Bahir Dar Institution of Technology

Faculty of Computing

Information Technology

Operating Systems Individual Assignment

Student’s Name: Misgan Kassew

Student’s ID: BDU1602156

Submitted to: Lecture Wondmu

Submission Date: 16/08/2017E.C

Table Of Contents

Introduction to Slackware Linux OS........2

Objectives........2

Requirements........2

Installation Steps........4

Issues (Problems Faced) and their solution........13

System call implementation........14

Filesystem Support........17

Advantages and Disadvantages of Slackware Linux OS........18

Future Outlook........18

Virtualization in Modern Operating Systems........19

Conclusion........21

Introduction

Operating systems are the backbone of modern computing, responsible for managing hardware resources and providing essential services to applications. Gaining hands-on experience with operating systems helps students build a strong foundation in both system design and real-world computing. This assignment focuses on three core areas: installing Slackware Linux on VMware Workstation, understanding virtualization in modern operating systems, and implementing a Linux system call.

Slackware Linux was selected for installation due to its simplicity, stability, and traditional UNIX-like structure. Installing it in a virtual environment using VMware Workstation enables experimentation without the risk of damaging the host system, promoting safe and effective learning.

Virtualization, the second focus area, allows multiple operating systems to run on a single physical machine. It plays a critical role in today's computing environment by optimizing resource usage, improving security, and enabling rapid testing and deployment.

Finally, system calls are the interface between user applications and the OS kernel. Implementing a system call such as clock_gettime() demonstrates how user programs interact with low-level system services, enhancing understanding of Linux internals and time-related operations.

Objectives

The main objective of this assignment is to develop practical and conceptual understanding of core operating system functionalities through installation, virtualization, and system-level programming. By installing Slackware Linux on VMware Workstation, the goal is to gain hands-on experience with manual system setup, user account configuration, and resource allocation in a virtual environment. This is supported by exploring the concept of virtualization, including its purpose, advantages, and how it enables flexible, secure, and efficient computing. Additionally, implementing a system call like clock_gettime() aims to deepen understanding of how user-space applications interact with kernel-level services, thereby reinforcing low-level programming skills and exposing the inner workings of time management within an operating system.

Background and Motivation of Slackware Linux Operating system

Slackware Linux was created in 1993 by Patrick Volkerding with the goal of providing a stable, secure, and Unix-like Linux distribution. Unlike many modern Linux systems that focus heavily on automation and user-friendliness, Slackware maintains a simple design that stays close to UNIX principles. This makes it especially valuable for users who want full control over their system and a deeper understanding of Linux internals.

Slackware Linux was one of the earliest Linux distributions and has remained one of the most traditional. It was developed to offer a complete and consistent Linux environment that is free from unnecessary complexity. Its design avoids automatic system configuration, which gives users full control over how their system behaves.

Slackware is often chosen by learners and professionals who want to gain real hands-on experience with Linux, especially in tasks like manual configuration, software compilation, and system optimization.

Slackware is especially suitable for educational purposes and those who want a deeper understanding of Linux without relying on heavy graphical tools or automatic scripts.

Motivation behind Slackware includes the following

✓ Offering a lightweight and minimal system for performance and customization.

✓ Providing an educational platform for system administrators and advanced users.

✓ Ensuring long-term stability by avoiding unnecessary changes and sticking to well-tested tools.

Requirements

A successful installation of Slackware Linux on VMware Workstation requires the following hardware and software resources:

Hardware Requirements

✓ Processor- Intel/AMD 64-bit capable CPU (Dual-core or higher recommended)

✓RAM- Minimum 2 GB (4 GB or more is recommended for a smoother experience)

✓ Disk Space- At least 15 GB of free space dedicated to the virtual machine

✓Graphics- Standard VGA-compatible graphics card (VM-compatible)

Software Requirements

✓ Host Operating System- Windows, Linux, or macOS (the OS running VMware)

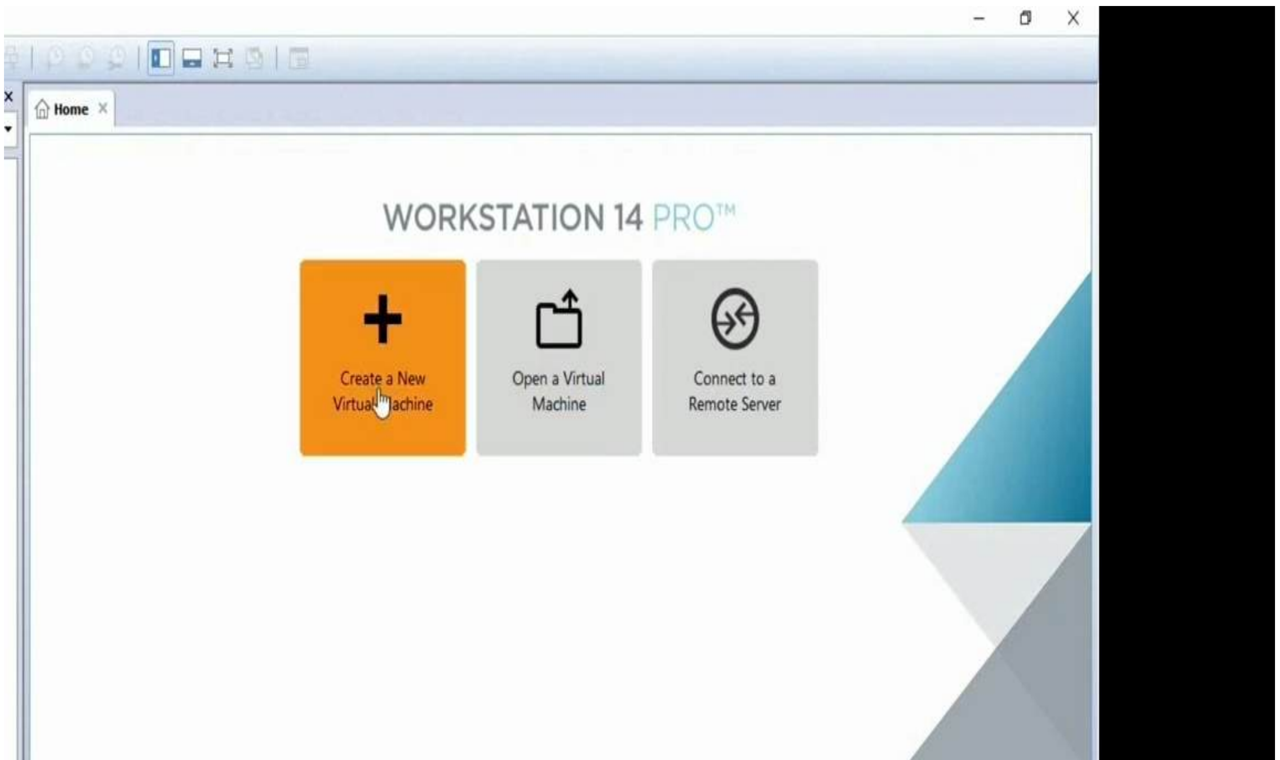
✓ VMware Workstation- A virtualization platform used to create and run virtual machines. Free alternatives like VMware Workstation Player or Oracle VirtualBox may also work.

✓ Slackware Linux ISO Image- A bootable disk image file containing the latest version of Slackware (can be downloaded from slackware.com or a mirror site)

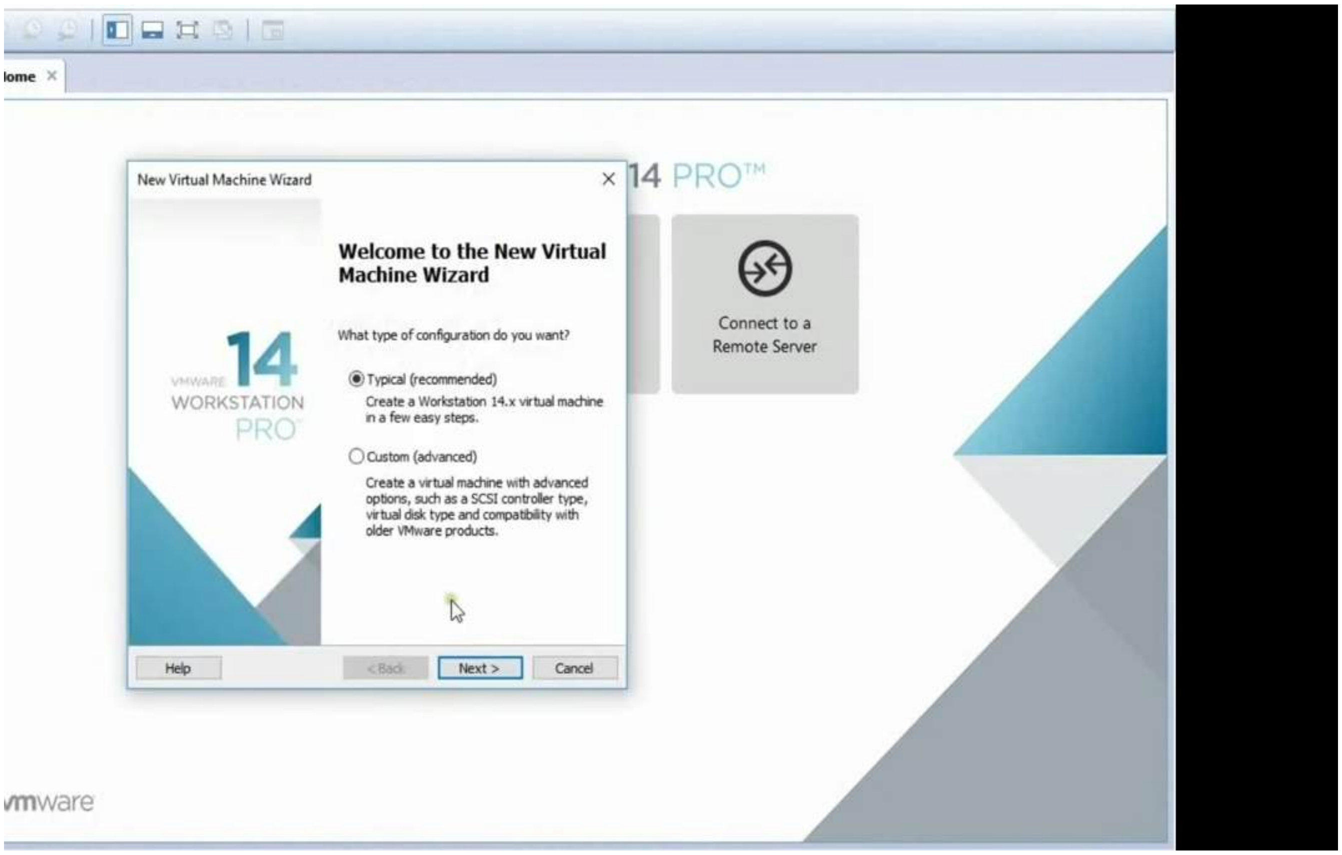
✓VMware Tools (optional)- For better integration between the host and guest systems, such as clipboard sharing and automatic screen resizing

Installation Steps

Step 1. Open VMware Workstation and click “Create a New Virtual Machine”.



