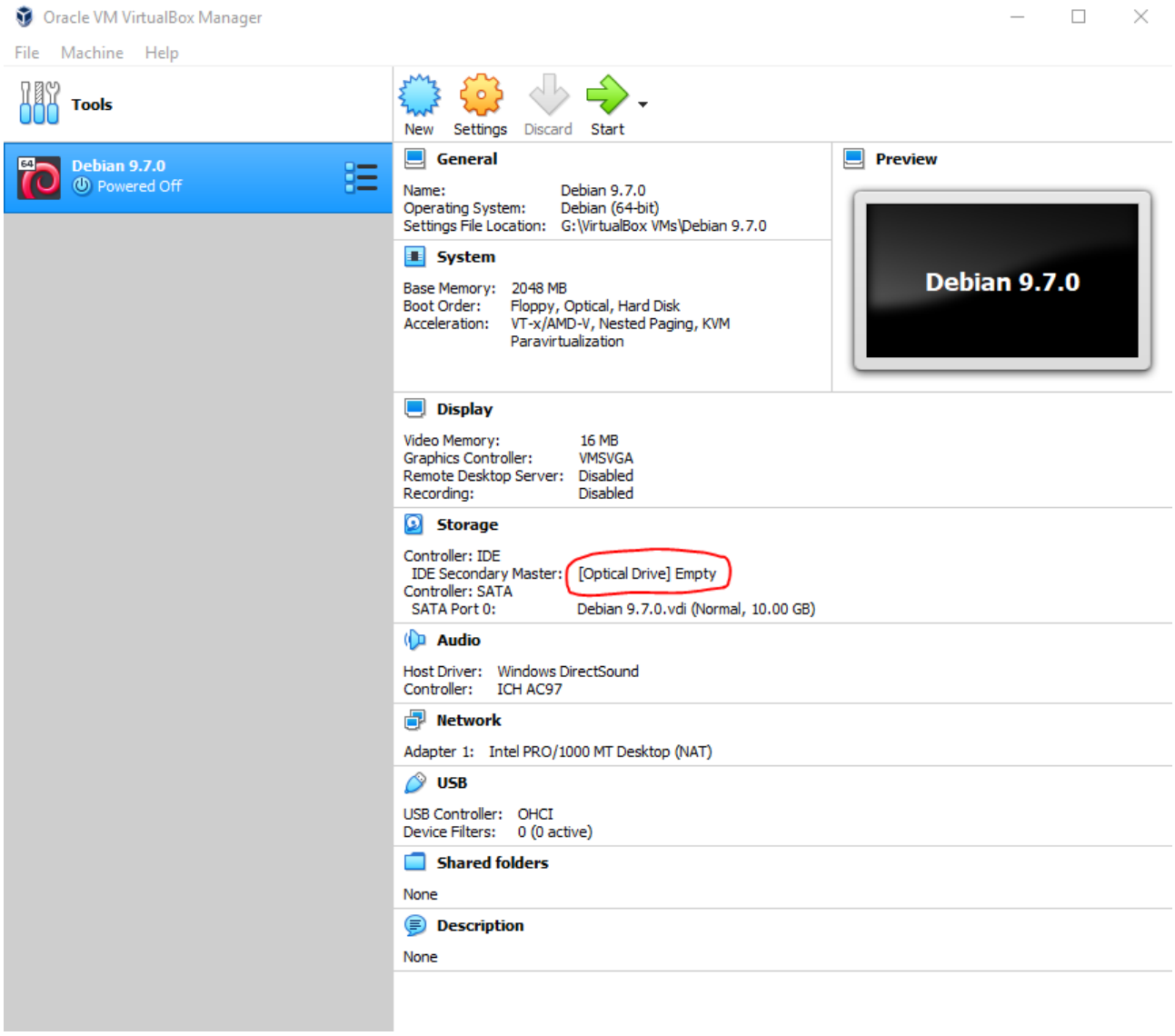
Choose **VDI** option and click **Next.**

Choose **Dynamically Allocated** option and click **Next**.

