OS-LAB-PROJECT MINIX

Roll	Name
0802	Iftekhar Jamil
0811	Tulsi Chandra Das
0817	Atiq Ahammed
0825	Aba Kowsar Tushar
0832	Afia Sajeeda
0842	Md. Jewel Rana

ABSTRACT

This document contains a short description about Minix operating system and a little information for installing and using it by answering a few questions.

Group – 2 (Ice-breakers)

401 – Operating system

Submitted to

Saeed Siddik

Introduction to MINIX 3

How often have you rebooted your TV set in the past year? Probably a lot less than you have rebooted your computer. Of course there are many "reasons" for this, but increasingly, nontechnical users don't want to hear them. They just want their computer to work perfectly all the time and never crash. MINIX 3 is a project to develop an operating system as reliable as a TV set, for embedded systems and mission critical applications, but also for future $50 single-chip laptops and general desktop use. The focus is being small, simple, and reliable. **Note:** This is the last entry for the *Alternative OS Contest.*

History

MINIX 3 has a bright future but somewhat checkered past. The first version, MINIX 1, was released in 1987 and was the first UNIX clone with all the source code available. It developed rapidly and soon had its own USENET newsgroup (comp.os.minix), with 40,000 subscribers within 3 months, a large number at a time when the Internet was only available to university researchers and students. One of the early MINIX adopters was a Finnish student named Linus Torvalds, who went out and bought a PC just to run MINIX, studied it very carefully, and then decided to write his own operating system, inspired by MINIX. Although Linus knew MINIX very well, he didn't steal any code from it, as some people have alleged. Linus system grew into the modern Linux system. MINIX' author, Andrew Tanenbaum and Torvalds have had some fairly public discussions on operating system design, originally in 1992 and most recently in 2006.

Rebirth

Although MINIX was (and still is) widely used for teaching operating systems courses at universities, it got a new impetus in 2005 when Tanenbaum assembled a new team of people to completely redo it as a highly reliable system. MINIX 3 has some history with MINIX 1 and MINIX 2 (released in 1997 as a POSIX-conformant OS), but it is really a new system (analogous to the relationship between Windows XP and Windows 3.1).

Various studies have shown that software broadly contains something like 6-16 bugs per 1000 lines of code and that device drivers have 3-7 times as many bugs as the rest of the operating system. When combined with the fact that 70% of a typical operating system consists of device drivers, it is clear that device drivers are a big source of trouble. For Windows XP, 85% of the crashes are do to bugs in device drivers. Obviously, to make OSes reliable, something has to be done to deal with buggy device drivers. Building a reliable system despite the inevitable bugs in device drivers was the original driving force behind MINIX 3.

Design

The approach that MINIX 3 uses to achieve high reliability is fault isolation. In particular, unlike traditional OSes, where all the code is linked into a single huge binary running in kernel mode, in MINIX 3, only a tiny bit of code runs in kernel mode--about 4000 lines in all. This code handles interrupts, process scheduling, and interprocess communication. The rest of the operating system runs as a collection of user-mode processes, each one encapsulated by the MMU hardware and none of them running as superuser. One of these processes, dubbed the *reincarnation server*, keeps tabs on all the others and when one of them begins acting sick or crashes, it automatically replaces it by a fresh version. Since many bugs are transient, triggered by unusual timing, in most cases, restarting the faulty component solves the problem and allows the system to repair itself without a reboot and without the user even noticing it. This property is called self healing, and traditional systems do not have it.

The structure of MINIX 3 is shown in Fig. 1. It is constructed as a series of layers. At the bottom, running in kernel mode, is a microkernel, consisting of about 3000 lines of C and 800 lines of assembler. Above that comes a layer of device drivers, with each driver in a separate user-mode process to ease in replacing it should it fail. Then come the servers, which form the core of the operating system. These include the reincarnation server mentioned above, the file server, the process manager, and others, including the X server, the data store, and various others. Finally, on top of that come the user processes. Although internally, MINIX 3 is completely different from other UNIX systems, it supports the standard POSIX interface to applications, so normal UNIX software can be ported fairly easily.

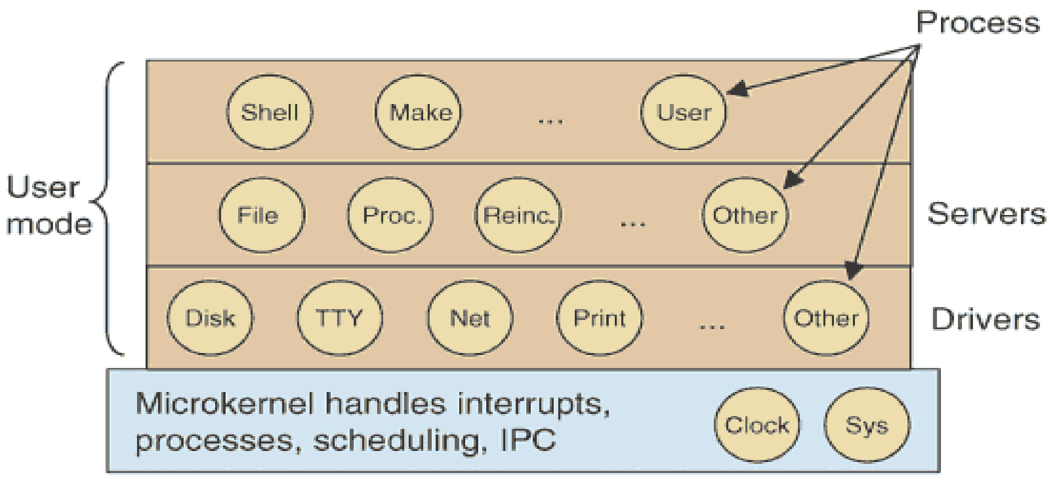


Fig. 1. The MINIX 3 architecture.

The components communicate by passing fixed-length messages. For example, a user process requests file I/O send sending a message to the file server, which then checks its cache and if the needed block is not present, sends a message to the disk driver process to go get the block. While sending a message adds a little bit of overhead (about 500 nsec on a 3-GHz Pentium 4), the system is still quite responsive. For example, a complete system build, which requires over 120 compilations, takes well under 10 sec.

User view

From the user's point of view, MINIX 3 looks like UNIX, except less bloated. It comes with the X Window System and over 400 standard UNIX programs, including:

Shells: ash, bash, pdksh, rsh
Editors: emacs, nvi, vim, elvis, elle, mined, sed, ed, ex
Language tools: cc, gcc, g++, bison, flex, perl, python, yacc
Programming tools: cdiff, make, patch, tar, touch
Networking: ssh, telnet, ftp, lynx, mail, rlogin, wget, pine
File utilities: cat, cp, bzip2, compress, mv, dd, uue, GNU utilities
Text utilities: grep, head, paste, prep, sort, spell, tail
Administration: adduser, cron, fdisk, mknod, mount, cvs, rcs
Games: dungeon, nethack

Currently the user interface is just X, but someday a GUI may be added if a suitable lightweight GUI can be found. Here are some screen shots.

Availability

MINIX 3 is open source software, under the BSD license. It has its own Website from which the a bootable CD-ROM image containing all the sources and binaries can be downloaded. To install it, just boot the CD-ROM, login as root, and type: *setup*. Installation takes about 10 minutes. After installation, a large number of packages can be installed from the CD-ROM or the Website by just typing: *packman* to select the choices. Currently MINIX 3 runs on x86 hardware, but ports to the PowerPC and Xscale are underway. It also runs fine on virtual machines such as VMware and Xen.

Since MINIX 3 went public in late 2005, the Website has had over 300,000 unique visitors and the CD-ROM image has been downloaded some 75,000 times. Currently, the site is getting over 1000 visitors a day. There is an active Google USENET newsgroup, comp.os.minix, where people ask and answer questions, post new software, and discuss MINIX 3. MINIX 3 is a community effort and your help is most welcome. Go get the system, try it out, and join the future.

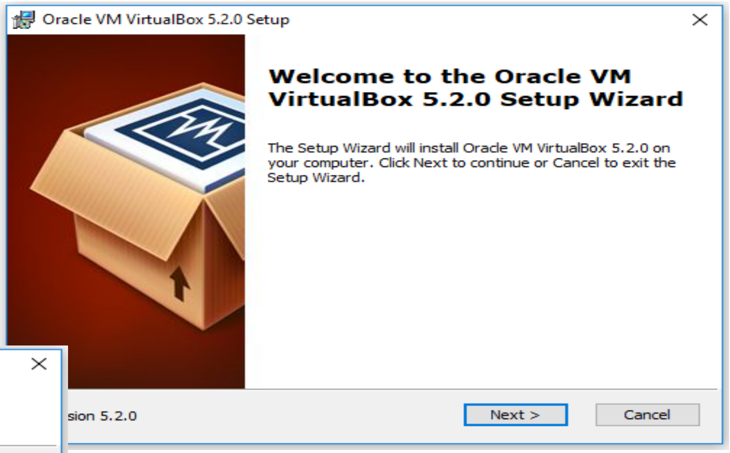
1.1 Installation

Requirements

- You need a windows / linux – 64 / 32 bits operating system. Other operating systems are also workable but we used windows 64 bits operating system.
- The second thing that you need is an environment for virtualizing minix os. To fulfill this you need a virtual machine such as virtualbox or VMware. For this project we use both of them. Note: For virtualbox, in some machines there occur a GURU Meditation problem. If the machine's boot' advance setting's virtualization is being disabled then this particular problem occur. But for the VMware this problem is quite rare. For downloading you can visit the following urls: https://www.virtualbox.org/wiki/Downloads https://my.vmware.com/en/web/vmware/info/slug/desktop_end_user_computing/vmware_workstation_pro/14_0#open_source.
- Operating system (bootable iso file). For downloading you can visit http://www.minix3.org/

Installing the VirtualBox

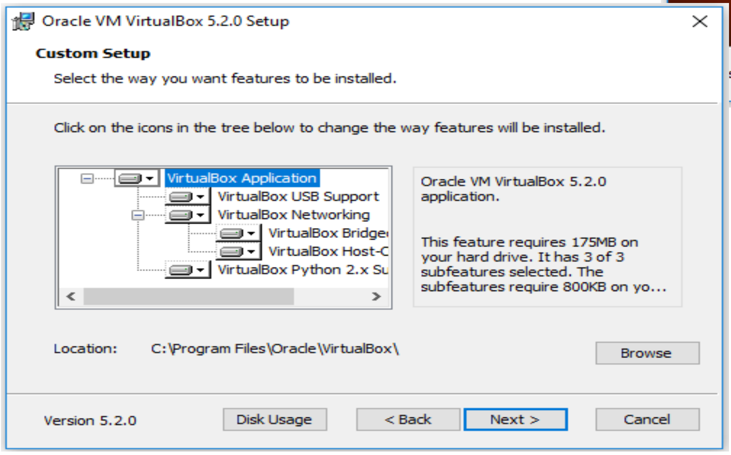
- First download the .exe file from the given url.
- Then install the run the downloaded file to install the virtual machine. The following figure for visualizing the process.