Step 2. Select “Typical (recommended)” and click “Next”.



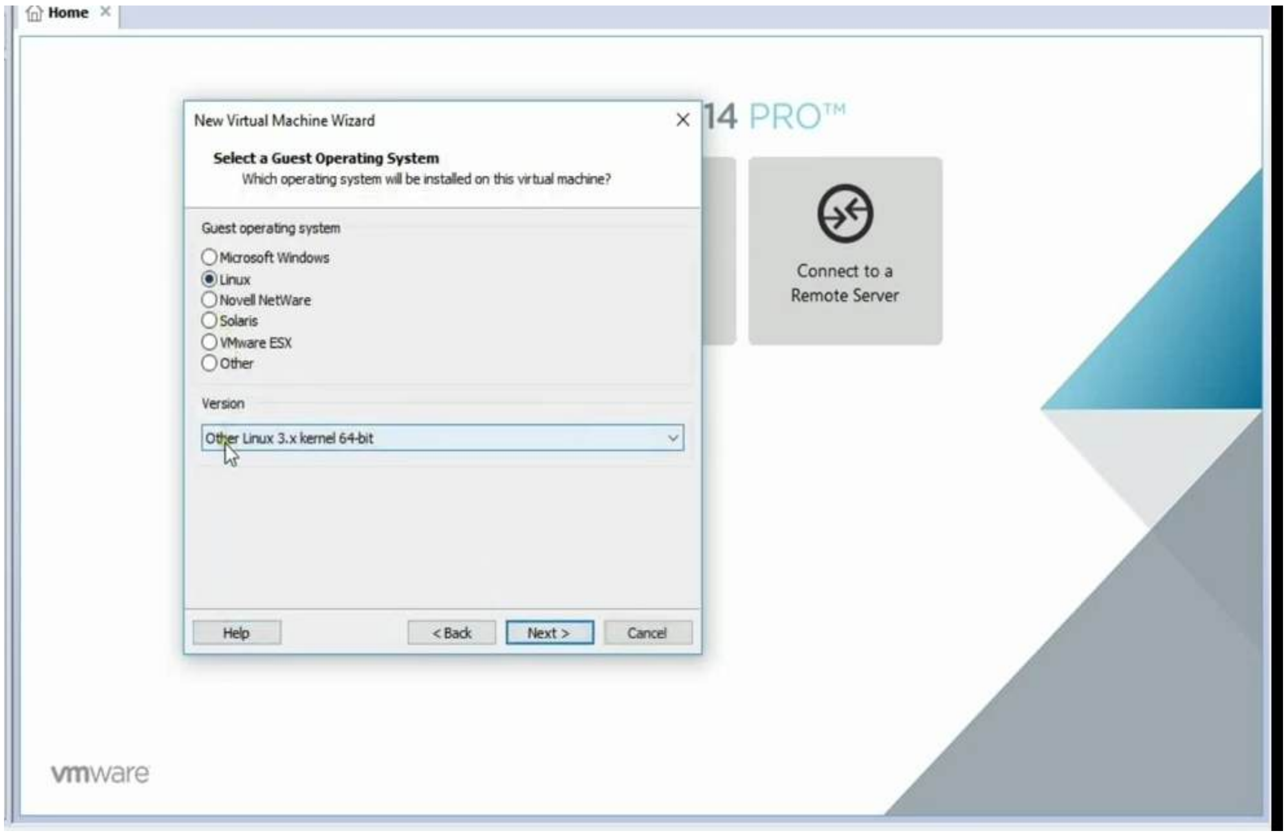
New Virtual Machine Wizard

Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

- [✓ Typical (recommended)] Create a Workstation 14.x virtual machine in a few easy steps.
- Custom (advanced) Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

Step 3. Select “Linux” as the operating system and “Other Linux 4.x or later kernel 64-bit” as version.



New Virtual Machine Wizard

Select a Guest Operating System

Which operating system will be installed on this virtual machine?

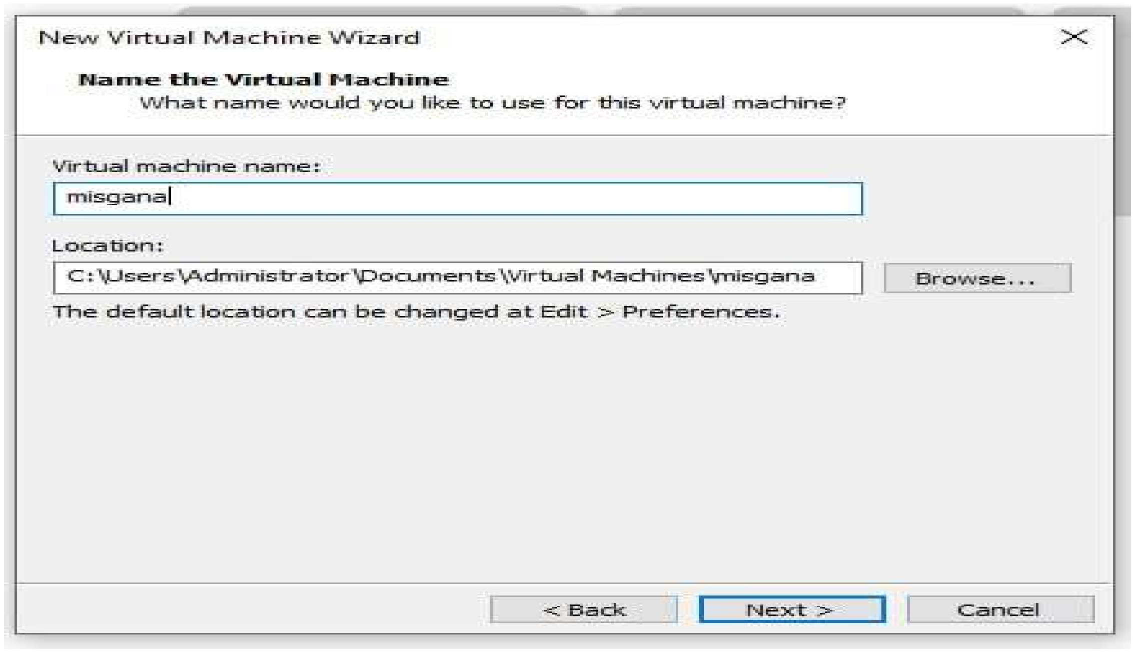
Guest operating system

- Microsoft Windows
- [✓ Linux]
- Novell NetWare
- Solaris
- VMware ESX
- Other

Version

Other Linux 3.x kernel 64-bit

Step 4. Name the virtual machine misgana and choose a location to save it.


New Virtual Machine Wizard

Name the Virtual Machine

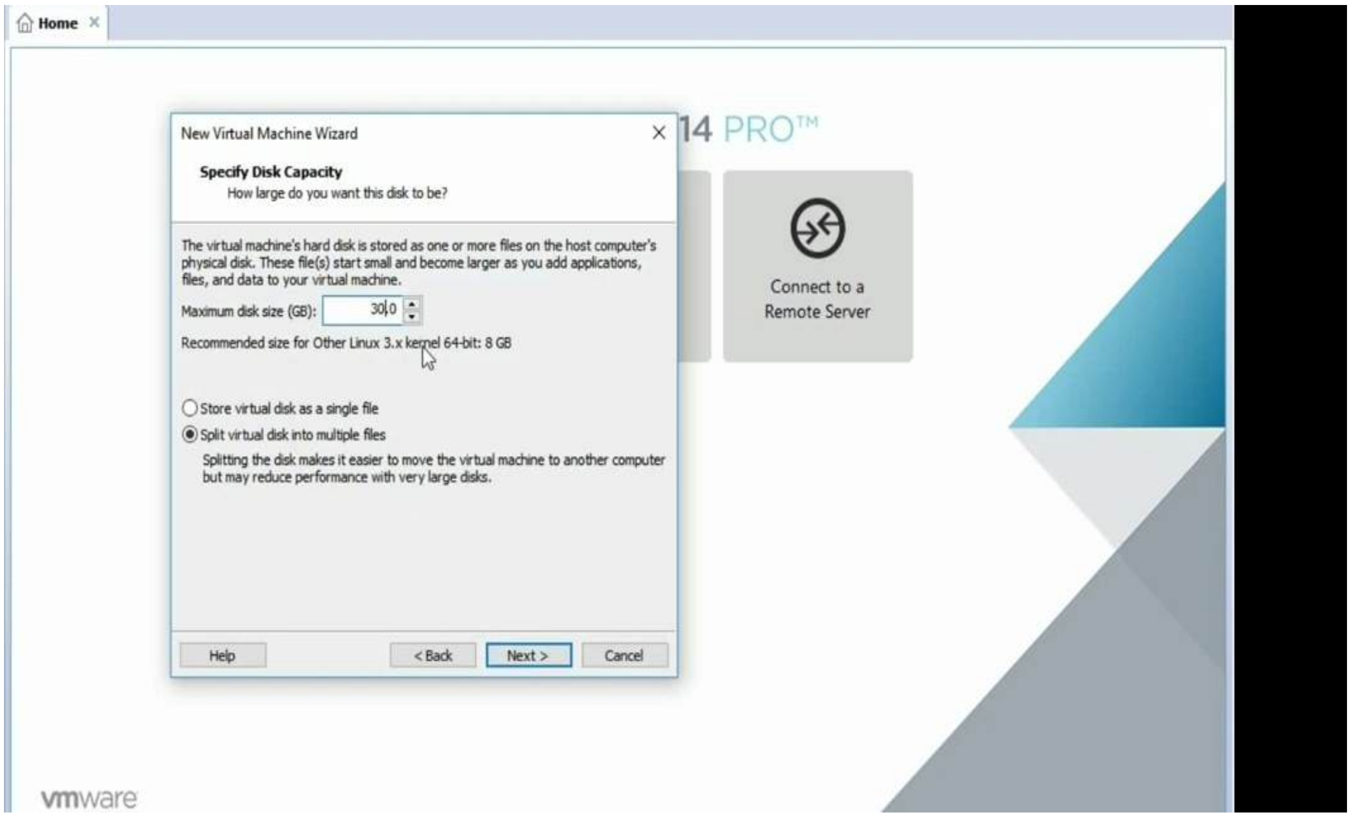
What name would you like to use for this virtual machine?