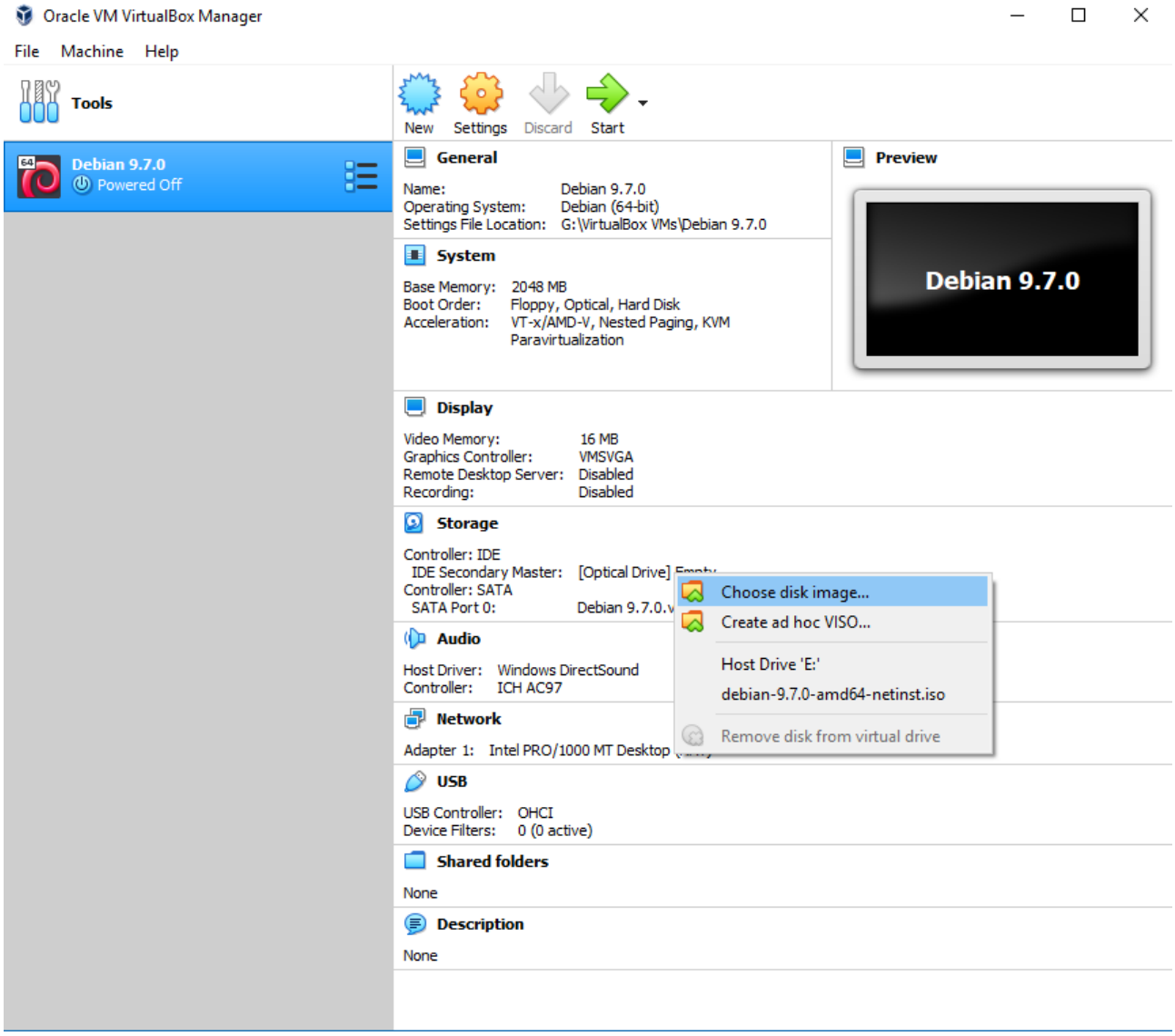
Provide a **Hard Disk size** for your VM and click **Create**.

Debian Installation

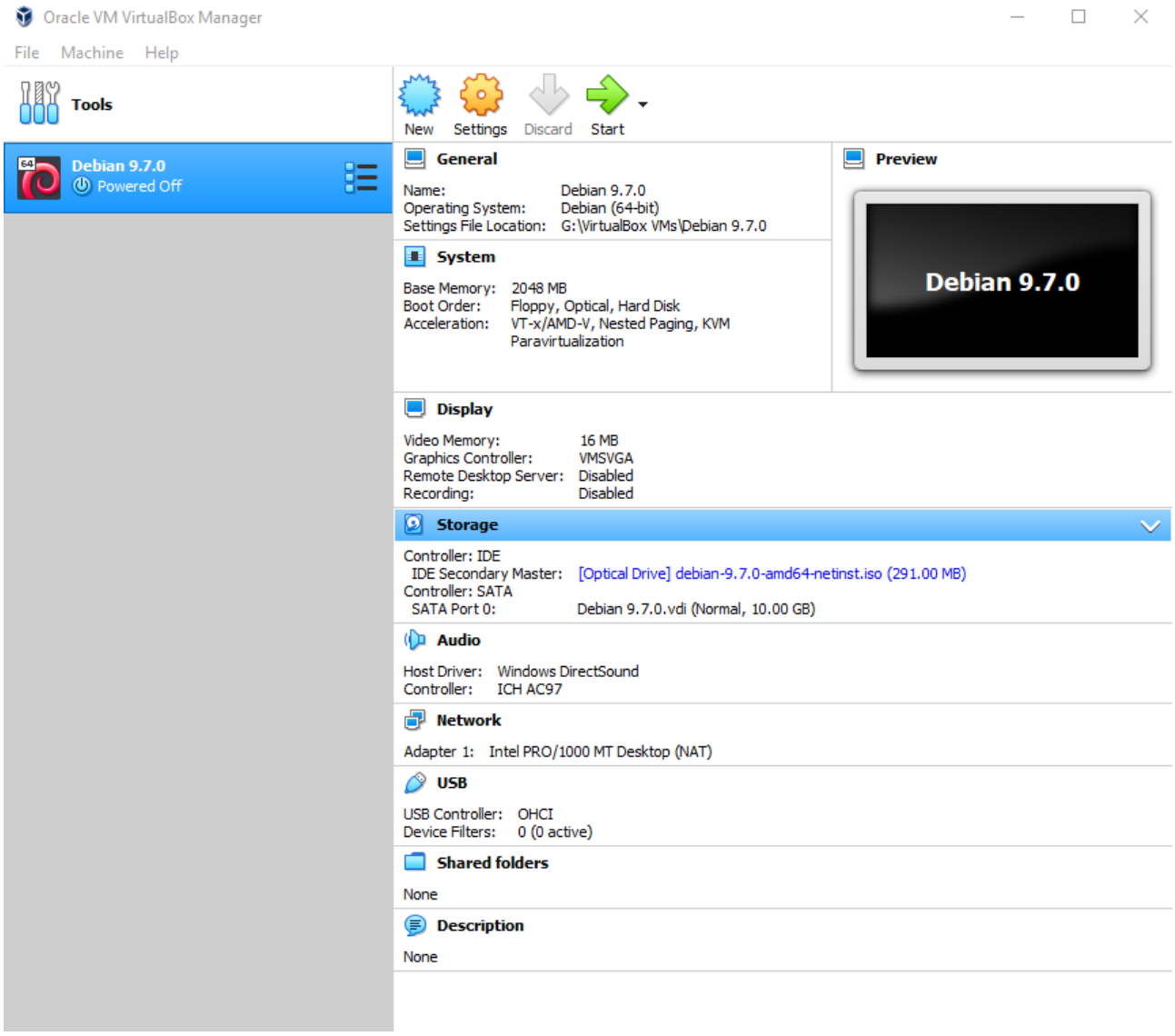
Provide the downloaded Debian ISO image to the newly created VM.