Oracle VM VirtualBox 5.2.0 Setup

Welcome to the Oracle VM VirtualBox 5.2.0 Setup Wizard

The Setup Wizard will install Oracle VM VirtualBox 5.2.0 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.



Oracle VM VirtualBox 5.2.0 Setup

Custom Setup

Select the way you want features to be installed.

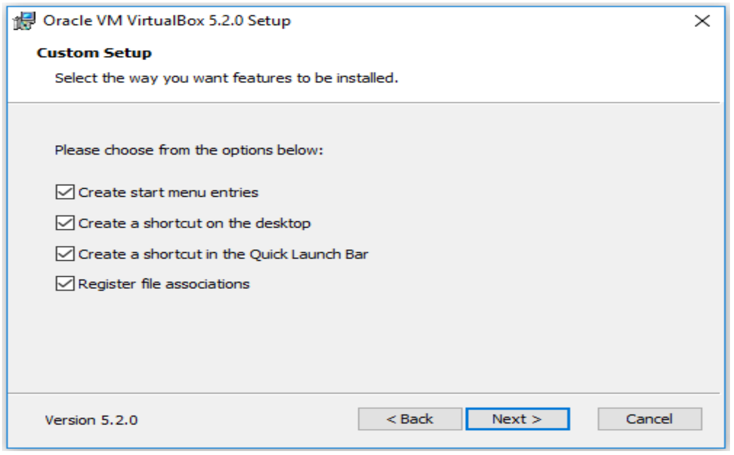
Click on the icons in the tree below to change the way features will be installed.

- VirtualBox Application
  - VirtualBox USB Support
  - VirtualBox Networking
    - VirtualBox Bridge
    - VirtualBox Host-C
  - VirtualBox Python 2.x Su

Oracle VM VirtualBox 5.2.0 application.

This feature requires 175MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 800KB on yo...

Location: C:\Program Files\Oracle\VirtualBox\



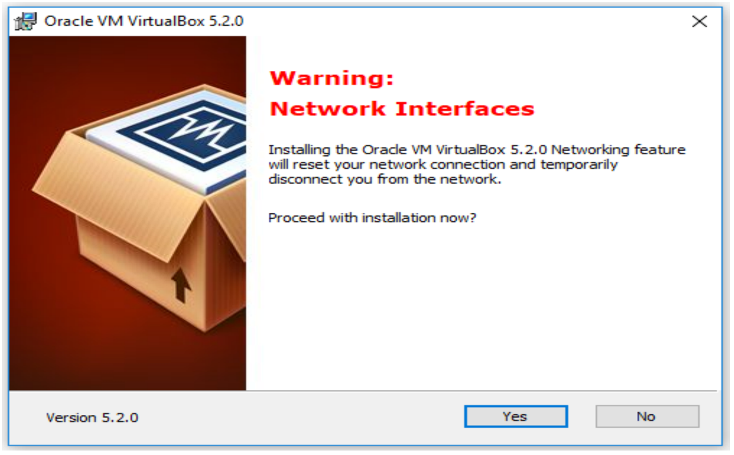
Custom Setup

Select the way you want features to be installed.

Please choose from the options below:

- ☑ Create start menu entries
- ☑ Create a shortcut on the desktop
- ☑ Create a shortcut in the Quick Launch Bar
- ☑ Register file associations

Version 5.2.0

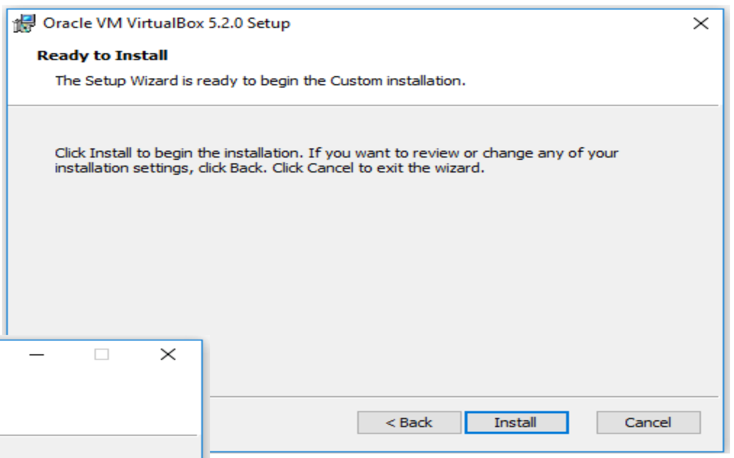


Warning:
Network Interfaces

Installing the Oracle VM VirtualBox 5.2.0 Networking feature will reset your network connection and temporarily disconnect you from the network.

Proceed with installation now?

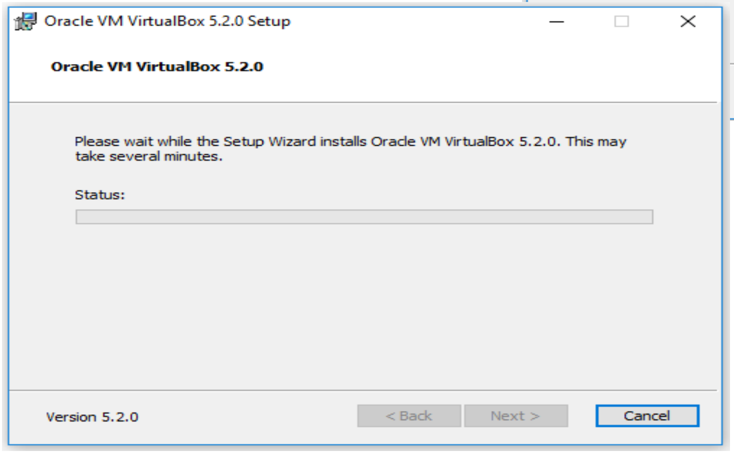
Version 5.2.0



Ready to Install

The Setup Wizard is ready to begin the Custom installation.

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.



Oracle VM VirtualBox 5.2.0

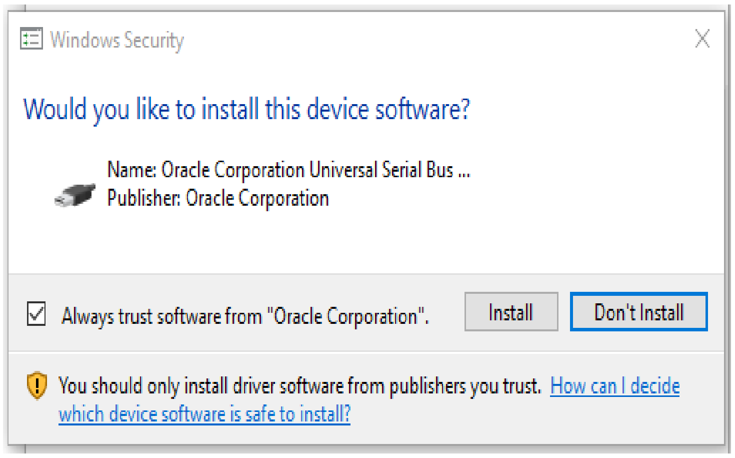
Please wait while the Setup Wizard installs Oracle VM VirtualBox 5.2.0. This may take several minutes.

Status:

Version 5.2.0

Wait a while in this step and don't cancel

Please install this device software.

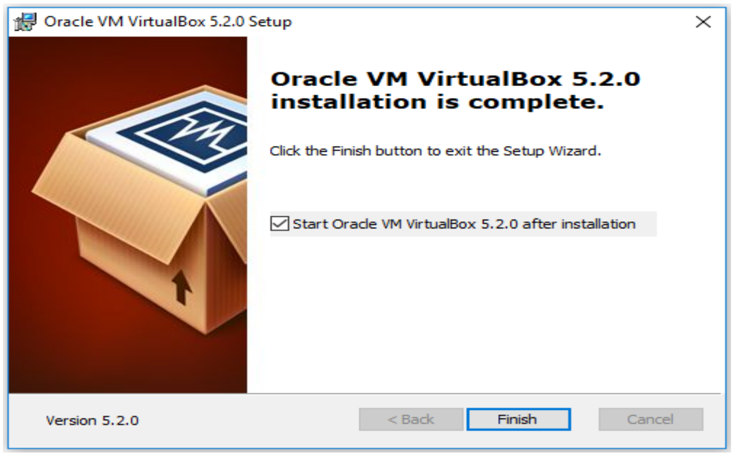

Windows Security

Would you like to install this device software?

Name: Oracle Corporation Universal Serial Bus ...
Publisher: Oracle Corporation

☑ Always trust software from "Oracle Corporation".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?


Oracle VM VirtualBox 5.2.0 Setup

Oracle VM VirtualBox 5.2.0 installation is complete.

Click the Finish button to exit the Setup Wizard.