Virtual machine name: misgana

Location: C:\Users\Administrator\Documents\Virtual Machines\misgana

The default location can be changed at Edit > Preferences.

Step 5. Set the disk size to 30GB and choose “Store virtual disk as a single file”.


New Virtual Machine Wizard

Specify Disk Capacity

How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

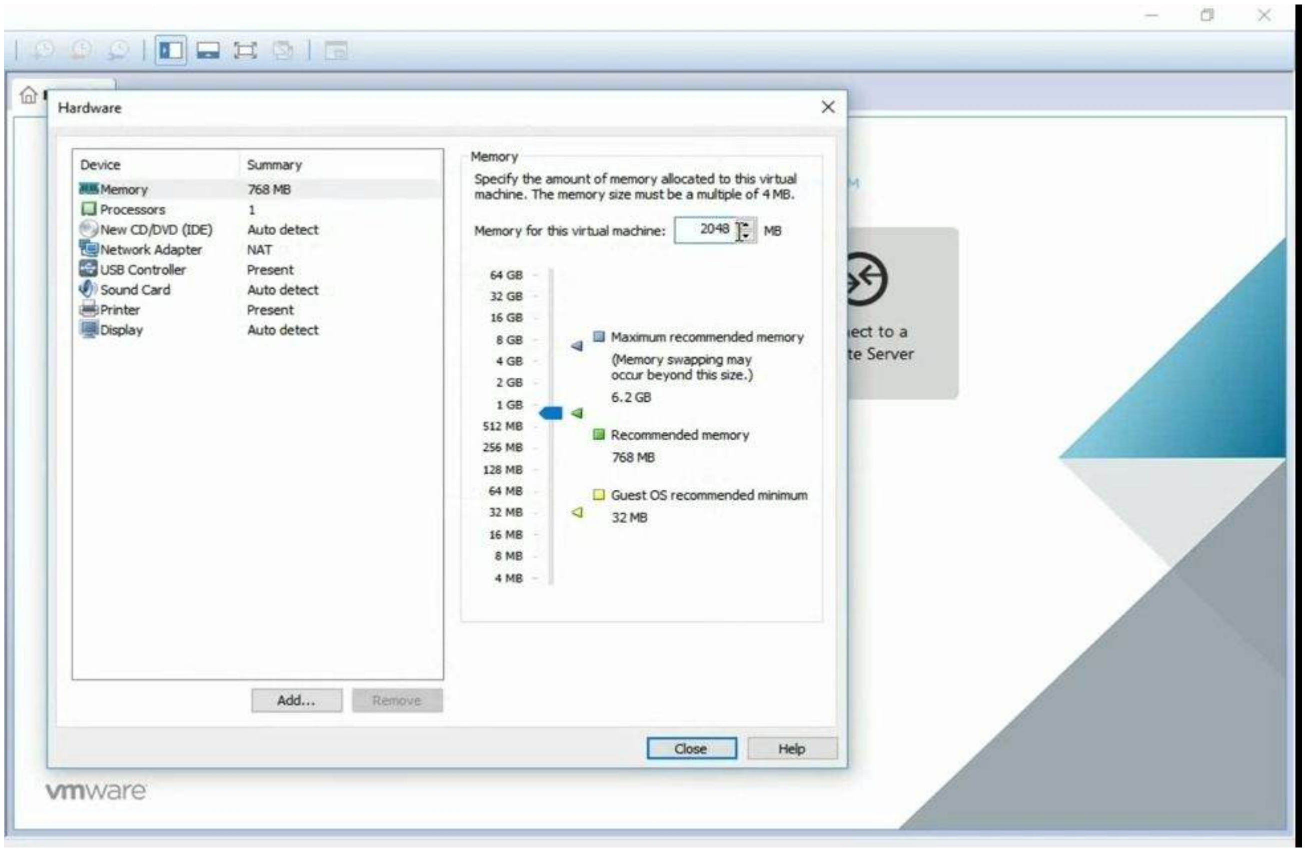
Maximum disk size (GB): 30.0

Recommended size for Other Linux 3.x kernel 64-bit: 8 GB

Store virtual disk as a single file [✓ Split virtual disk into multiple files]

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Step 6. Allocate 2GB memory (2048 MB) and complete the VM setup by clicking “Finish”.



Hardware

Device	Summary
Memory	768 MB
Processors	1
New CD/DVD (IDE)	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

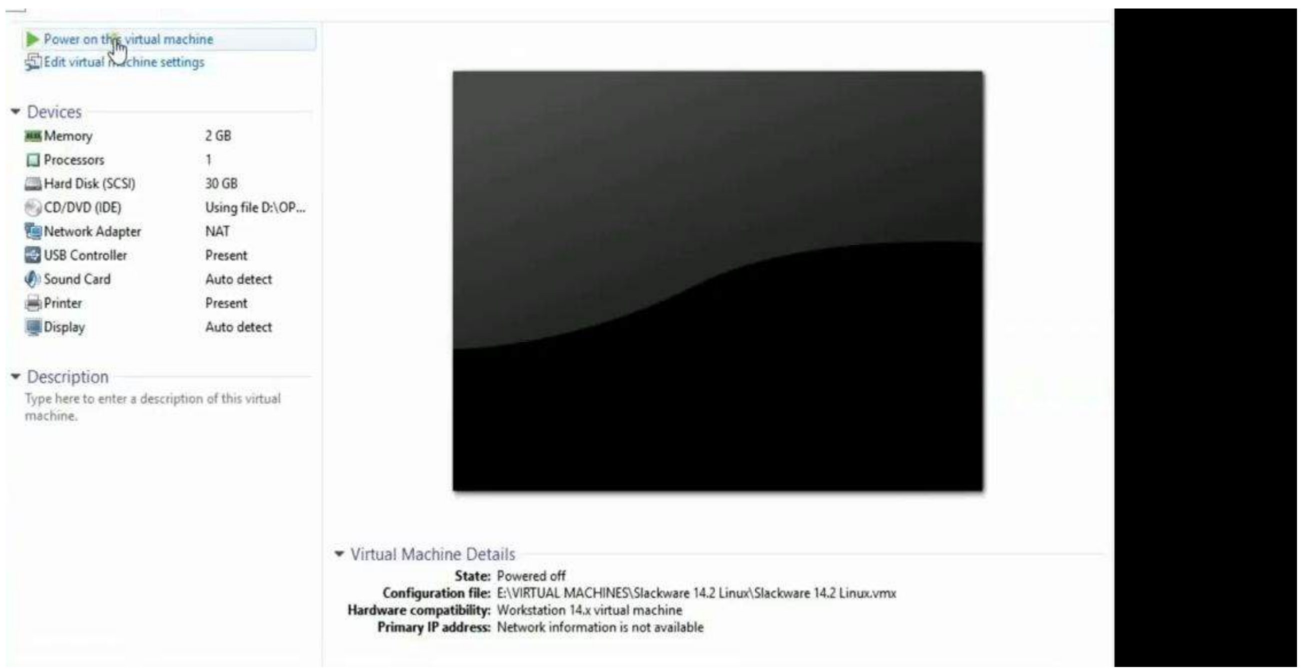
Memory for this virtual machine: 2048 MB

Maximum recommended memory (Memory swapping may occur beyond this size.) 6.2 GB

Recommended memory 768 MB

Guest OS recommended minimum 32 MB

Step 7. Power on the VM and wait for the Slackware boot screen.



Devices

Device	Summary
Memory	2 GB
Processors	1
Hard Disk (SCSI)	30 GB
CD/DVD (IDE)	Using file D:\OP...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