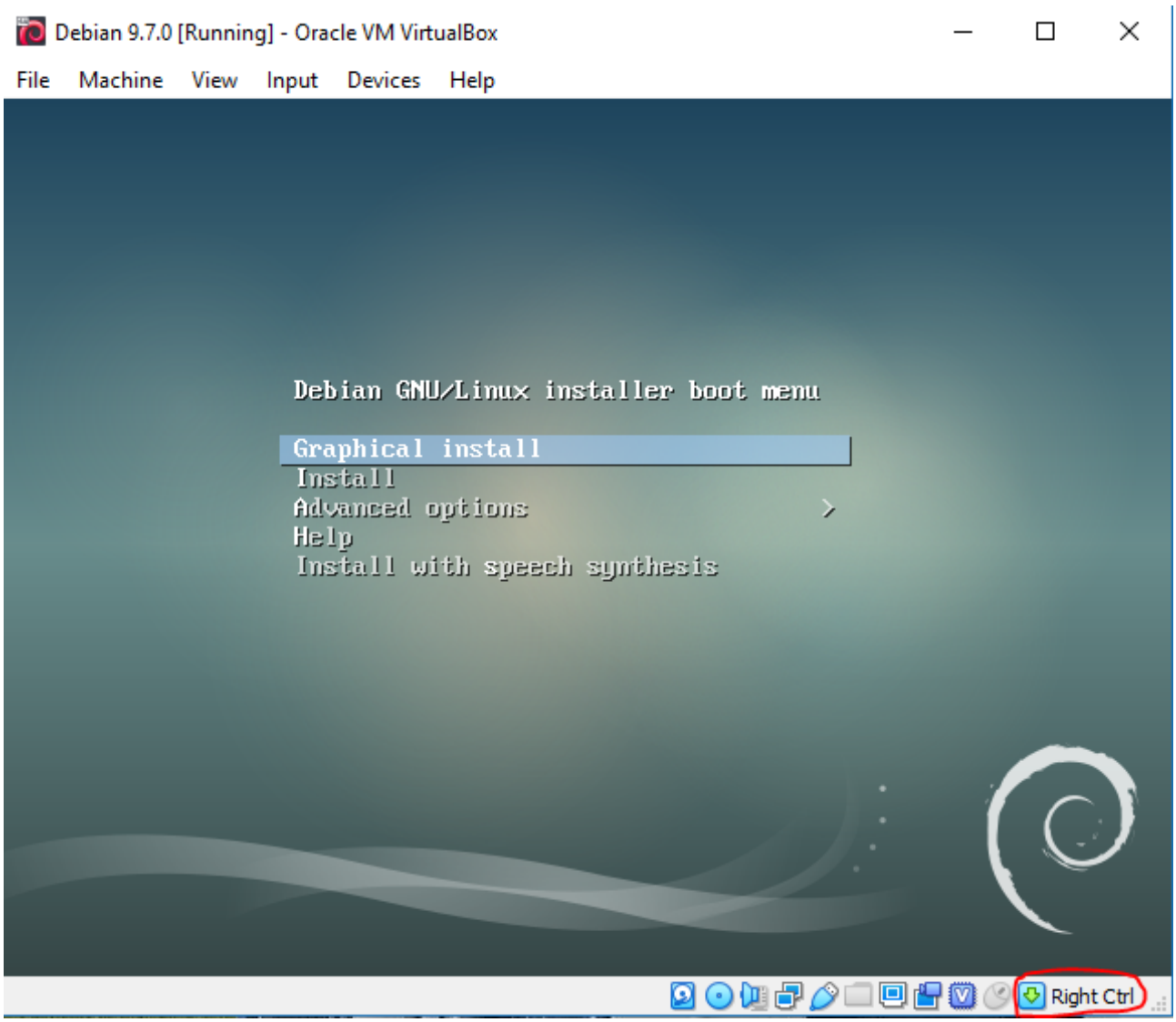
Now, click on [**Optical Drive**] **Empty**



Select **Choose disk image...** and give it you ISO location



Now things will look like this. Click **start** button to start your VM.



This here is `grub menu` from here, you can start to install your linux os on VM.

Refer to this website https://medium.com/platform-engineer/how-to-install-debian-linux-on-virtualbox-with-guest-additions-778afa0ee7e0 for more info.