☑ Start Oracle VM VirtualBox 5.2.0 after installation

Version 5.2.0

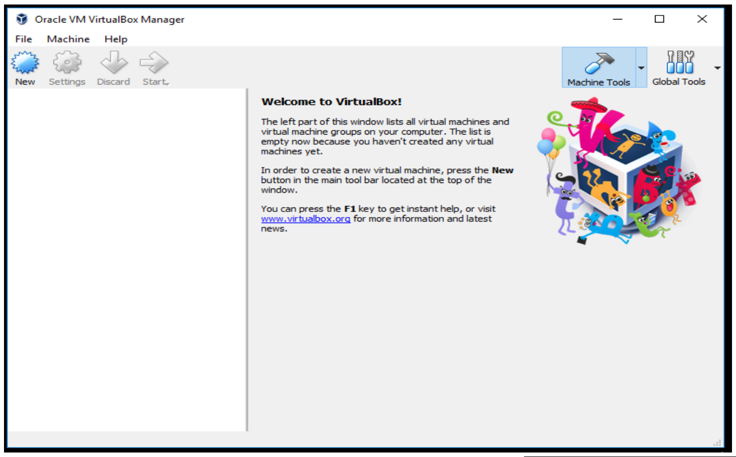
Over here you can ensure yourself that your virtual machine is successfully installed.

Installing MINIX os

In this step you need to install the operating system ios file to the virtual machine. The following figures and instractions will be helpful for you.

First download the minix iso file frm the given url. I used the minix 3.2.1 as in this version cc and clang package is pre-installed that is helpful for compiling c programs.

Open the virtual machine and follow the following steps.


Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Machine Tools Global Tools

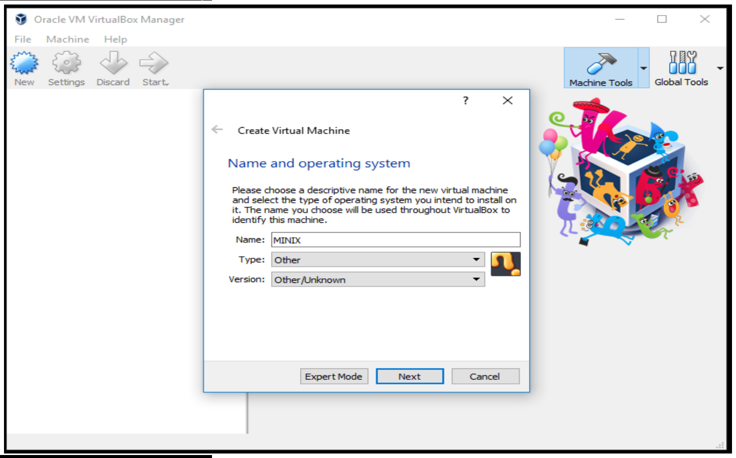
Welcome to VirtualBox!

The left part of this window lists all virtual machines and virtual machine groups on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Select the new option from the menu. Then the following window will be displayed.

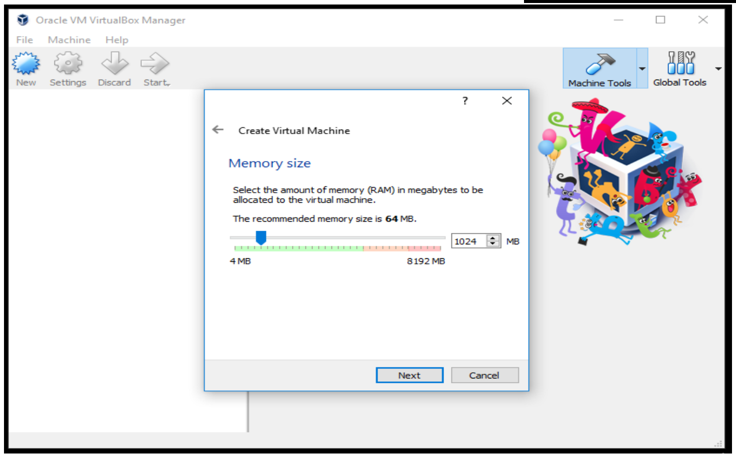

Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Machine Tools Global Tools
Create Virtual Machine
Expert Mode
Next
Cancel

Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: MINIX
Type: Other
Version: Other/Unknown

Give a name for virtualizing your operating system. I named it MINIX. You can choose your own. Then select the type "Other" and Version "Other/Unknown". Then press next.


Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Machine Tools Global Tools
Create Virtual Machine
4 MB
8192 MB
Next
Cancel

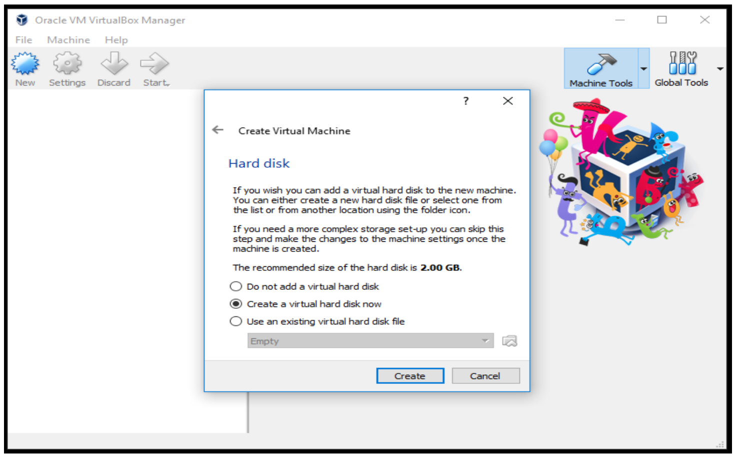
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **64** MB.

1024 MB

Take a memory size more than or equal 512 mb. For my installing I always prefer for 1024 mb.


Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Machine Tools Global Tools
?
Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

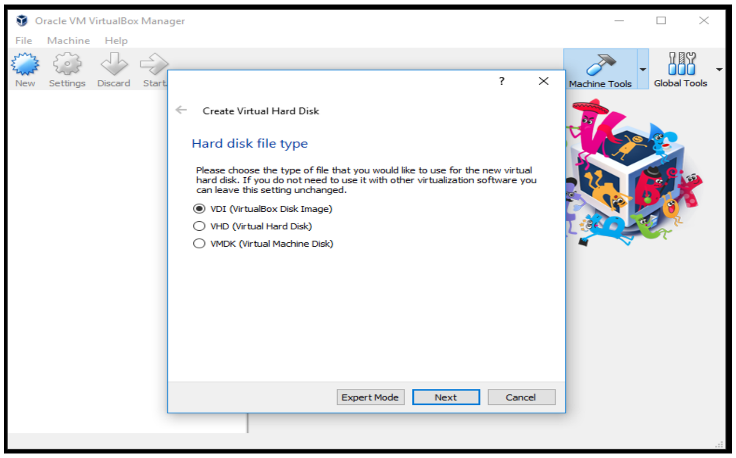
The recommended size of the hard disk is **2.00 GB**.

- ○ Do not add a virtual hard disk
- ● Create a virtual hard disk now
- ○ Use an existing virtual hard disk file

Empty

Choose the default "Hard disk" and then create.

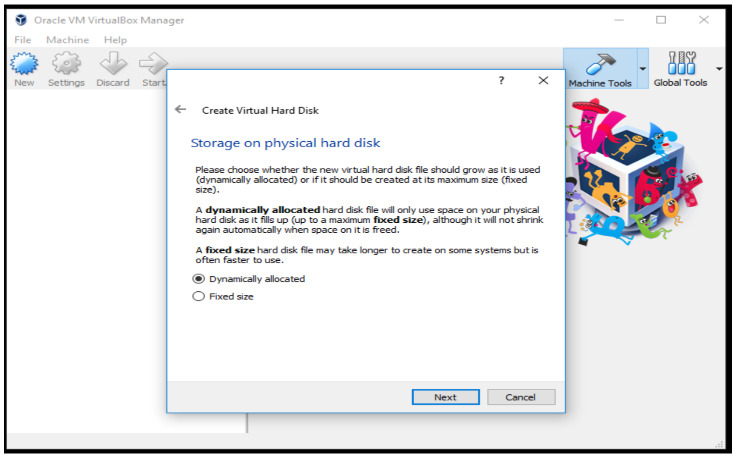
Choose the default "Hard disk file time" and then go to next.


Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Machine Tools Global Tools
?
Create Virtual Hard Disk

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

- ● VDI (VirtualBox Disk Image)
- ○ VHD (Virtual Hard Disk)
- ○ VMDK (Virtual Machine Disk)


Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Machine Tools Global Tools
?
Create Virtual Hard Disk

Storage on physical hard disk

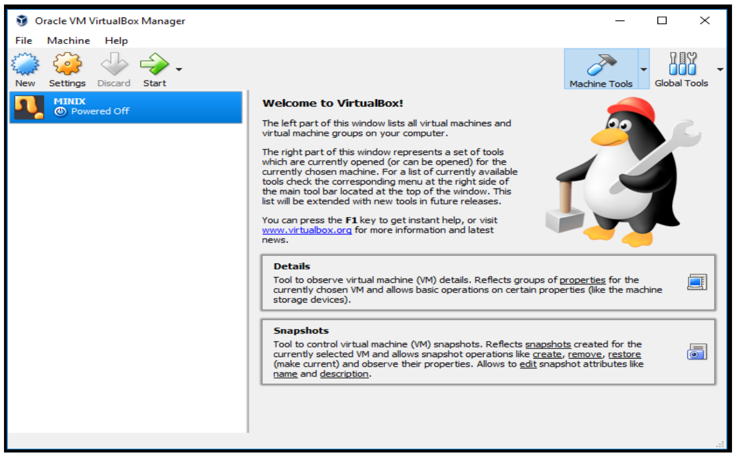
Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

- ● Dynamically allocated
- ○ Fixed size

System on physical hard disk will be dynamically allocation. So choose the "Dynamically allocated" option and then go to next.


Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Machine Tools Global Tools
MINIX
Powered Off

Welcome to VirtualBox!

The left part of this window lists all virtual machines and virtual machine groups on your computer.