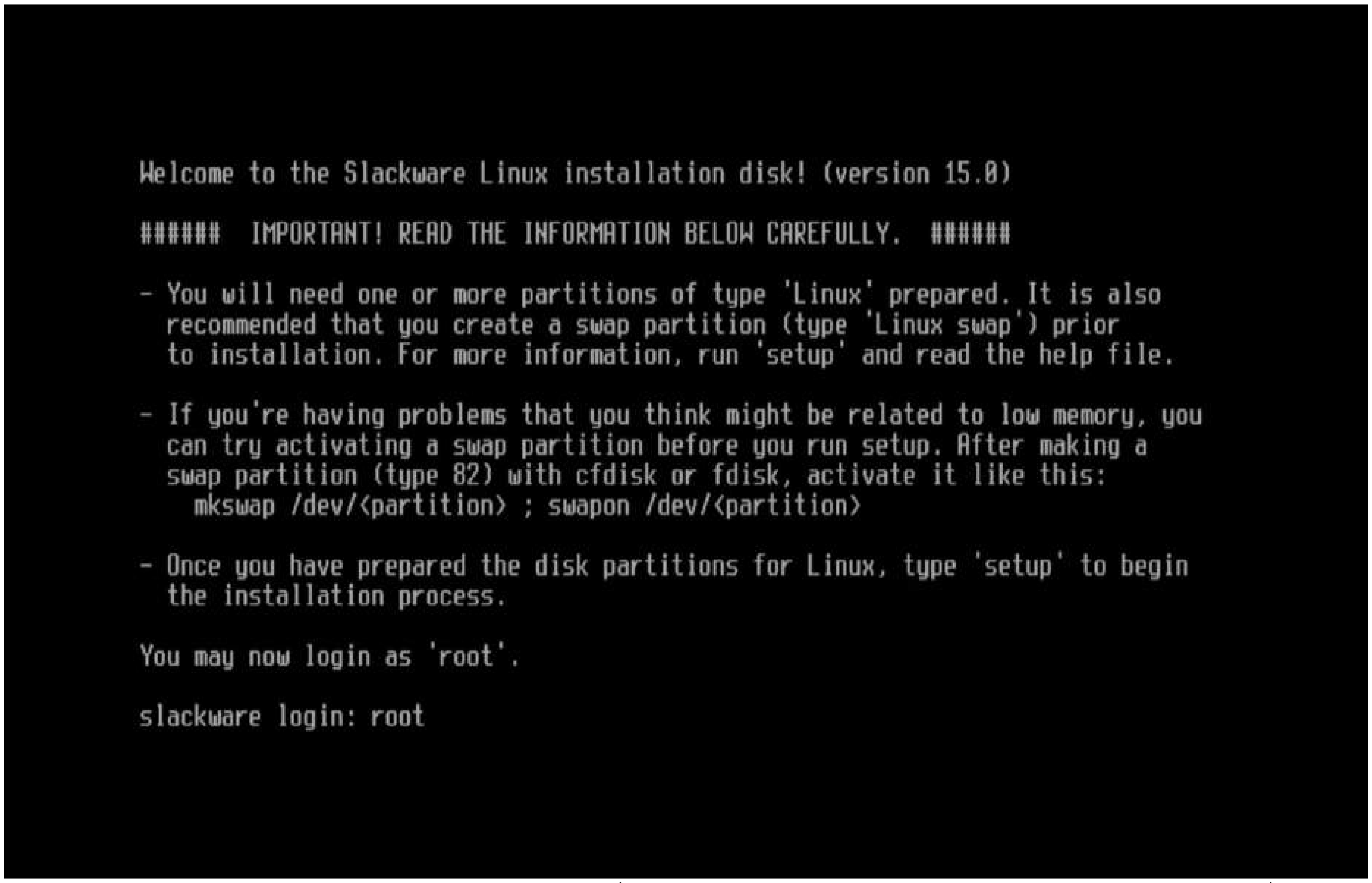
Description

Type here to enter a description of this virtual machine.

Virtual Machine Details

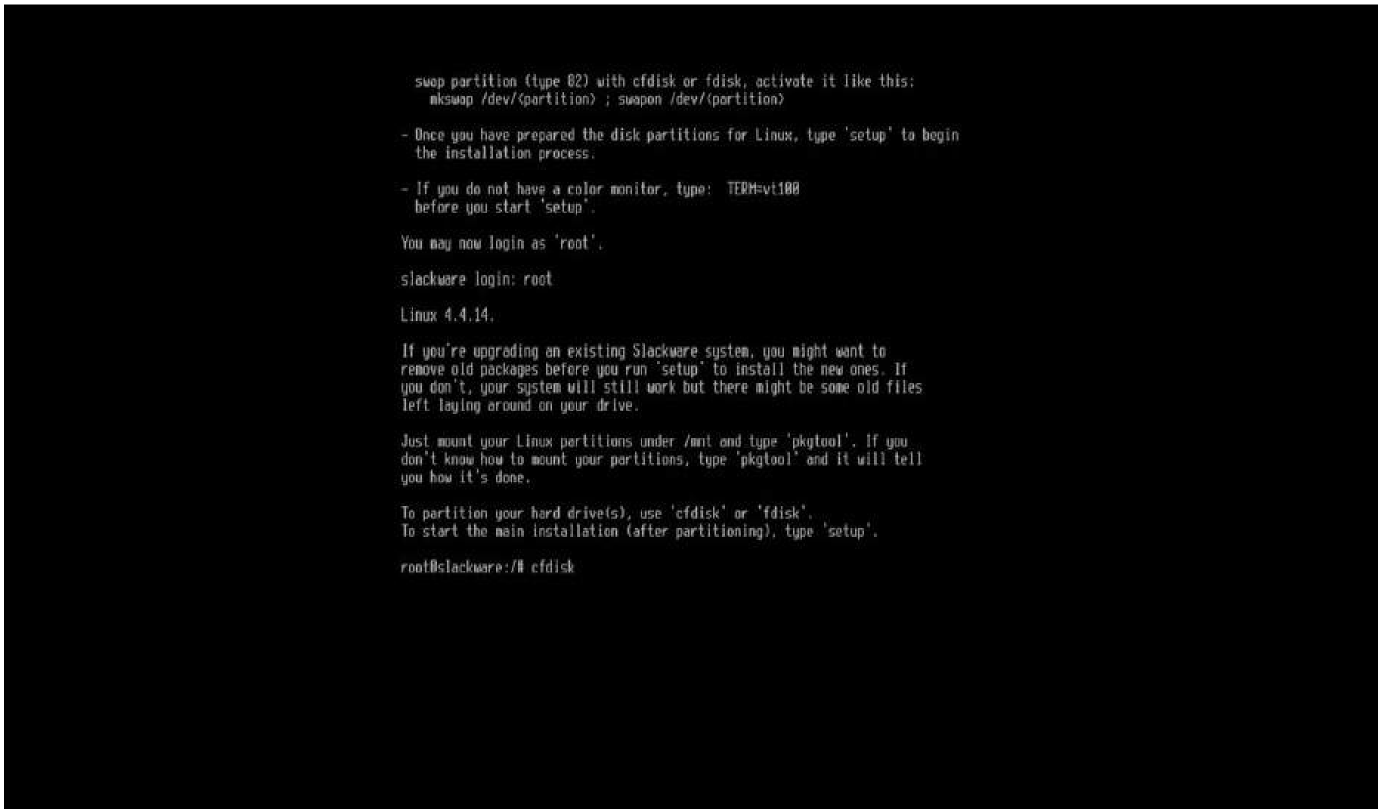
State:	Powered off
Configuration file:	E:\VIRTUAL MACHINES\Slackware 14.2 Linux\Slackware 14.2 Linux.vmx
Hardware compatibility:	Workstation 14.x virtual machine
Primary IP address:	Network information is not available

Step 8. Login as root (no password required).



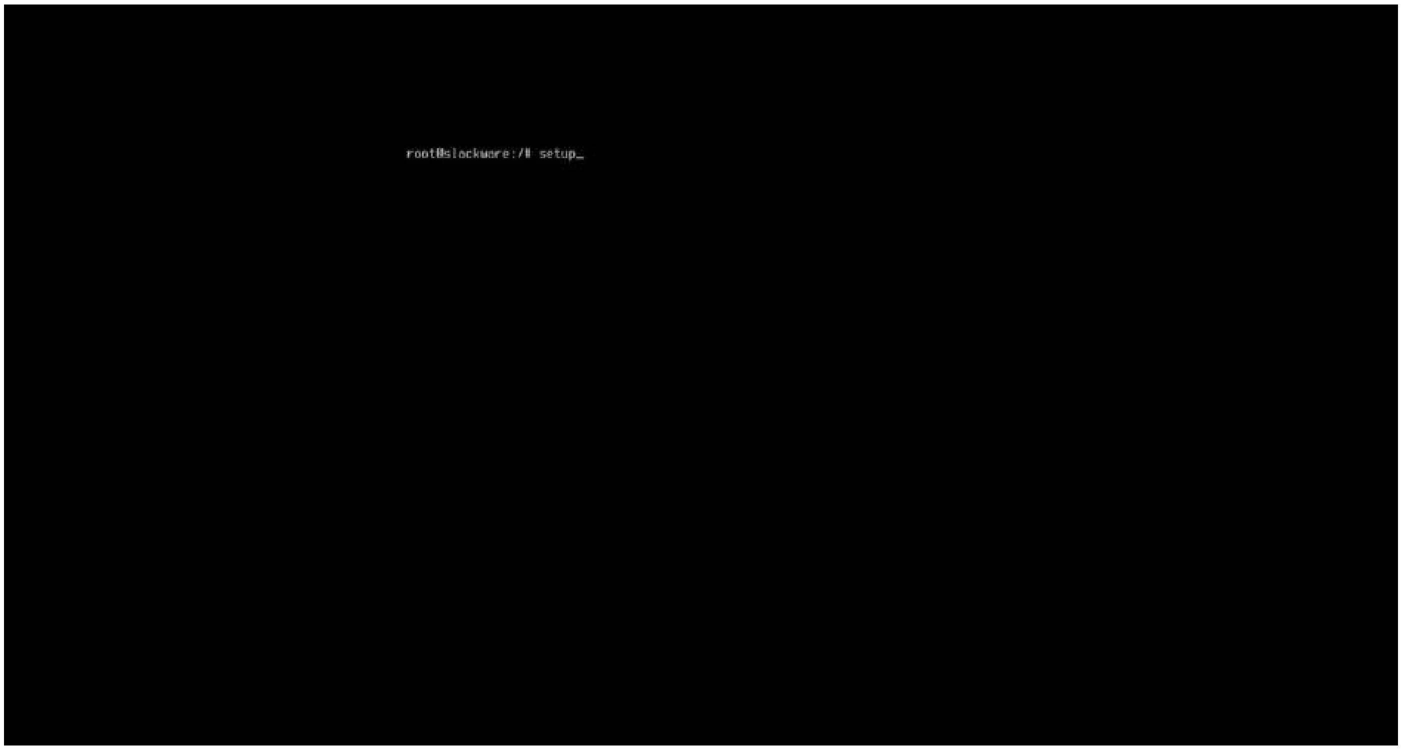
```
Welcome to the Slackware Linux installation disk! (version 15.0)

######  IMPORTANT! READ THE INFORMATION BELOW CAREFULLY.  ######

- You will need one or more partitions of type 'Linux' prepared. It is also
  recommended that you create a swap partition (type 'Linux swap') prior
  to installation. For more information, run 'setup' and read the help file.

- If you're having problems that you think might be related to low memory, you
  can try activating a swap partition before you run setup. After making a
  swap partition (type 82) with cfdisk or fdisk, activate it like this:
    mkswap /dev/<partition> ; swapon /dev/<partition>

- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.

You may now login as 'root'.

slackware login: root
```

Step 9. Type cfdisk and create partitions (primary root partition with ext4, and optional swap).