Practicing Linux Commands

Introduction

- All the linux commands are run in terminal(exceptions are there.)
- There are severel types of terminal in linux, like:
  - termite
  - alacritty
  - konsole
  - gnome-terminal
  - xterm
  - urxvt
  - kitty
  - terminator etc.
- Linux commands are *case-sensitive*.
- The terminal can be used to accomplish all administrative tasks. This includes:
  - Package Installation
  - File editing
  - File manipulation
  - User and group management and many other things

General Commands

I'm going to give some screenshots of general commands (acc. to the syllabus). These commands are:

- **date** - displays the current system date and time
- **cal** - display calender of a specific month or a whole year
- **clear** - used to clear the terminal
- **who** - displays the information about all currently logged in user on the system
- **whoami** - display the username of current user
- **exit** - exits the shell/terminal
- **history** - displays the previously executed commands
- **bc** - start command line calculator
- **alias** - used to give user defined name to a command or sequence of commands
- **shutdown** - used to shutdown the system
- **reboot** - restart or reboot the system
- **banner** - prints string in large ascii character set

File Related Commands

- **mkdir** - used to create directory/ies
- **rmdir** - uset to remove blank directory
- **cat** - display content of file, also used to overwrite or append content of file
- **ls** - used to list files and directory/ies
- **rm** - used to remove files from system
- **pwd** - prints current working directory
- **find** - finds different types of files and directories in system
- **gzip** - compresses file/s in *.gz* format
- **gunzip** - uncompress the *.gz* format
- **wc** - used to count letters, words and line from file or stdin
- **cd** - used to change directory
- **mv** - used to move files/directories, same as cut in windows