The right part of this window represents a set of tools which are currently opened (or can be opened) for the currently chosen machine. For a list of currently available tools check the corresponding menu at the right side of the main tool bar located at the top of the window. This list will be extended with new tools in future releases.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Details

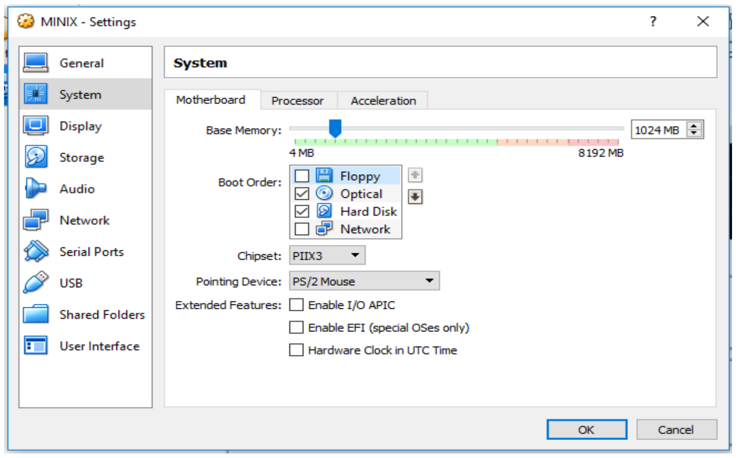
Tool to observe virtual machine (VM) details. Reflects groups of properties for the currently chosen VM and allows basic operations on certain properties (like the machine storage devices).

Snapshots

Tool to control virtual machine (VM) snapshots. Reflects snapshots created for the currently selected VM and allows snapshot operations like create, remove, restore (make current) and observe their properties. Allows to edit snapshot attributes like name and description.

Then choose the setting option to upload the iso file.

In the system setting make the boot order not floppy. To do this unselect the Floppy option.



System

Motherboard | Processor | Acceleration

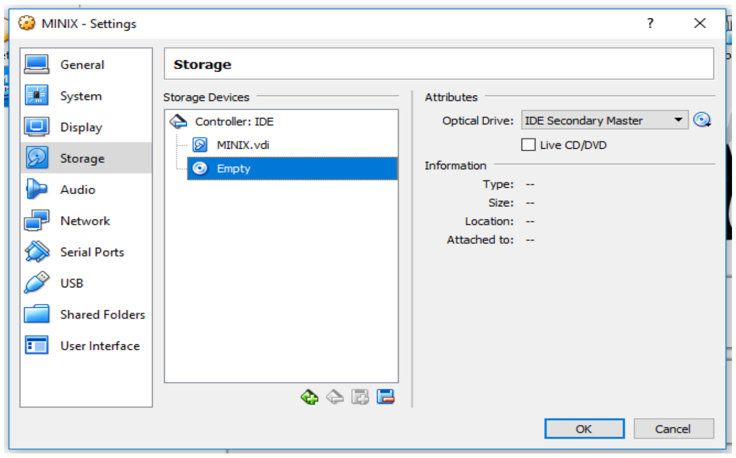
Base Memory: 1024 MB (4 MB – 8192 MB)

Boot Order:
- [ ] Floppy
- [✓] Optical
- [✓] Hard Disk
- [ ] Network

Chipset: PIIX3

Pointing Device: PS/2 Mouse

Extended Features:
- [ ] Enable I/O APIC
- [ ] Enable EFI (special OSes only)
- [ ] Hardware Clock in UTC Time



Storage

Storage Devices
- Controller: IDE
  - MINIX.vdi
  - Empty

Attributes

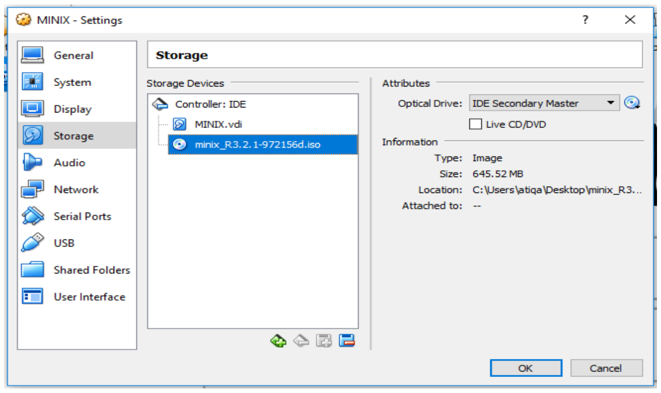
Optical Drive: IDE Secondary Master

- [ ] Live CD/DVD

Information

Type: --
Size: --
Location: --
Attached to: --

In the storage setting choose the Empty space to upload the ios file.



Storage

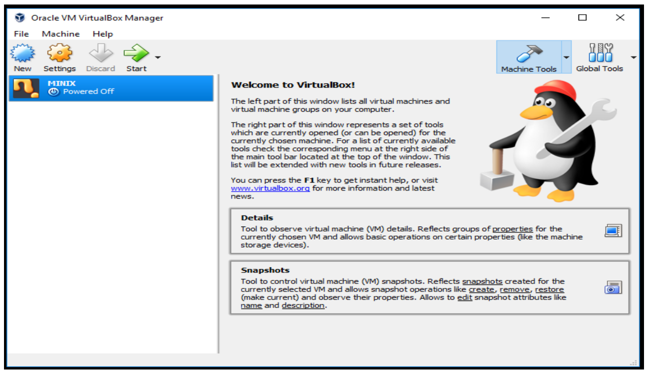
Storage Devices
- Controller: IDE
  - MINIX.vdi
  - minix_R3.2.1-972156d.iso

Attributes
- Optical Drive: IDE Secondary Master
- Live CD/DVD

Information
- Type: Image
- Size: 645.52 MB
- Location: C:\Users\atiqa\Desktop\minix_R3...
- Attached to: --

After choosing the iso file press ok to save change of the setting.

Now run the virtual machine selecting your required operating system. Log in the system by root. Then setup the system. While setup processing is running please accept all the required condition. After setup completed please power off the system using "poweroff" command.



Welcome to VirtualBox!

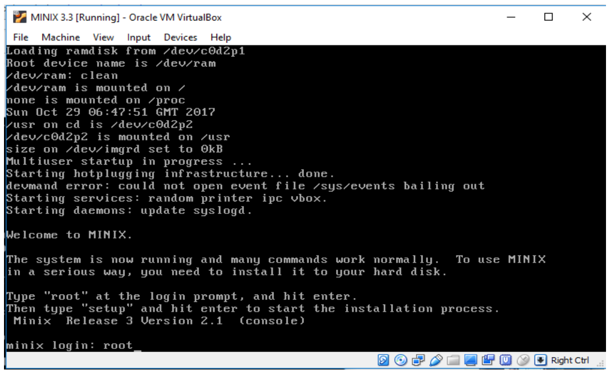
The left part of this window lists all virtual machines and virtual machine groups on your computer.

The right part of this window represents a set of tools which are currently opened (or can be opened) for the currently chosen machine. For a list of currently available tools check the corresponding menu at the right side of the main tool bar located at the top of the window. This list will be extended with new tools in future releases.

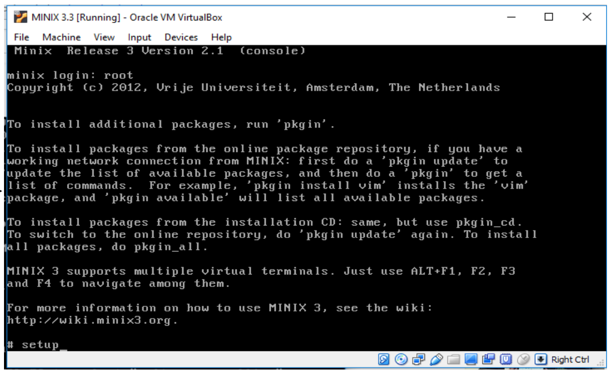
You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Details
Tool to observe virtual machine (VM) details. Reflects groups of properties for the currently chosen VM and allows basic operations on certain properties (like the machine storage devices).

Snapshots
Tool to control virtual machine (VM) snapshots. Reflects snapshots created for the currently selected VM and allows snapshot operations like create, remove, restore (make current) and observe their properties. Allows to edit snapshot attributes like name and description.

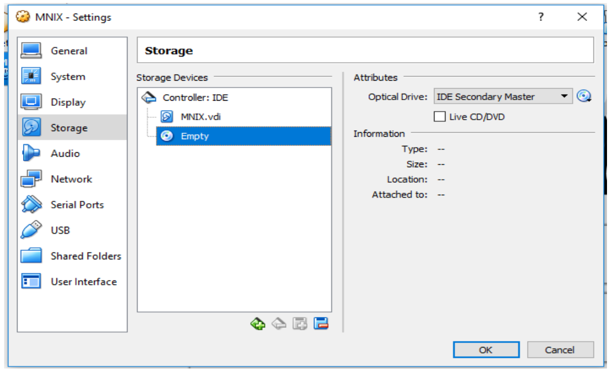


```
Loading ramdisk from /dev/c0d2p1
Root device name is /dev/ram
/dev/ram: clean
/dev/ram is mounted on /
none is mounted on /proc
Sun Oct 29 06:47:51 GMT 2017
/usr on cd is /dev/c0d2p2
/dev/c0d2p2 is mounted on /usr
size on /dev/imgrd set to 0kB
Multiuser startup in progress ...
Starting hotplugging infrastructure... done.
devmand error: could not open event file /sys/events bailing out
Starting services: random printer ipc vbox.
Starting daemons: update syslogd.

Welcome to MINIX.

The system is now running and many commands work normally.  To use MINIX
in a serious way, you need to install it to your hard disk.

Type "root" at the login prompt, and hit enter.
Then type "setup" and hit enter to start the installation process.
 Minix  Release 3 Version 2.1  (console)

minix login: root_
```



```
 Minix  Release 3 Version 2.1  (console)

minix login: root
Copyright (c) 2012, Vrije Universiteit, Amsterdam, The Netherlands


To install additional packages, run 'pkgin'.

To install packages from the online package repository, if you have a
working network connection from MINIX: first do a 'pkgin update' to
update the list of available packages, and then do a 'pkgin' to get a
list of commands.  For example, 'pkgin install vim' installs the 'vim'
package, and 'pkgin available' will list all available packages.

To install packages from the installation CD: same, but use pkgin_cd.
To switch to the online repository, do 'pkgin update' again. To install
all packages, do pkgin_all.

MINIX 3 supports multiple virtual terminals. Just use ALT+F1, F2, F3
and F4 to navigate among them.

For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org.

# setup_
```

After stopping the system please remove the uploaded file from the virtual machine. If it remains as usual then the system reboot as linux sometime does.