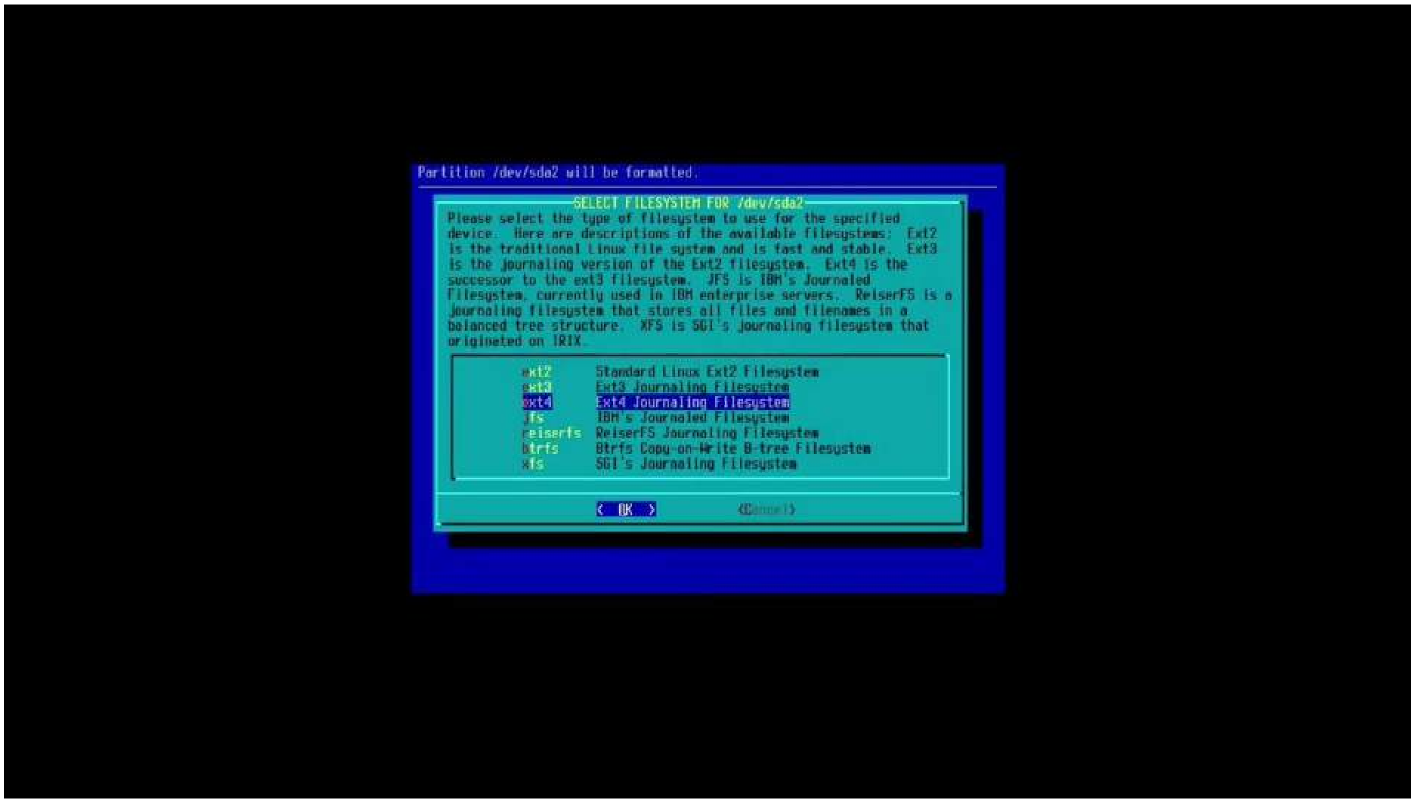
```
  swap partition (type 82) with cfdisk or fdisk, activate it like this:
    mkswap /dev/<partition> ; swapon /dev/<partition>

- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.

- If you do not have a color monitor, type:  TERM=vt100
  before you start 'setup'.

You may now login as 'root'.

slackware login: root

Linux 4.4.14.

If you're upgrading an existing Slackware system, you might want to
remove old packages before you run 'setup' to install the new ones. If
you don't, your system will still work but there might be some old files
left laying around on your drive.

Just mount your Linux partitions under /mnt and type 'pkgtool'. If you
don't know how to mount your partitions, type 'pkgtool' and it will tell
you how it's done.

To partition your hard drive(s), use 'cfdisk' or 'fdisk'.
To start the main installation (after partitioning), type 'setup'.

root@slackware:/# cfdisk
```

Step 10. Exit cfdisk, then type setup to start the Slackware installation menu.



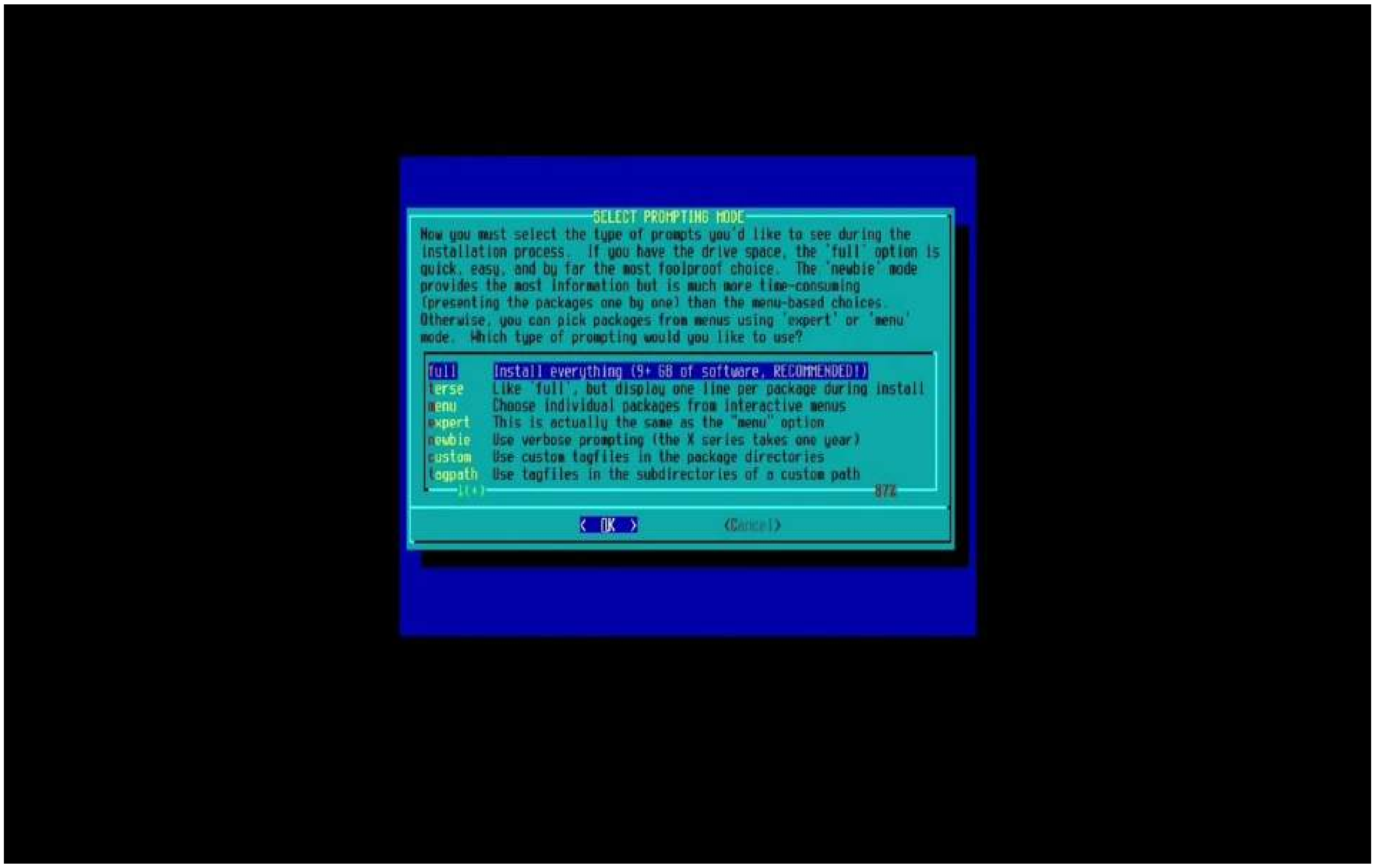
```
root@slackware:/# setup_
```

Step 11. Select the root partition, format it with ext4, and enable the swap partition.