Here, are few screenshots of the above mentioned commands:

```
raytracer  ~/.cache/temp   master M   date
Sun 17 May 2020 09:33:30 AM IST
raytracer  ~/.cache/temp   master M   cal
       May 2020
Su Mo Tu We Th Fr Sa
                   1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
raytracer  ~/.cache/temp   master M   cal 5 2020
       May 2020
Su Mo Tu We Th Fr Sa
                   1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
raytracer  ~/.cache/temp   master M
```

```
raytracer  ~/.cache/temp   master M   cal 2020                                                    NORM
                                2020

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1    1  2  3  4  5  6  7
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    8  9 10 11 12 13 14
12 13 14 15 16 17 18    9 10 11 12 13 14 15   15 16 17 18 19 20 21
19 20 21 22 23 24 25   16 17 18 19 20 21 22   22 23 24 25 26 27 28
26 27 28 29 30 31      23 24 25 26 27 28 29   29 30 31

        April                  May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                   1  2      1  2  3  4  5  6
 5  6  7  8  9 10 11    3  4  5  6  7  8  9    7  8  9 10 11 12 13
12 13 14 15 16 17 18   10 11 12 13 14 15 16   14 15 16 17 18 19 20
19 20 21 22 23 24 25   17 18 19 20 21 22 23   21 22 23 24 25 26 27
26 27 28 29 30         24 25 26 27 28 29 30   28 29 30
                       31
        July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1          1  2  3  4  5
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    6  7  8  9 10 11 12
12 13 14 15 16 17 18    9 10 11 12 13 14 15   13 14 15 16 17 18 19
19 20 21 22 23 24 25   16 17 18 19 20 21 22   20 21 22 23 24 25 26
26 27 28 29 30 31      23 24 25 26 27 28 29   27 28 29 30
                       30 31
       October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7          1  2  3  4  5
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    6  7  8  9 10 11 12
11 12 13 14 15 16 17   15 16 17 18 19 20 21   13 14 15 16 17 18 19
18 19 20 21 22 23 24   22 23 24 25 26 27 28   20 21 22 23 24 25 26
25 26 27 28 29 30 31   29 30                   27 28 29 30 31
```

```
raytracer  ~/.cache/temp  master M  who                                                INS
raytracer tty7         2020-05-16 09:36 (:0)
raytracer  ~/.cache/temp  master M  whoami                                             INS
raytracer
raytracer  ~/.cache/temp  master M  history | head                                     INS
 4997  bspc node -t \~fullscreen
 4998  v /home/raytracer/.config/sxhkd/sxhkdrc
 5000  ranger
 5009  pandoc -V geometry:margin=1in -o os_assign.pdf os_assign.md
 5010  zathura os_assign.pdf
 5012  who am i
 5016  fg
 5017  ~/.cache/temp
 5018  date
 5019  cal
raytracer  ~/.cache/temp  master M  bc                                                 INS
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
quit
raytracer  ~/.cache/temp  master M  please                                             INS

  This is why nobody likes you.

zsh: command not found: please
zsh: exit 127   please
raytracer  ~/.cache/temp  master M  alias please=pwd                                127 INS
raytracer  ~/.cache/temp  master M  please                                             INS
/home/raytracer/.cache/temp
raytracer  ~/.cache/temp  master M                                                     INS
```

```
raytracer  ~/.cache/temp  master M  banner linux                                       INS

 **
  *      *
  *
  *     **    * ***   *    *  *    *
  *      *    **   *  *    *   *  *
  *      *    *    *  *    *    **
  *      *    *    *  *    *    **
  *      *    *    *  *   **   *  *
*****  *****  *    *   *** *  *    *


raytracer  ~/.cache/temp  master M  mkdir A                                            INS
raytracer  ~/.cache/temp  master M  mkdir X Y Z                                        INS
raytracer  ~/.cache/temp  master M  ls | grep /                                        INS
A/
assignment-march2020/
learn_awk/
wd_hw/
X/
Y/
Z/
raytracer  ~/.cache/temp  master M  mkdir -p D/E/F                                     INS
raytracer  ~/.cache/temp  master M  ls -r                                              INS
Z/                          ques7.py                        hello.download.png
Y/                          python_ictiitk.html             hello.download-removebg-preview.png
X/                          play_url.txt                    hello.blurred.png
wd_hw/                      msu                             hello.587508.png
learn_awk/                  ls_usr_bin.txt                  hello.2020-04-13-063935_1134x302_scrot.png
D/                          list_pack1.txt                  hello.6n7h9uS.png*
assignment-march2020/       list_pack.txt                   getcolor.txt
A/                          john-abraham__5461341.jpg       first.md
```

```
raytracer ~/.cache/temp  master M  ls -R D                                                INS
E/

D/E:
F/

D/E/F:
raytracer ~/.cache/temp  master M  cat "hello" > abc.txt                                   INS
cat: hello: No such file or directory
zsh: exit 1     cat "hello" > abc.txt
raytracer ~/.cache/temp  master M                                                       1  INS
raytracer ~/.cache/temp  master M  touch abc.txt                                        1  INS
raytracer ~/.cache/temp  master M  cat "hello" > abc.txt                                   INS
cat: hello: No such file or directory
zsh: exit 1     cat "hello" > abc.txt
raytracer ~/.cache/temp  master M                                                       1  INS
raytracer ~/.cache/temp  master M  cat > abc.txt                                        1  INS
Hey, this is linux
^C
zsh: interrupt  cat > abc.txt
raytracer ~/.cache/temp  master M  cat abc.txt                                        130  INS
Hey, this is linux
raytracer ~/.cache/temp  master M  cat >> abc.txt                                          INS
Name of the shell is zsh.
raytracer ~/.cache/temp  master M  cat abc.txt                                            NORM
Hey, this is linux
Name of the shell is zsh.
raytracer ~/.cache/temp  master M  cat > abc.txt                                           INS
I'm abhay
raytracer ~/.cache/temp  master M  cat abc.txt                                            NORM
I'm abhay
raytracer ~/.cache/temp  master M                                                          INS
```

```
raytracer ~/.cache/temp  master M  find . -type d -iname 'D'
./D
raytracer ~/.cache/temp  master M  find ~/Documents/nhtml -type f -iname '*html*' | head
/home/raytracer/Documents/nhtml/multiple_file3.html
/home/raytracer/Documents/nhtml/multiple_file.html
/home/raytracer/Documents/nhtml/first.html
/home/raytracer/Documents/nhtml/live_class/id-selector.html
/home/raytracer/Documents/nhtml/live_class/2020-05-11(1).html
/home/raytracer/Documents/nhtml/live_class/2020-05-14(2).html
/home/raytracer/Documents/nhtml/live_class/loop_js.html
/home/raytracer/Documents/nhtml/live_class/2020-05-07.html
/home/raytracer/Documents/nhtml/live_class/2020-05-04.html
/home/raytracer/Documents/nhtml/live_class/2020-04-30.html
zsh: broken pipe  find ~/Documents/nhtml -type f -iname '*html*' |
zsh: done         head
raytracer ~/.cache/temp  master M  find ~/Documents/nhtml -type f -iname '*html*' | wc -l
96
```

```
raytracer ~/.cache/temp  master M  gzip -N a.out                                           INS
raytracer ~/.cache/temp  master M  ls | grep gz                                            INS
a.out.gz*
raytracer ~/.cache/temp  master M  gzip -d a.out.gz                                        INS
raytracer ~/.cache/temp  master M  ls | grep .out                                          INS
a.out*
raytracer ~/.cache/temp  master M  cd D                                                    INS
E/
raytracer ~/.cache/temp/D  master M  cd E                                                  INS
F/
raytracer ~/.cache/temp/D/E  master M  cd ../..                                            INS
A/                      first.md                                        info.txt                        Untitled-1.yaml
assignment-march2020/   getcolor.txt                                    john-abraham__546134.jpg*       wificard_info.txt
D/                      hello.6n7h9uS.png*                              john-abraham__5461341.jpg       x.sh*
learn_awk/              hello.2020-04-13-063935_1134x302_scrot.png      list_pack.txt                   y.sh*
wd_hw/                  hello.587508.png                                list_pack1.txt
X/                      hello.blurred.png                               ls_usr_bin.txt
Y/                      hello.download-removebg-preview.png             msu
Z/                      hello.download.png                              play_url.txt
3+calc.sh               hello.images-removebg-preview.png               python_ictiitk.html
671965.jpg*             hello.images.png                                ques7.py
a.out*                  hello.john-abraham-wallpapers-hd-67491-9249295.png*  ques11.py
character_counting.c    hello.lock-xxl.png                              send_to_ayush.txt
compiler.sh*            hello.lock2-removebg-preview.png                shcheck.sh*
cssassignment.zip       hello.lock2.png                                 st-copyurl-20190202-3be4cf1.diff
diff_cron_norm_env.txt  hello.new_image.png                             test.rb
equal_stack_hackerrank.c hello.wallpaper.png                            trial*
fact.py*                hello.yellow-and-gray-padlock-vector-art-png-clip-art.png  trial.c
raytracer ~/.cache/temp  master M  cd D/E                                                  INS
F/
raytracer ~/.cache/temp/D/E  master M  cd ~                                                INS
Documents/  Downloads/  go/  Music/  Pictures/  R/  Templates/  Videos/  vimwiki/  yay/  canyonbottom  README.md
raytracer ~  master M ??                                                                   INS
```

```
raytracer  ~   master M ?? cd -
~/.cache/temp/D/E
F/
raytracer  ~/.cache/temp/D/E   master M  mv F ../
raytracer  ~/.cache/temp/D/E   master M  ls
raytracer  ~/.cache/temp/D/E   master M  ..
E/  F/
raytracer  ~/.cache/temp/D   master M
```