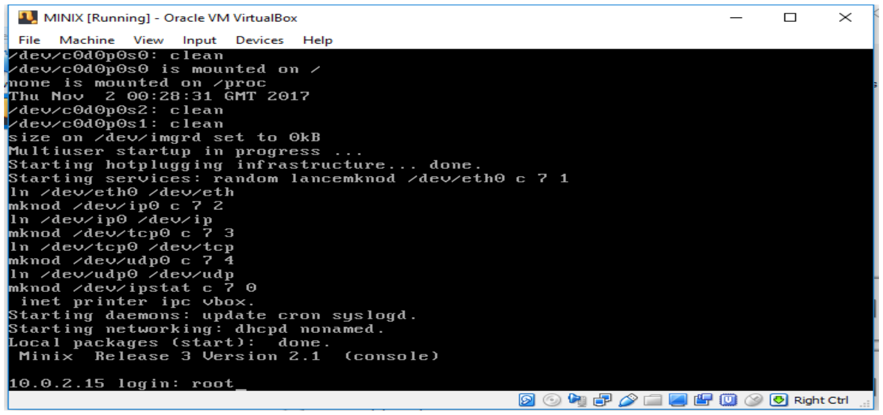
Storage

Storage Devices
- Controller: IDE
  - MINIX.vdi
  - Empty

Attributes
- Optical Drive: IDE Secondary Master
- Live CD/DVD

Information
- Type: --
- Size: --
- Location: --
- Attached to: --

Now your minix operating system is successfully installed to your system. Now simple you can run minix from your virtual machine. To log in you must have log in through the root if you don't have other accounts.


MINIX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

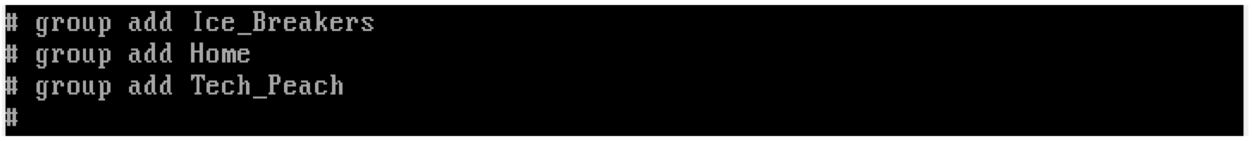
```
/dev/c0d0p0s0: clean
/dev/c0d0p0s0 is mounted on /
none is mounted on /proc
Thu Nov  2 00:28:31 GMT 2017
/dev/c0d0p0s2: clean
/dev/c0d0p0s1: clean
size on /dev/imgrd set to 0kB
Multiuser startup in progress ...
Starting hotplugging infrastructure... done.
Starting services: random lancemknod /dev/eth0 c 7 1
ln /dev/eth0 /dev/eth
mknod /dev/ip0 c 7 2
ln /dev/ip0 /dev/ip
mknod /dev/tcp0 c 7 3
ln /dev/tcp0 /dev/tcp
mknod /dev/udp0 c 7 4
ln /dev/udp0 /dev/udp
mknod /dev/ipstat c 7 0
 inet printer ipc vbox.
Starting daemons: update cron syslogd.
Starting networking: dhcpd nonamed.
Local packages (start):  done.
 Minix  Release 3 Version 2.1  (console)

10.0.2.15 login: root_
```

1.2 Using Command Line For Operation

Creating Group

For adding a group the required command is

$ group add groupName



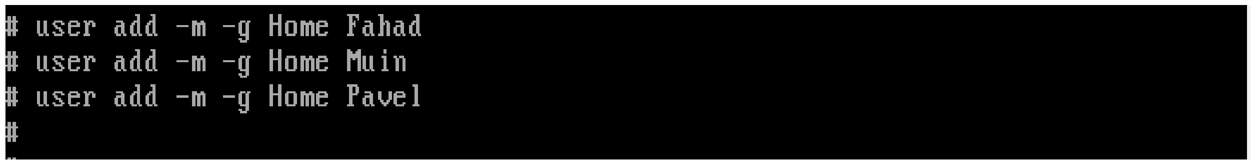
```
# group add Ice_Breakers
# group add Home
# group add Tech_Peach
#
```

I have added 3 groups, Ice_Breakers, Home, Teach_Peach. In such way you can add one or more groups where multiple or single user can be members of any particular group.

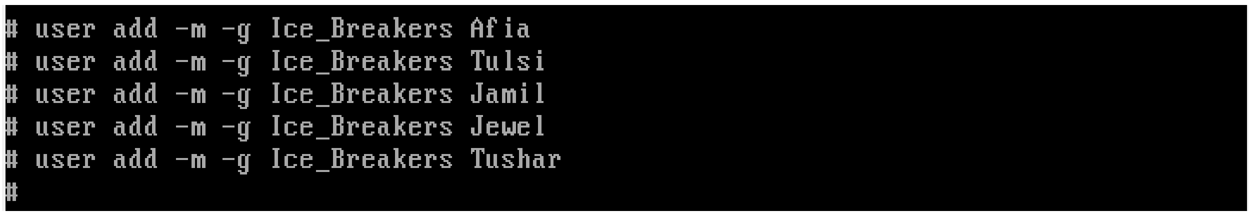
Adding Users In Group

For adding users in a group you can use the following command. There are other programs that can also add users.

$ user add –m –g groupName username



```
# user add -m -g Home Fahad
# user add -m -g Home Muin
# user add -m -g Home Pavel
#
```



```
# user add -m -g Ice_Breakers Afia
# user add -m -g Ice_Breakers Tulsi
# user add -m -g Ice_Breakers Jamil
# user add -m -g Ice_Breakers Jewel
# user add -m -g Ice_Breakers Tushar
#
```

Here I have added Fahad and Pavel to Home group, Afia, Tulshi, Jamil, Jewel and Tushar to IceBreakers group.

Add Files

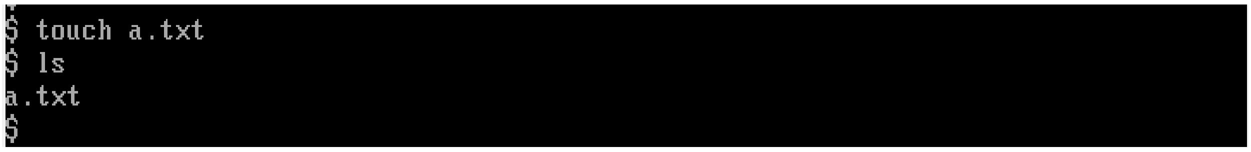
For adding files we can use multiple command lines, such as

$ touch filename

$ nano filename

$ vi filename

I use touch command to create a file in a user's account.



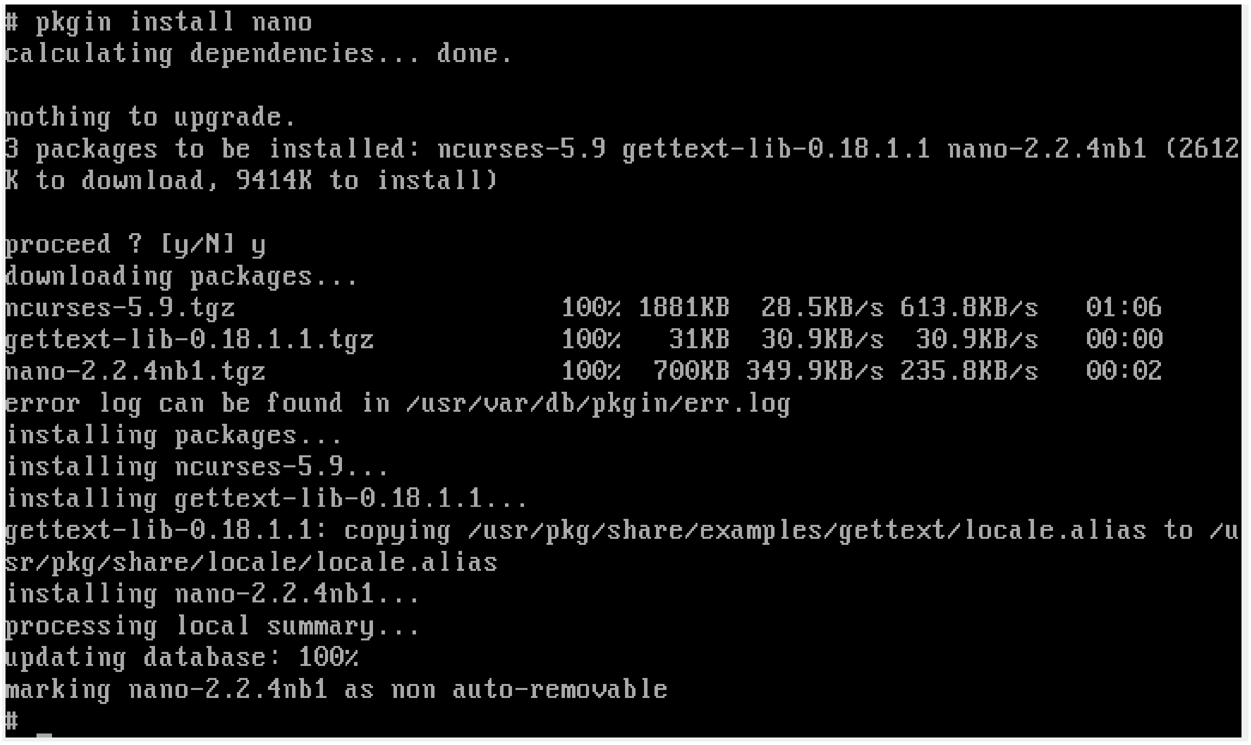
```
$ touch a.txt
$ ls
a.txt
$
```

Here a.txt file was created.