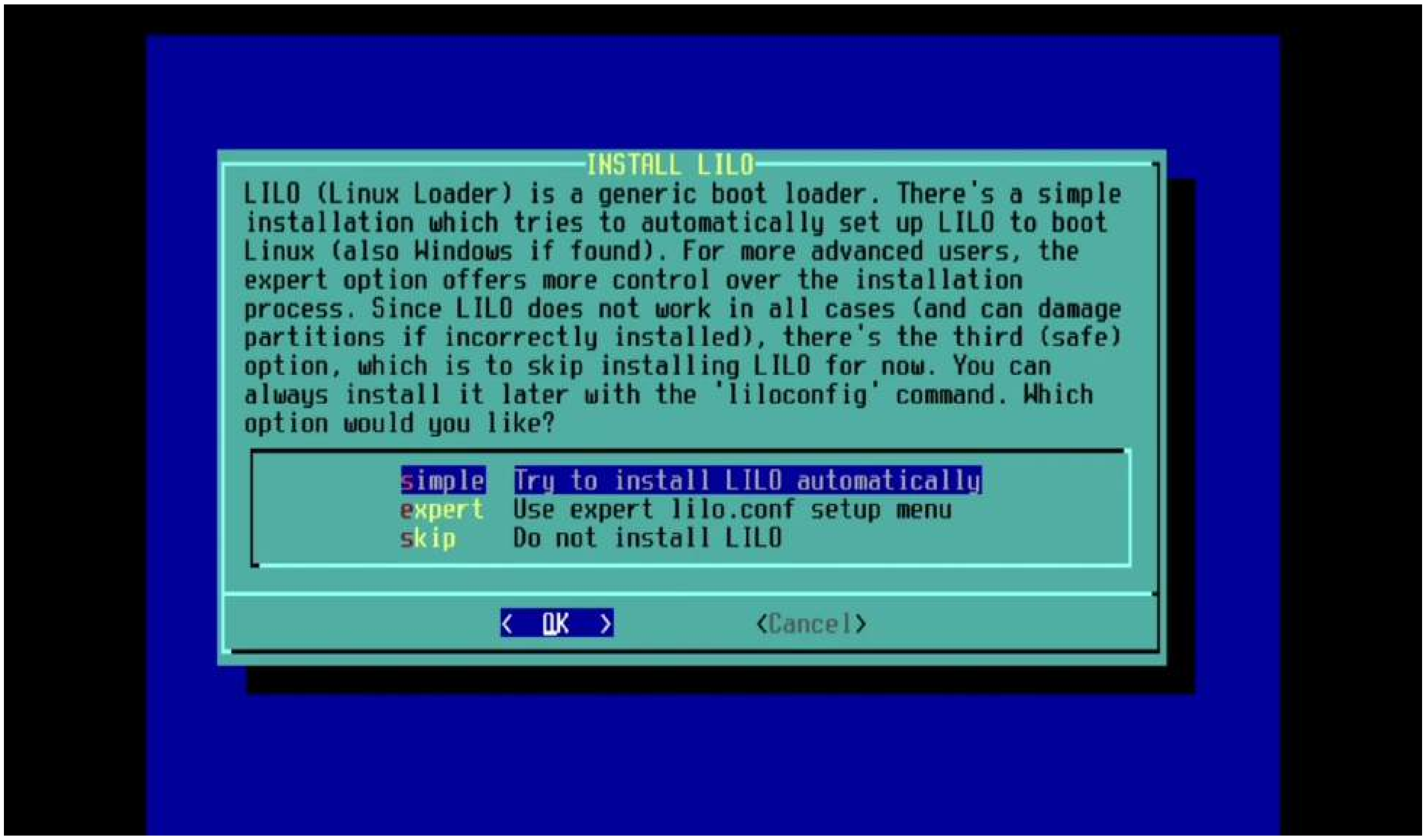
```
Partition /dev/sda2 will be formatted.

SELECT FILESYSTEM FOR /dev/sda2
Please select the type of filesystem to use for the specified
device.  Here are descriptions of the available filesystems:  Ext2
is the traditional Linux file system and is fast and stable.  Ext3
is the journaling version of the Ext2 filesystem.  Ext4 is the
successor to the ext3 filesystem.  JFS is IBM's Journaled
Filesystem, currently used in IBM enterprise servers.  ReiserFS is a
journaling filesystem that stores all files and filenames in a
balanced tree structure.  XFS is SGI's journaling filesystem that
originated on IRIX.

        ext2        Standard Linux Ext2 Filesystem
        ext3        Ext3 Journaling Filesystem
        ext4        Ext4 Journaling Filesystem
        jfs         IBM's Journaled Filesystem
        reiserfs    ReiserFS Journaling Filesystem
        btrfs       Btrfs Copy-on-Write B-tree Filesystem
        xfs         SGI's Journaling Filesystem

              <  OK  >          <Cancel>
```

Step 12. Choose installation source (CD/DVD) and select “Full” package Installation .



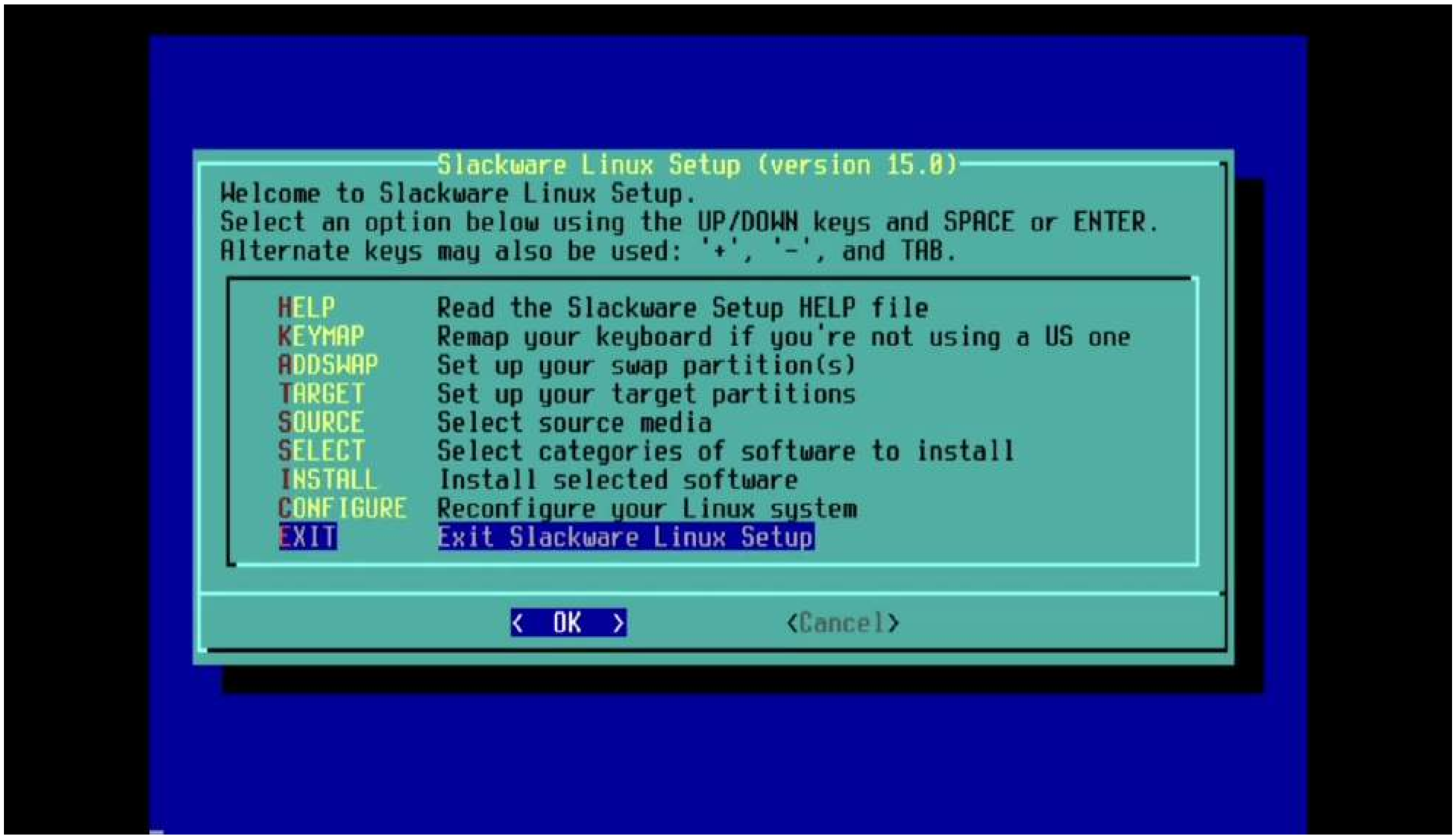
```
SELECT PROMPTING MODE
Now you must select the type of prompts you'd like to see during the
installation process.  If you have the drive space, the 'full' option is
quick, easy, and by far the most foolproof choice.  The 'newbie' mode
provides the most information but is much more time-consuming
(presenting the packages one by one) than the menu-based choices.
Otherwise, you can pick packages from menus using 'expert' or 'menu'
mode.  Which type of prompting would you like to use?

full     Install everything (9+ GB of software, RECOMMENDED!)
terse    Like 'full', but display one line per package during install
menu     Choose individual packages from interactive menus
expert   This is actually the same as the "menu" option
newbie   Use verbose prompting (the X series takes one year)
custom   Use custom tagfiles in the package directories
tagpath  Use tagfiles in the subdirectories of a custom path
   1(+)                                                            87%

                 < OK >          <Cancel>
```

Step 13. Install LILO bootloader (simple method), set the timezone, and configure the network.



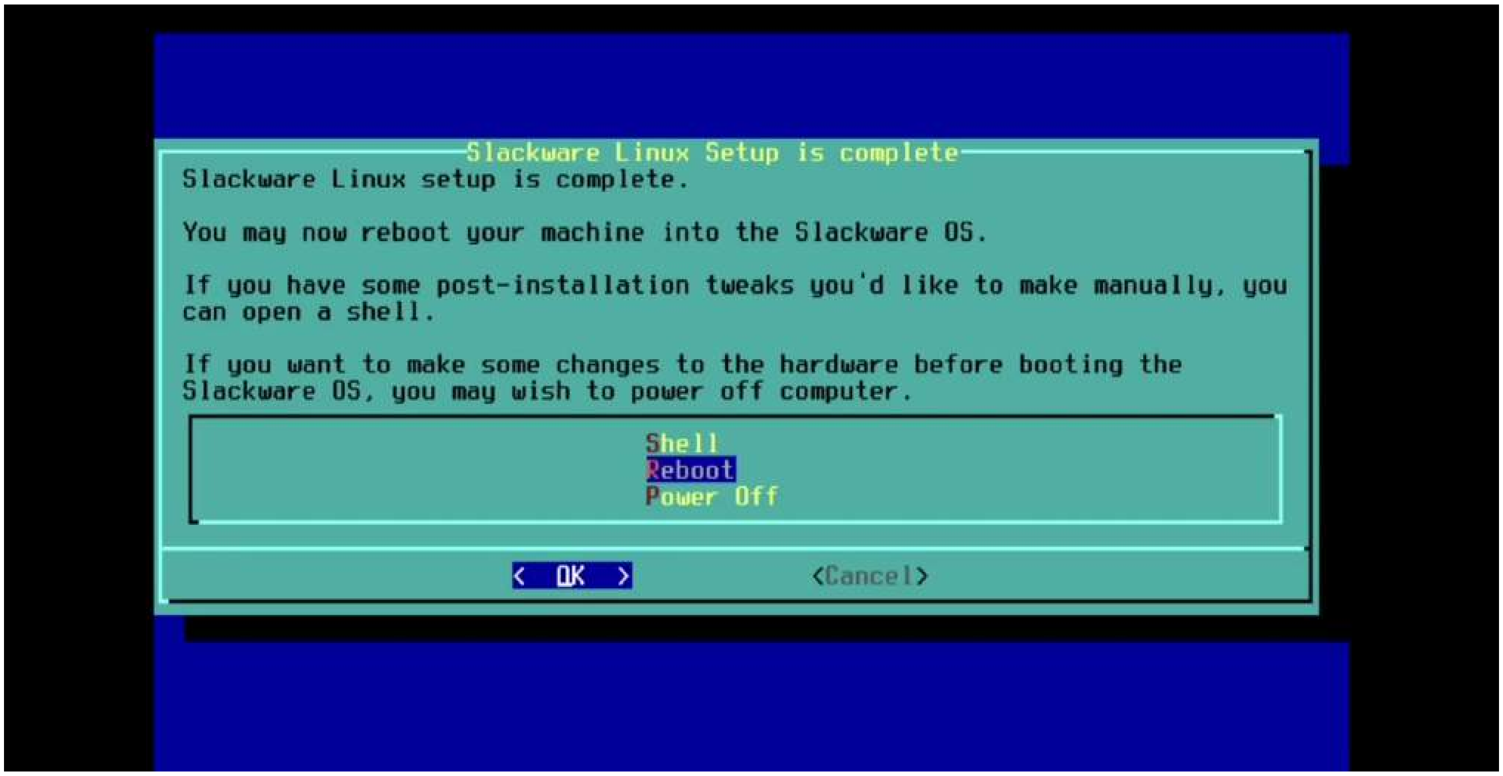
```
INSTALL LILO
LILO (Linux Loader) is a generic boot loader. There's a simple
installation which tries to automatically set up LILO to boot
Linux (also Windows if found). For more advanced users, the
expert option offers more control over the installation
process. Since LILO does not work in all cases (and can damage
partitions if incorrectly installed), there's the third (safe)
option, which is to skip installing LILO for now. You can
always install it later with the 'liloconfig' command. Which
option would you like?

        simple   Try to install LILO automatically
        expert   Use expert lilo.conf setup menu
        skip     Do not install LILO

            < OK >            <Cancel>
```