There are several arguments, options and flags in almost every linux command. To know more you can do:

```
find(1)
FIND(1)                                General Commands Manual                                FIND(1)

NAME
       find - search for files in a directory hierarchy

SYNOPSIS
       find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

DESCRIPTION
       This manual page documents the GNU version of find.  GNU find searches the directory tree rooted at each given starting-point by eval-
       uating the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the outcome is
       known (the left hand side is false for and operations, true for or), at which point find moves on to the next file name.  If no start-
       ing-point is specified, `.' is assumed.

       If you are using find in an environment where security is important (for example if you are using it to search directories that are
       writable by other users), you should read the `Security Considerations' chapter of the findutils documentation, which is called Find-
       ing Files and comes with findutils.  That document also includes a lot more detail and discussion than this manual page, so you may
       find it a more useful source of information.

OPTIONS
       The -H, -L and -P options control the treatment of symbolic links.  Command-line arguments following these are taken to be names of
       files or directories to be examined, up to the first argument that begins with `-', or the argument `(' or `!'.  That argument and any
       following arguments are taken to be the expression describing what is to be searched for.  If no paths are given, the current direc-
       tory is used.  If no expression is given, the expression -print is used (but you should probably consider using -print0 instead, any-
       way).

       This manual page talks about `options' within the expression list.  These options control the behaviour of find but are specified im-
       mediately after the last path name.  The five `real' options -H, -L, -P, -D and -O must appear before the first path name, if at all.
       A double dash -- can also be used to signal that any remaining arguments are not options (though ensuring that all start points begin
       with either `./' or `/' is generally safer if you use wildcards in the list of start points).
NORM [1]  master  man://find(1) [-]                                   [man]  0   32:0x20  1%  13:1132 {1}
```

```
$ man find   # this will give manual of find command
$ man man   # manual of man command
```

Implementing Shell Scripting on Linux

Need of Shell Scripting

Sometimes, we want to execute a bunch of commands routinely, so we have to type in all commands each time in terminal. As shell can also take commands as input from file we can write commands in a file and can execute them in shell to avoid this repetitive work.

Introduction

- Shell scripts are also known as `Shell Programs` or `Shell Procedures.`
- Shell script file means, a file contains a set of commands within it. If any file contains commands, then the can be used as executable file (after making executable.)
- Shell scripts are similar to the **"batch file"** in windows environment.
- It can be useful to execute the set of commands at a single moment of time, we will get our required outputs and those can also be saved under a file.
- Executing commands seperately will consume more time, using shell scripts we can reduce this time to greater extent.
- Shell scripts can also take arguments, which are known as command line arguments.

Steps to execute shell scripts

- Write a shell script in a file in your editor and save it.
- It isn't necessary to save file with **.sh** extension, because in linux every these files can be identified as scripts with there shebang.

```
#!/usr/bin/env sh

#!/bin/bash
```

- Add an executable permission to the script file

```
$ chmod u+rx file1.sh

$ chmod +x file2.sh

$ chmod 754 file3.sh
```

- Execute the file with in your shell environment

```
$ ./file2.sh

$ dash file.sh

$ sh file3.sh
```

Types of Shell Scripts

1. Static Scripts(Non-Interactive Scripts)
2. Dynamic Scripts(Interactive Shell Scripts)

Static Scripts

It does not require any input from the user once the execution has started

Program-1: Write a static script using the cat command to execute the following commands- `ls, date, cal, who`

```
$ cat > script1.sh
ls
date
cal
who  # press ctrl-d
$
$ chmod u+x script1.sh
```

```
$ ./script1.sh
os_assign.html
os_assign.md
os_assignment.aux
os_assignment.fdb_latexmk
os_assignment.fls
os_assignment.lof
os_assignment.log
os_assignment.lot
os_assignment.md
os_assignment.pdf
os_assignment.tex
os_assignment.toc
os_assign.pdf
oslab_images
Sun 17 May 2020 11:16:10 AM IST
      May 2020
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
raytracer tty7         2020-05-16 09:36 (:0)
$
```

Program-2:: Write a static script using the cat command to execute the following commands- `ls`, `date`, `cal`, `whoami` along with seperator and appropriate messages

```
$ cat > script2.sh
echo "-----------------------------"
ls
echo "-----------------------------"
echo "The date is $(date)"
echo "-----------------------------"
cal
echo "-----------------------------"
printf "%s" "I'm $(whoami)"
echo "-----------------------------"
$
$ chmod 754 script2.sh
$ ./script2.sh
-----------------------------
bullet_style.tex
chap_breaks.tex
listings_setup.tex
os_assign.md
oslab_images
pdf_property.tex
-----------------------------
The date is Sun 17 May 2020 01:33:39 PM IST
-----------------------------
      May 2020
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
-----------------------------
```

```
raytracer
-----------------------------
```