Edit File

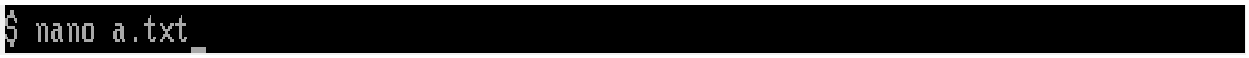
I was using nano editor for managing files. For this I had to install nano editor package. The process was to install nano or any other software is :

$ pkgin install softwareName

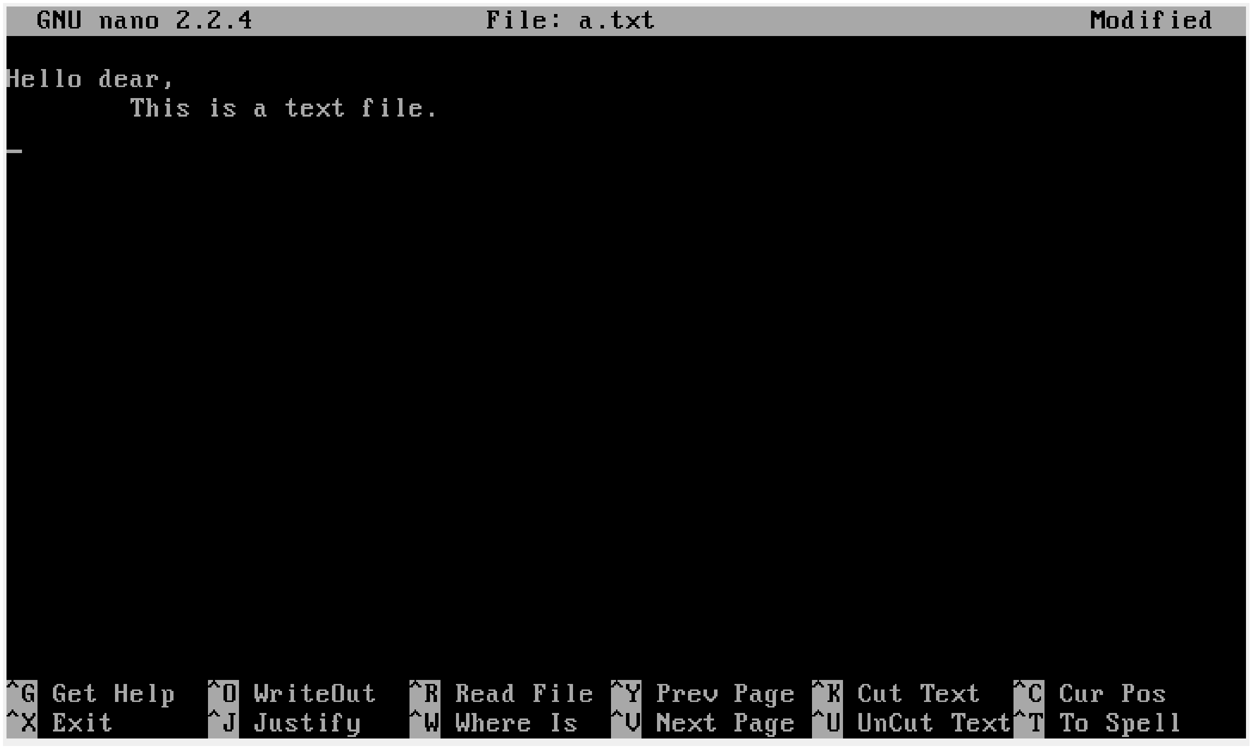


```
# pkgin install nano
calculating dependencies... done.

nothing to upgrade.
3 packages to be installed: ncurses-5.9 gettext-lib-0.18.1.1 nano-2.2.4nb1 (2612
K to download, 9414K to install)

proceed ? [y/N] y
downloading packages...
ncurses-5.9.tgz                    100% 1881KB  28.5KB/s 613.8KB/s   01:06
gettext-lib-0.18.1.1.tgz           100%   31KB  30.9KB/s  30.9KB/s   00:00
nano-2.2.4nb1.tgz                  100%  700KB 349.9KB/s 235.8KB/s   00:02
error log can be found in /usr/var/db/pkgin/err.log
installing packages...
installing ncurses-5.9...
installing gettext-lib-0.18.1.1...
gettext-lib-0.18.1.1: copying /usr/pkg/share/examples/gettext/locale.alias to /u
sr/pkg/share/locale/locale.alias
installing nano-2.2.4nb1...
processing local summary...
updating database: 100%
marking nano-2.2.4nb1 as non auto-removable
# _
```

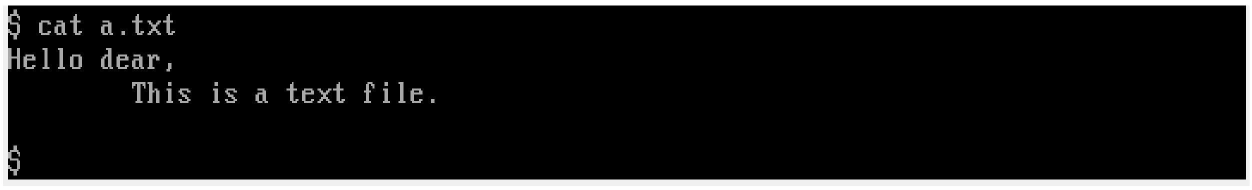
For editing file you need to command the system as "*nano fileName*".



```
$ nano a.txt
```



```
  GNU nano 2.2.4              File: a.txt                        Modified

Hello dear,
        This is a text file.


^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```



```
$ cat a.txt
Hello dear,
        This is a text file.

$
```

Change Permission Of Files for Users

There is a set of rules for each file which defines who can access that file, and how they can access it. These rules are called file permissions or file *modes*. The command name **chmod** stands for "change mode", and it is used to define the way a file can be accessed.

Before continuing, you should read the section What Are File Permissions, And How Do They Work? in our documentation of the **umask** command. It contains a comprehensive description of how to define and express file permissions.

In general, **chmod** commands take the form:

$ chmod options permissions file name

If no options are specified, chmod modifies the permissions of the file specified by filename to the permissions specified by permissions.

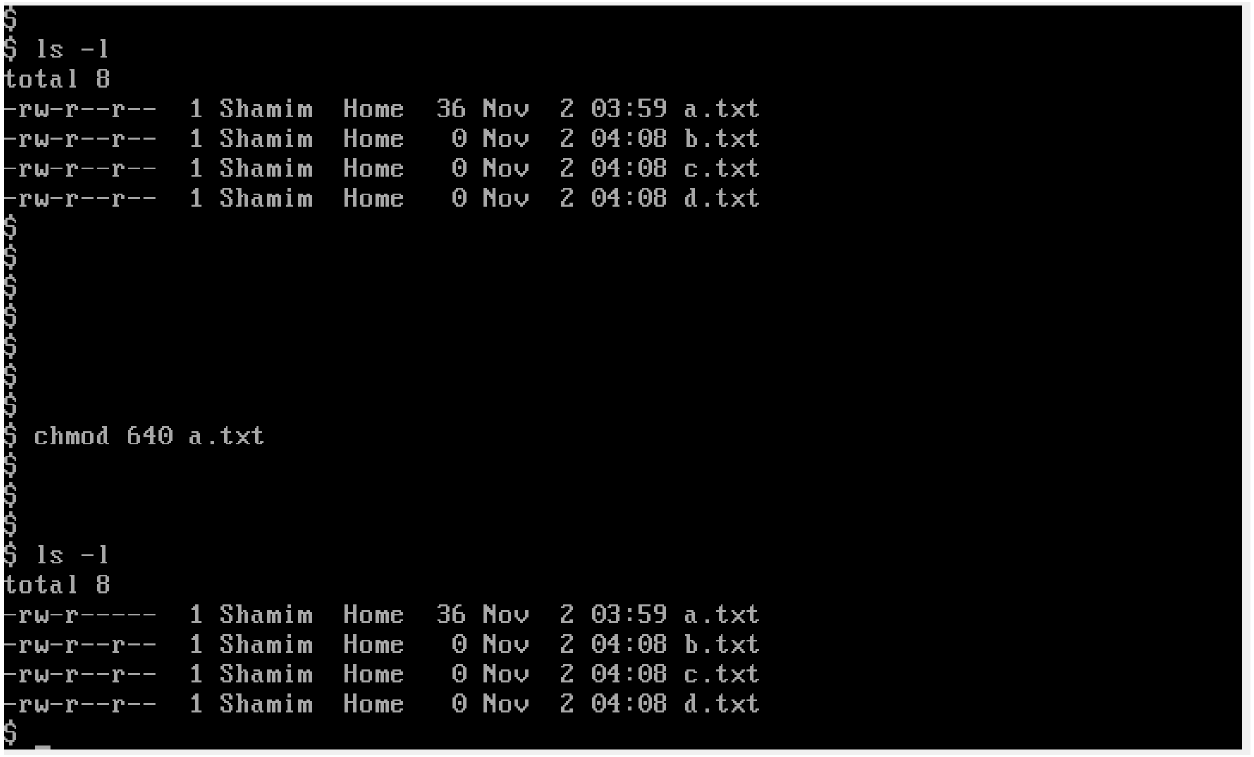
permissions defines the permissions for the owner of the file (the "user"), members of the group who owns the file (the "group"), and anyone else ("others"). There are two ways to represent these permissions: with symbols (alphanumeric characters), or with octal numbers (the digits 0 through 7).

Let's say you are the owner of a file named myfile, and you want to set its permissions so that:

1. the user can read, write, and execute it;
2. members of your group can read and execute it; and
3. others may only read it.

The chmod numerical format accepts up to four octal digits. The three rightmost digits refer to permissions for the file owner, the group, and other users. The optional leading digit, when 4 digits are given, specifies the special setuid, setgid, and sticky flags.

#	Permission	rwx
7	Read, write and execution	rwx
6	Read and write	rw-
5	Read and execution	r-x
4	Read only	r--
3	Write and execution	-we
2	Write only	-w-
1	Read only	--e
0	None	---

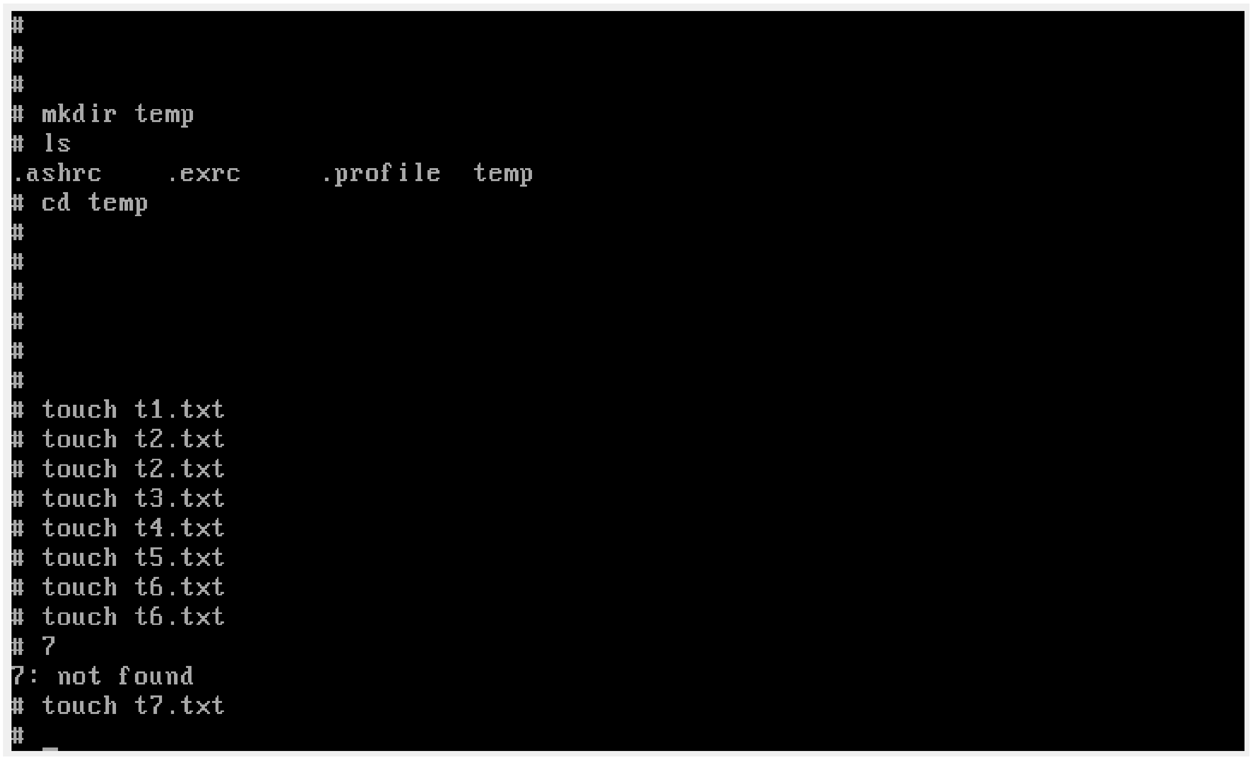