Step 14. Select exit Slackware Linux setup.



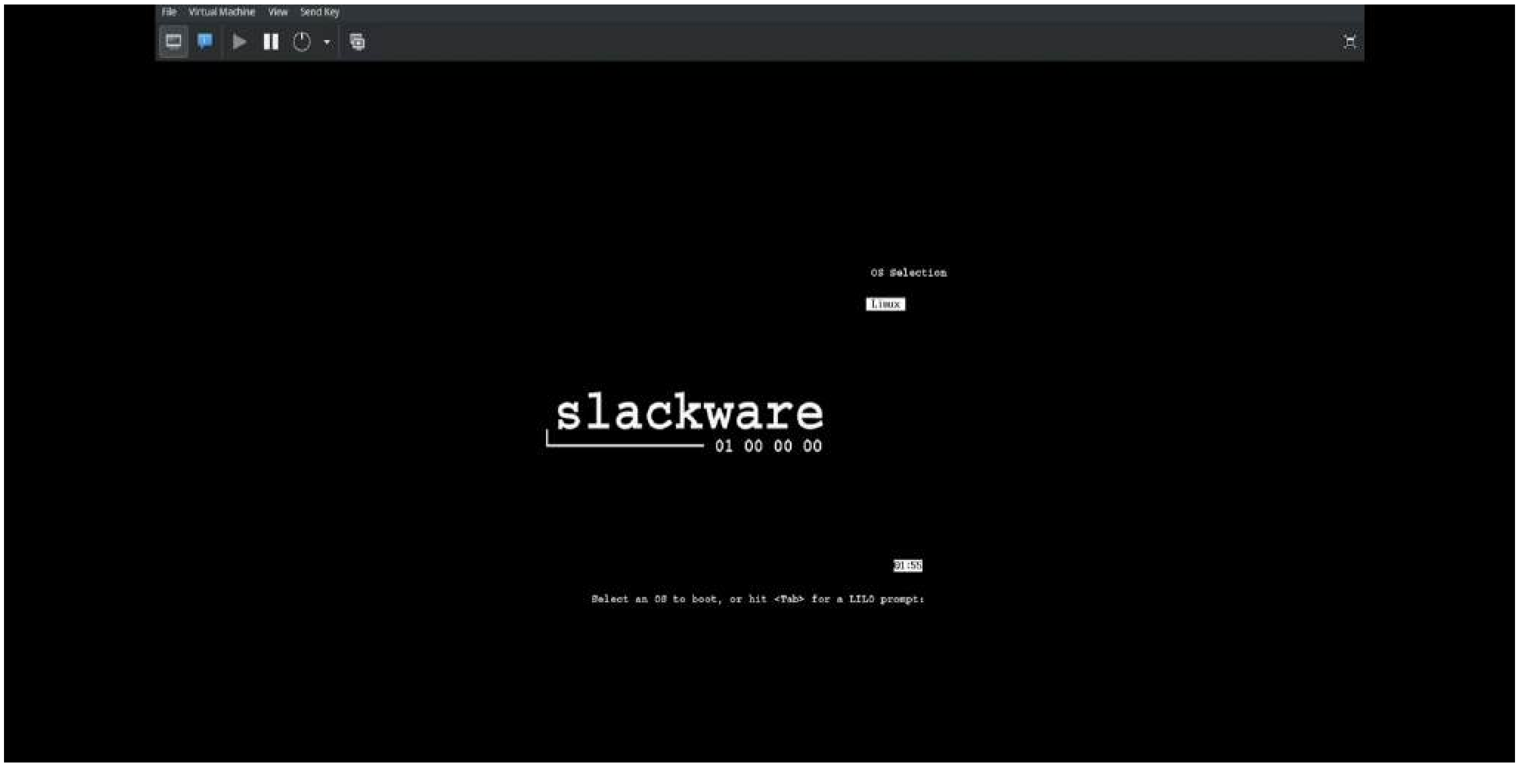
```
Slackware Linux Setup (version 15.0)
Welcome to Slackware Linux Setup.
Select an option below using the UP/DOWN keys and SPACE or ENTER.
Alternate keys may also be used: '+', '-', and TAB.

HELP        Read the Slackware Setup HELP file
KEYMAP      Remap your keyboard if you're not using a US one
ADDSWAP     Set up your swap partition(s)
TARGET      Set up your target partitions
SOURCE      Select source media
SELECT      Select categories of software to install
INSTALL     Install selected software
CONFIGURE   Reconfigure your Linux system
EXIT        Exit Slackware Linux Setup

        <  OK  >            <Cancel>
```

Step 15. reboot the system.



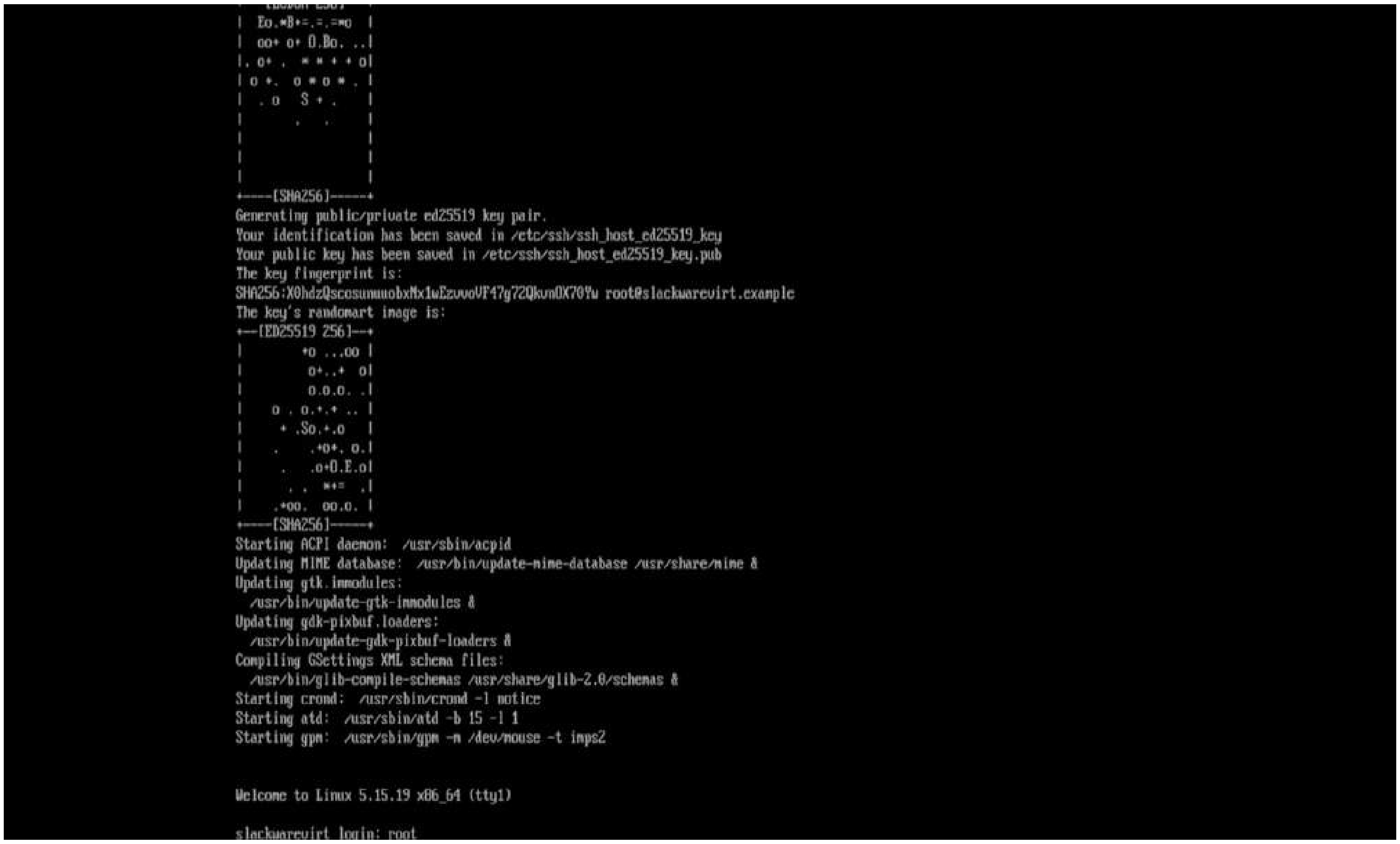
```
Slackware Linux Setup is complete
Slackware Linux setup is complete.

You may now reboot your machine into the Slackware OS.

If you have some post-installation tweaks you'd like to make manually, you
can open a shell.

If you want to make some changes to the hardware before booting the
Slackware OS, you may wish to power off computer.

                    Shell
                    Reboot
                    Power Off

        <  OK  >            <Cancel>
```

After rebooting it is going to...