Student Task: Write a static script using cat command to execute the following commands- `whaomi, cal 2020, banner <Your Name>` with appropriate titles

whoami done above

```
$ cat > student1.sh
echo "Calender of this year"
cal
echo "My name is: "
banner Abhay
figlet -f /usr/share/figlet/fonts/3D\ Diagonal.flf 'Abhay'
$
$ chmod +x student1.sh
$ ./student1.sh
Calender of this year
                                2020

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1    1  2  3  4  5  6  7
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    8  9 10 11 12 13 14
12 13 14 15 16 17 18    9 10 11 12 13 14 15   15 16 17 18 19 20 21
19 20 21 22 23 24 25   16 17 18 19 20 21 22   22 23 24 25 26 27 28
26 27 28 29 30 31      23 24 25 26 27 28 29   29 30 31

        April                   May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                   1  2       1  2  3  4  5  6
 5  6  7  8  9 10 11    3  4  5  6  7  8  9    7  8  9 10 11 12 13
12 13 14 15 16 17 18   10 11 12 13 14 15 16   14 15 16 17 18 19 20
19 20 21 22 23 24 25   17 18 19 20 21 22 23   21 22 23 24 25 26 27
26 27 28 29 30         24 25 26 27 28 29 30   28 29 30
                       31
        July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1           1  2  3  4  5
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    6  7  8  9 10 11 12
12 13 14 15 16 17 18    9 10 11 12 13 14 15   13 14 15 16 17 18 19
19 20 21 22 23 24 25   16 17 18 19 20 21 22   20 21 22 23 24 25 26
26 27 28 29 30 31      23 24 25 26 27 28 29   27 28 29 30
                       30 31
       October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7           1  2  3  4  5
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    6  7  8  9 10 11 12
11 12 13 14 15 16 17   15 16 17 18 19 20 21   13 14 15 16 17 18 19
18 19 20 21 22 23 24   22 23 24 25 26 27 28   20 21 22 23 24 25 26
25 26 27 28 29 30 31   29 30                   27 28 29 30 31
My name is:

  **    *       *
 *  *   *       *
*    *  *       *
*    *  * ***   * ***    ****   *    *
*    *  **   *  **   *       *  *    *
******  *    *  *    *   *****  *    *
*    *  *    *  *    *  *    *  *   **
*    *  **   *  *    *  *   **   *** *
*    *  * ***   *    *   *** *       *
```

```
                                  *    *
                                   ****


   ,---,                    ,---,
  '   .' \         ,---,    ,--.' |
 /  ;    '.     ,---.'|    |  |  :
:  :       \    |   | :    :  :  :
:  |   /\   \   :   : :    :  |  |,--.   ,--.--.         .--,
|  :  ' ;.   : :      |,-.|  :  '   | /       \      /_ ./|
|  |  ;/  \   \|    : '  ||  |   /' :.--.   .-. | , ' , ' :
'  :  | \  \ ,'|    |  / :'  :  | | | \__\/: . ./___/ \: |
|  |  '  '--'  '    : |: ||  |  ' | : ," .--.; | .  \  ' |
|  :  :         |    | '/ :|  :  :_:,'/  /  ,.  |  \  ;   :
|  | ,'         |    :    ||  | ,'    ;  :   .'   \  \  \  ;
`--''           /    \  / `--''      |  ,     .-./   :  \  \
                `-'----'              `--`---'         \  ' ;
                                                        `--`
```

Dynamic Script

It requires input from the user once the execution has started

Program-3: Write a dynamic script to find list of the files or directories from a given directory

```
cat > give_list.sh
# taking path
echo "list of the files/directories are:"
echo
ls "$1"
$
$ chmod +x give_list.sh
$ ./give_list.sh "$HOME"
list of the files/directories are:

canyonbottom
Documents
Downloads
go
Music
Pictures
R
README.md
Templates
Videos
vimwiki
yay
$
```

Student Task Write a dynamic script to take month number and year number and display it's cal

```
$ cat > show_cal.sh
echo "Enter month and year"
read month
read year
cal "${month}" "${year}"
$
$ chmod u+x show_cal.sh
$ ./show_cal.sh
Enter month and year
12
2004
    December 2004
```

```
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
$
```

Program-4: Write a dynamic script to arithmetic calculations of given values

```
$ cat > calculation.sh
echo "Enter two numbers"
read -r fnum
read -r snum
echo "You entered:"
printf "%s\n%s\n" "first num: ${fnum}" "second num: ${snum}"
printf "Addition: %s\n" $(( fnum + snum ))
printf "Substraction: %s\n" $(( fnum - snum ))
printf "Multiplication: %s\n" $(( fnum * snum ))
printf "Division: %s\n" $(( fnum / snum ))
printf "Remainder: %s\n" $(( fnum % snum ))
$
$ chmod 740 calculation.sh
$ ./calculation.sh
Enter two numbers
5
4
You entered:
first num: 5
second num: 4
Addition: 9
Substraction: 1
Multiplication: 20
Division: 1
Remainder: 1
```

Program-5: Write a dynamic script to compare whether the two strings are equal or not

```
$ cat compare.sh
echo "Enter 2 names"
read fname
read sname
echo "You entered:"
printf "%s\n%s\n" "first name: ${fname}" "second name: ${sname}"
echo "String comparision result is:`expr $fname = $sname`"
$
$ chmod u+x compare.sh
$ ./compare.sh
Enter 2 names
Ram
Shyam
You entered:
first name: Ram
second name: Shyam
String comparision result is:0
```

Here's an example of shell scripting, this script opens man page in vim and also lets you fuzzy find all *apropos* commands

```
maninvim
 14
 13 getmanname() {
-12     man -k . | \
|11     fzf | \
|10     awk '{print $1}'
 9 }
 8
 7 getshufman() {
-6     getname=$(fd . --type f /usr/share/man/man1 \
+ 5 +--  2 lines folded ----------------------------------------------------------------------------------------
|4     propername=$(basename "${getname}" \
-3         | sed 's/.1.gz//g')
|2     echo "${propername}"
 1 }
27
 1 viman() {
-!2     text=$(man "$@") && \
|3     echo "$text" | \
|4     "$EDITOR" -R +":set ft=man" - ;
 5 }
 6
 7 case "$1" in
-8     -h) printf "%s\n%s\n%s\n%s\n" "There are three queries to pass:" \
+ 9 +--  2 lines folded ----------------------------------------------------------------------------------------
|10     -s) viman "$(getmanname)";;
|11     -r) viman "$(getshufman)";;
|12     *) viman "$1"
 13 esac
 14
 15 # arrays are not part of posix so changed shebang to bash
NORM [1] master [~1] maninvim                                   [sh]  984B  0:0x0  62% ℓ 27:43 {0}
"maninvim" 43L, 984C written
```