```
$
$ ls -l
total 8
-rw-r--r--  1 Shamim  Home  36 Nov  2 03:59 a.txt
-rw-r--r--  1 Shamim  Home   0 Nov  2 04:08 b.txt
-rw-r--r--  1 Shamim  Home   0 Nov  2 04:08 c.txt
-rw-r--r--  1 Shamim  Home   0 Nov  2 04:08 d.txt
$
$
$
$
$
$
$
$ chmod 640 a.txt
$
$
$
$ ls -l
total 8
-rw-r-----  1 Shamim  Home  36 Nov  2 03:59 a.txt
-rw-r--r--  1 Shamim  Home   0 Nov  2 04:08 b.txt
-rw-r--r--  1 Shamim  Home   0 Nov  2 04:08 c.txt
-rw-r--r--  1 Shamim  Home   0 Nov  2 04:08 d.txt
$ _
```

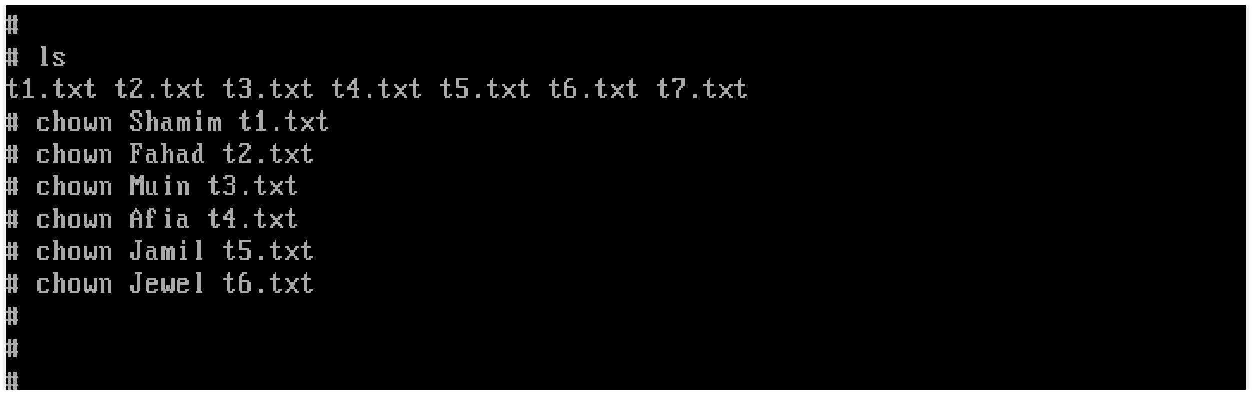
Here the permission of a.txt was change. Before changing others can read the book. Now this file is not even.

Change Or Revoke Permission

We can give authority to any user for any particular file. Such as:



```
#
#
#
# mkdir temp
# ls
.ashrc    .exrc     .profile  temp
# cd temp
#
#
#
#
#
#
# touch t1.txt
# touch t2.txt
# touch t2.txt
# touch t3.txt
# touch t4.txt
# touch t5.txt
# touch t6.txt
# touch t6.txt
# 7
7: not found
# touch t7.txt
# _
```



```
#
# ls
t1.txt t2.txt t3.txt t4.txt t5.txt t6.txt t7.txt
# chown Shamim t1.txt
# chown Fahad t2.txt
# chown Muin t3.txt
# chown Afia t4.txt
# chown Jamil t5.txt
# chown Jewel t6.txt
#
#
#
```

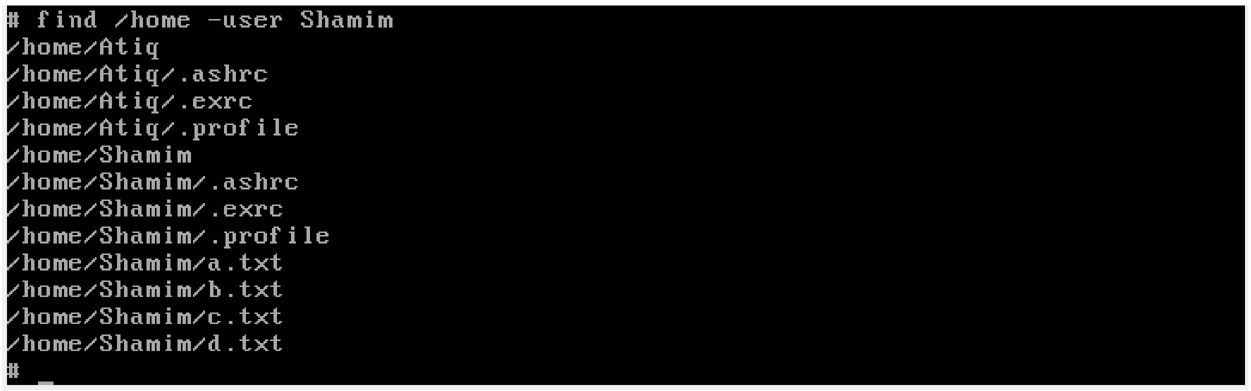
Here different owner become owner for different files.

List Of File For Particular User

You can find the list of files for a particular user. To do this the required command is:

$ find /userDirectory –user username

Here the result for Shamim‘s file search



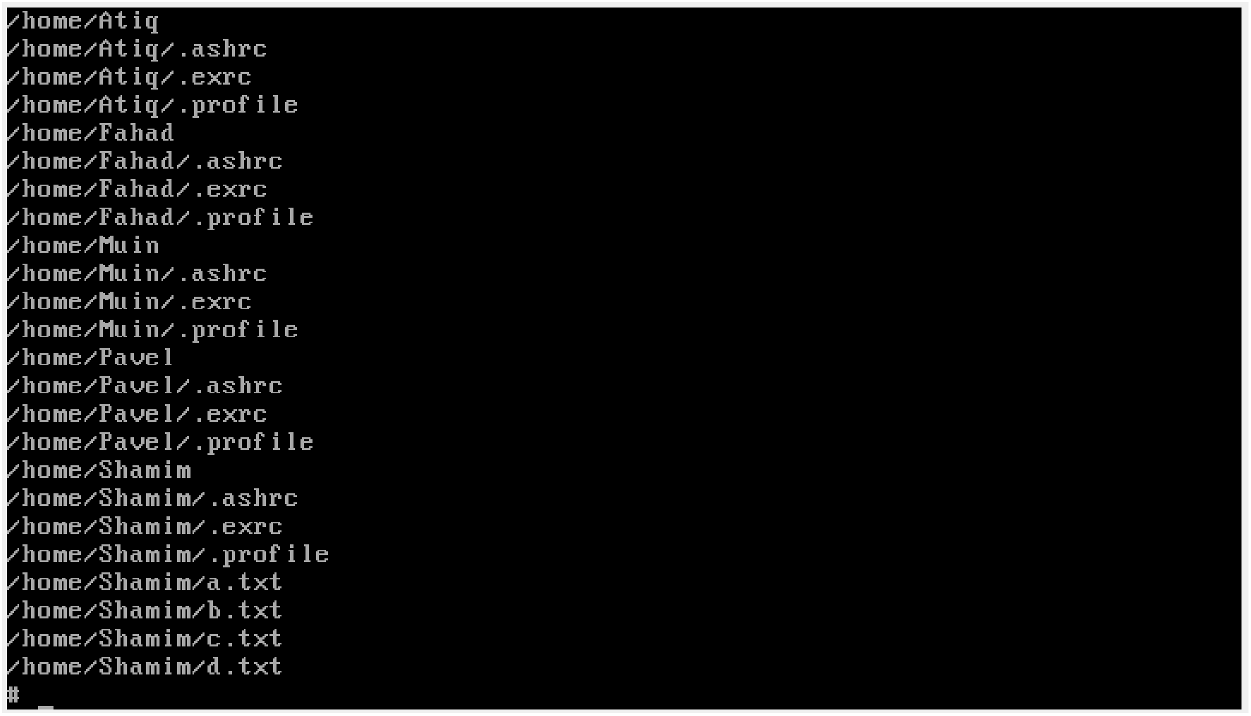
```
# find /home -user Shamim
/home/Atiq
/home/Atiq/.ashrc
/home/Atiq/.exrc
/home/Atiq/.profile
/home/Shamim
/home/Shamim/.ashrc
/home/Shamim/.exrc
/home/Shamim/.profile
/home/Shamim/a.txt
/home/Shamim/b.txt
/home/Shamim/c.txt
/home/Shamim/d.txt
# _
```

List Of File For Particular Group

You can find the list of files for a particular group. To do this the required command is:

$ find /groupDirectory –group groupname

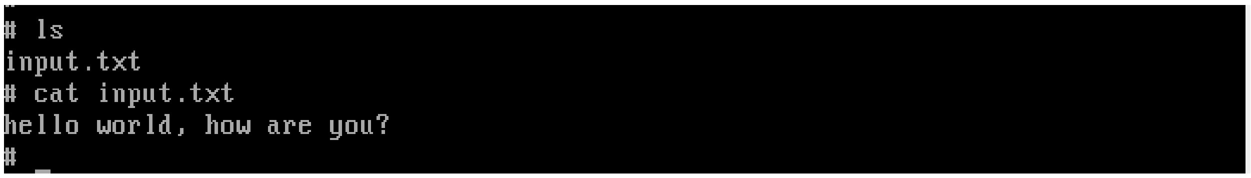
Here the result for Home‘s file search



```
/home/Atiq
/home/Atiq/.ashrc
/home/Atiq/.exrc
/home/Atiq/.profile
/home/Fahad
/home/Fahad/.ashrc
/home/Fahad/.exrc
/home/Fahad/.profile
/home/Muin
/home/Muin/.ashrc
/home/Muin/.exrc
/home/Muin/.profile
/home/Pavel
/home/Pavel/.ashrc
/home/Pavel/.exrc
/home/Pavel/.profile
/home/Shamim
/home/Shamim/.ashrc
/home/Shamim/.exrc
/home/Shamim/.profile
/home/Shamim/a.txt
/home/Shamim/b.txt
/home/Shamim/c.txt
/home/Shamim/d.txt
# _
```

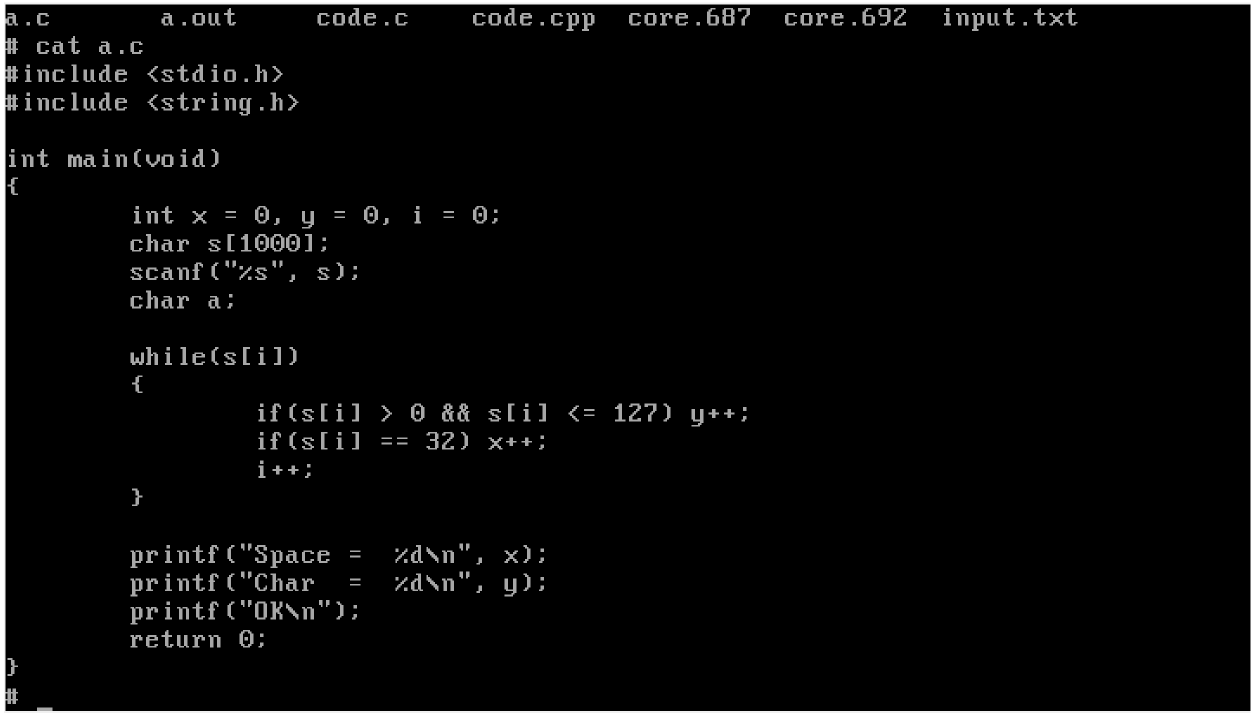
2 C Program Execution

The input file contains some text



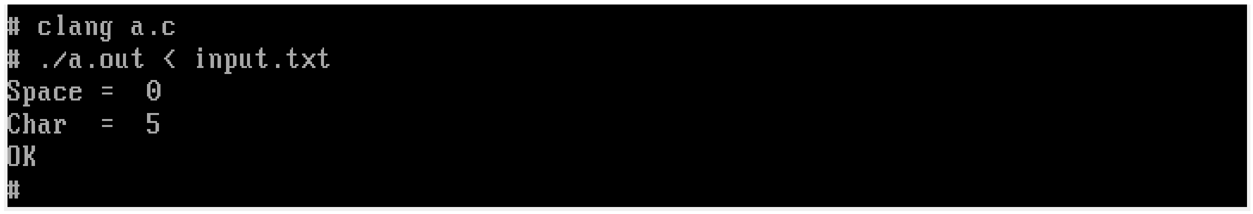
```
# ls
input.txt
# cat input.txt
hello world, how are you?
# _
```

A simple code:



```
a.c        a.out      code.c     code.cpp   core.687   core.692   input.txt
# cat a.c
#include <stdio.h>
#include <string.h>

int main(void)
{
        int x = 0, y = 0, i = 0;
        char s[1000];
        scanf("%s", s);
        char a;

        while(s[i])
        {
                if(s[i] > 0 && s[i] <= 127) y++;
                if(s[i] == 32) x++;
                i++;
        }

        printf("Space =  %d\n", x);
        printf("Char  =  %d\n", y);
        printf("OK\n");
        return 0;
}
# _
```

Output



```
# clang a.c
# ./a.out < input.txt
Space =  0
Char  =  5
OK
#
```

As minix 3 is being used here so sunread is not necessary here to compile and run c program As I told before.

Creating And Deploying Programming In MINIX (Solaris)

For solaris minix os the following process could be useful:

Resource :: https://view.officeapps.live.com/op/view.aspx?src=http://www.cis.syr.edu/~wedu/minix/projects/creprog.doc

This manual will illustrate as to how you can create and deploy your programs in minix. For Compiling Minix User Programs, we use the sunread and sunwrite commands. I have this example for your reference, so you start running your programs with ease. Please feel free to use it

Type cd /home/seed/785/san/smx/src/tools

Create your C program (let's call ours helloworld.c)

Type mcc helloworld.c –o helloworld (Note: you can create it anywhere, but you would need to put the program entry in your Makefile)

Run minix

Login as root

Type sunread helloworld > helloworld

Type chmod +x helloworld

Run helloworld

Suppose you want to run the following test.c program in Minix:

```
#include <stdio.h>

main (int argc, char **argv)

{

        printf ("Testing For Minix\n");

}
```

A copy of this program is located in your src/tools directory. Compile the program with the command mcc test.c -o test. Next, while running Minix you can type the command sunread test > test to read in the compiled test program. You can't read it into /usr/bin since those are read-only file systems. Next, make the program executable with the command chmod +x test. Then type test to run the program. There is an analogous sunwrite command, which writes a file from a Minix file system into a Solaris file system. Type man sunread and man sunwrite, in Minix, for more information.

NOTE: if you recompile your libraries (make in src/lib), the newly recompiled libraries will not be used until you then recompile the code that uses those libraries. So if you recompile your libraries, then recompile Minix, the Minix test code (if needed), and any Minix user programs that you've written.

3.1 UID Implementation Questions (i)

Figure out why "passwd", "chsh", and "su" commands need to be Set-UID programs. What will happen if they are not? If you are not familiar with these programs, you should first learn what they can do.

commands need to be Set-UID programs, because if they were not, any user would be able to change passwords, or change things dealing with the os, and they would be just as powerful as the root user. with the those commands being setuid programs, that ensures that roots are the only ones with high access privileges and that normal users are restricted from certain things.

3.2 UID Implementation Questions (ii)

Read the OS source codes of Minix, and figure out how Set-UID is implemented in the system. You are required to answer the below questions, and also identify the corresponding codes.

1. How does the operating system recognize whether a file is a Set-UID?

2. What are the procedures the OS performs when it recognizes the Set-UID file?

3. How does Set-UID affects the access control (i.e., when a Set-UID process tries to access a file, how does the OS check whether the process can access the file or not?).

Answer 1:

Operating system recognize whether a file is a set – UID by checking an access right flag.

Answer 2:

It does a validity check; then it get the exec file name and sees if the file is executable; fetches the stack from the user before the old core image; checks to see if the process' text can be shared with that one of already running; saves file id to allow it to be shared; patches up stack and copy it from MM to new core image; read in text and data segments; take care of setuid / setgid bits; save offset to initial argc; fix mproc fields, tell kernel that exec is done, reset caught sigs.

Answer 3:

It first checks to see if the mode is correct; then it temporarily opens the file whose access is to be checked; and finally, it checks the permissions.