Implementing Cron Jobs in Linux

Introduction to Cron Jobs

- `Cron` is one of the most useful utility on Linux Operating System. It is used to schedule commands at specific time. These scheduled commands or tasks are known as `Cron Jobs`.
  - There are some other programs also which are used to schedule taks in linux, they are:
    - ⋆ **Anacron**: Scheduler works even when your system is off
    - ⋆ **Fcron**: Best of both *cron* and *Anacron*
    - ⋆ **Hcron**: Lesser known, easy label of jobs, back-up etc.
    - ⋆ **Jobber**: Written in `go`, it features job execution history with status
    - ⋆ **entr**: Not same as above, rather it watches changes happened in content of files
- Cron is generally used for running scheduled backups, monitoring disk space, deleting files (periodically which is no longer required), running system maintainance tasks, etc.
- Mostly `cron` comes installed in most distros but if not you can do in following ways:
  - For `arch` based distros, it is available in `pacman`:

```
$ sudo pacman -S cronie

# after installation enable and start cron service(daemon)
$ sudo systemctl enable cronie.service
$ sudo systemctl start cronie.service

# to check if it's activated check status
$ sudo systemctl status cronie.service
 cronie.service - Periodic Command Scheduler
    Loaded: loaded (/usr/lib/systemd/system/cronie.service; enabled; vendor preset: disabled)
    Active: active (running) since Sun 2020-05-17 16:48:21 IST; 7s ago
  Main PID: 241166 (crond)
     Tasks: 1 (limit: 4051)
    Memory: 572.0K
    CGroup: /system.slice/cronie.service
             241166 /usr/bin/crond -n

May 17 16:48:21 server systemd[1]: Started Periodic Command Scheduler.
May 17 16:48:21 server crond[241166]: (CRON) STARTUP (1.5.5)
May 17 16:48:21 server crond[241166]: (CRON) INFO (Syslog will be used instead of sendmail.)
May 17 16:48:21 server crond[241166]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 26%
May 17 16:48:21 server crond[241166]: (CRON) INFO (running with inotify support)
May 17 16:48:21 server crond[241166]: (CRON) INFO (@reboot jobs will be run at computer's star
```

    systemctl works only systemd based distros.
- There is a cron "daemon" that runs on linux system.
  - **Daemon**: A daemon is a program that runs in the background all the time, usually initiated by the system)
- This cron daemon is responsible for launching the cron jobs on schedule
- Cron does contains its own environment variables, which are sometimes different than of your shell.

Syntax

There are two main parts: 1. The first part is Timing. 2. The second part is command that would run from command line.

Timing Syntax

This is the first part of the cron job string. It determines how often and when the cron jobs is going to run

It consists of 5 parts:

1. minute
2. hour
3. day of month
4. month
5. day of week

```
* * * * *
| | | | |
| | | | ---> Day of Week(0-6) where 0 represents Sunday
| | | -----> Month (1-12)
| | -------> Day of Month (1-31)
| ---------> Minute (0-23)
-----------> Minute (0-59)
```

An `asterisk(*)` represents all possible numbers for that position. For example, asterisk in the minute position would make it run every minute.

Examples:

Managing Cron Jobs

- This cron job will run every minute, all the time

`* * * * * [command]`

- This cron job will run at minute zero, every hour (i.e., an hourly cron job)

`0 * * * * [command]`

- This is an hourly cron job but run at minute 15 instead

`15 * * * * [command]`

- This will run once a day, at 2:30am

`30 2 * * * [command]`

- Division operator is also used. This will run 12 times per hour, i.e., every 5 minutes

`*/5 * * * * [command]`

- There are few special keyword that will let you run a cron job

Syntax	Work
@reboot[command]	Run once, at start-up
@yearly[command]	Run once a year
@yearly[command]	Run once a year
@annually[command]	Same as year
@monthly[command]	Run once a month
@weekly[command]	Run once a weekly
@daily[command]	Run once a daily
@midnight[command]	Same as daily

1. **crontab -e**: This command is used to edit the contents of the crontab file
2. **crontab -l**: This command is used to see existing cron jobs
3. **crontab -r**: This command is used to delete the existing cron jobs

Here, are the screenshots of the cron jobs which I'm using currently:

```
*/30 * * * * /usr/bin/updatedb
# */2 */4 * * * eval "export DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus"; /home/raytracer/.local/bin/.scripts/cron_scripts/pacman_chec
kup.sh
# */2 */4 * * * /home/raytracer/.local/bin/.scripts/cron_scripts/pacman_checkup.sh
0 */4 * * SUN /usr/bin/mandb
~
```

```
crontab.FZo401
# Minute hour DOM mon DOW cmd
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
8 7 * * * eval "export DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus"; /home/raytracer/.local/bin/.scripts/cron_scripts/music_update.sh
0 */8 * * */2 eval "export DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus"; /home/raytracer/.local/bin/.scripts/cron_scripts/
package_updates.sh
*/30 * * * * newsboat -x reload
```

Thank You