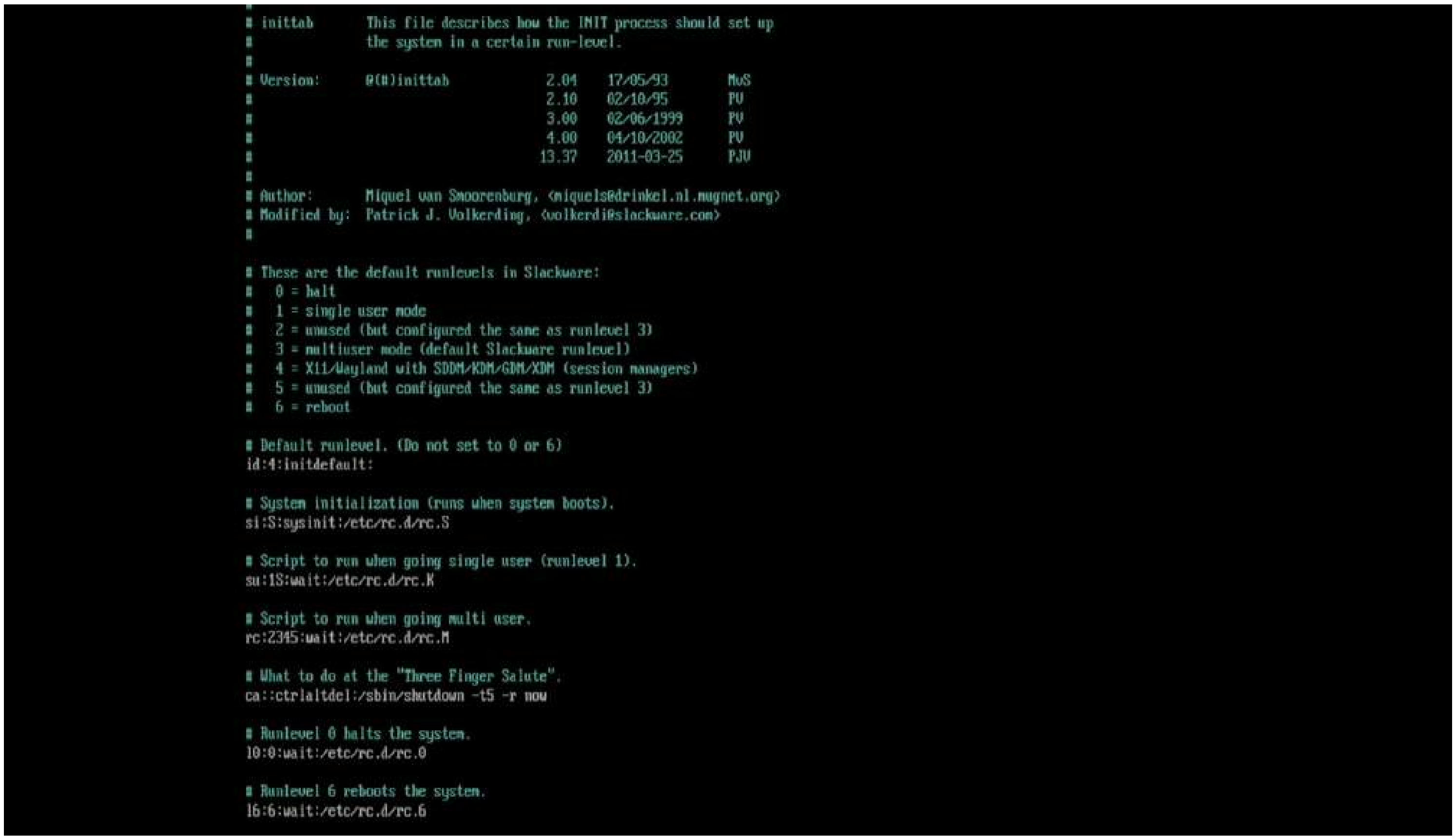
```
OS Selection
Linux

slackware
01 00 00 00

01:55

Select an OS to boot, or hit <Tab> for a LILO prompt:
```

Step 16. After reboot, login as root and create a user account.



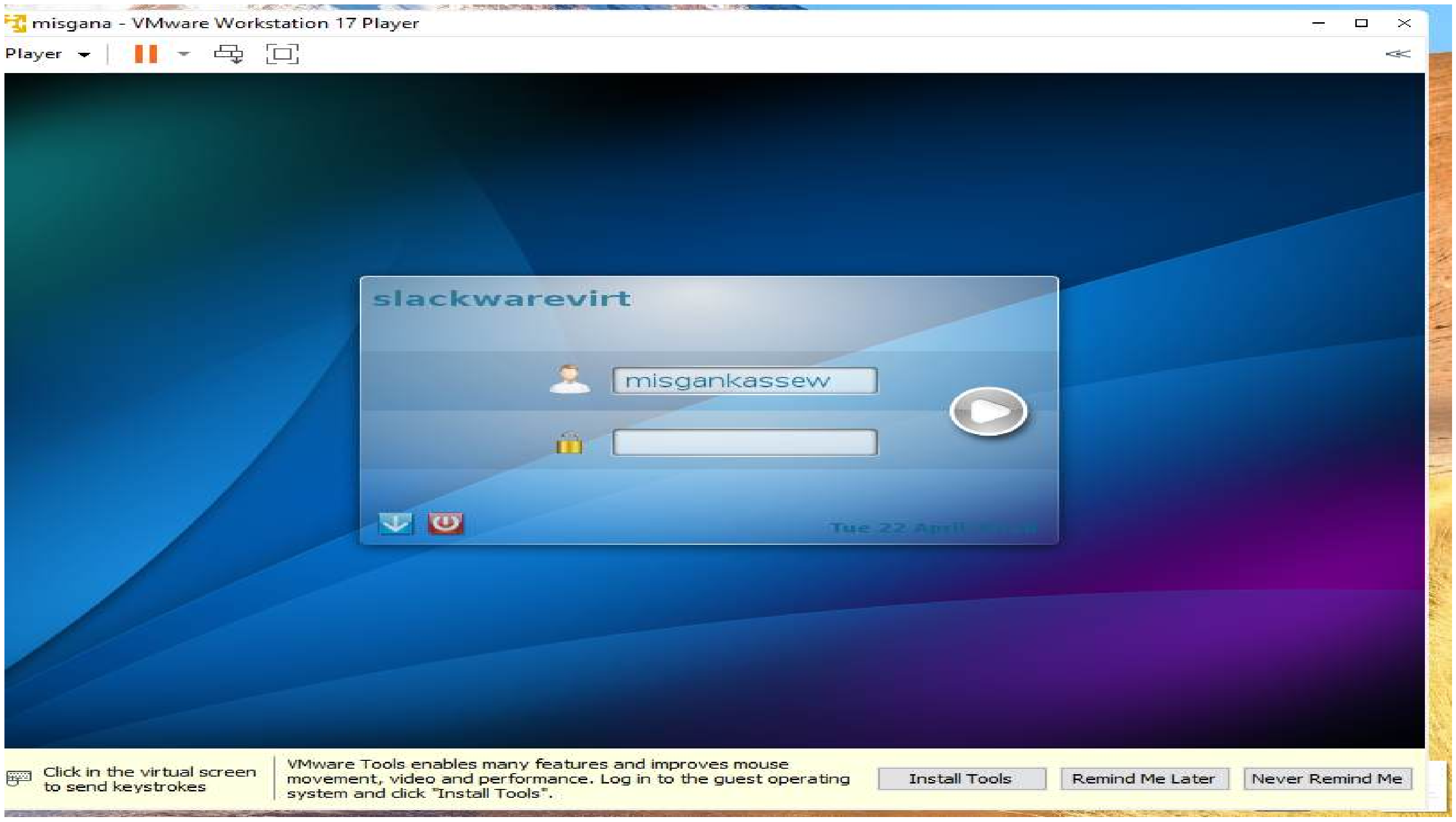
```
|  Eo.*B+=.=.=*o  |
|  oo+ o+ O.Bo. ..|
|. o+ .  * * + + o|
| o +.  o * o * . |
|  . o   S + .    |
|      .   .      |
|                 |
|                 |
|                 |
+----[SHA256]-----+
Generating public/private ed25519 key pair.
Your identification has been saved in /etc/ssh/ssh_host_ed25519_key
Your public key has been saved in /etc/ssh/ssh_host_ed25519_key.pub
The key fingerprint is:
SHA256:X0hdzQscosunuuobxMx1wEzvvoVF47g72QkunOX70Yw root@slackwarevirt.example
The key's randomart image is:
+--[ED25519 256]--+
|        +o ...oo |
|         o+..+  o|
|         o.o.o. .|
|    o . o.+.+ .. |
|     + .So.+.o   |
|    .    .+o+. o.|
|     .   .o+O.E.o|
|      . .  *+=  .|
|    .+oo.  oo.o. |
+----[SHA256]-----+
Starting ACPI daemon:  /usr/sbin/acpid
Updating MIME database:  /usr/bin/update-mime-database /usr/share/mime &
Updating gtk.immodules:
  /usr/bin/update-gtk-immodules &
Updating gdk-pixbuf.loaders:
  /usr/bin/update-gdk-pixbuf-loaders &
Compiling GSettings XML schema files:
  /usr/bin/glib-compile-schemas /usr/share/glib-2.0/schemas &
Starting crond:  /usr/sbin/crond -l notice
Starting atd:  /usr/sbin/atd -b 15 -l 1
Starting gpm:  /usr/sbin/gpm -m /dev/mouse -t imps2


Welcome to Linux 5.15.19 x86_64 (tty1)

slackwarevirt login: root
```

Step 17. Edit /etc/inittab → change id:3:initdefault: to id:4:initdefault: → reboot to see graphical login with user account.



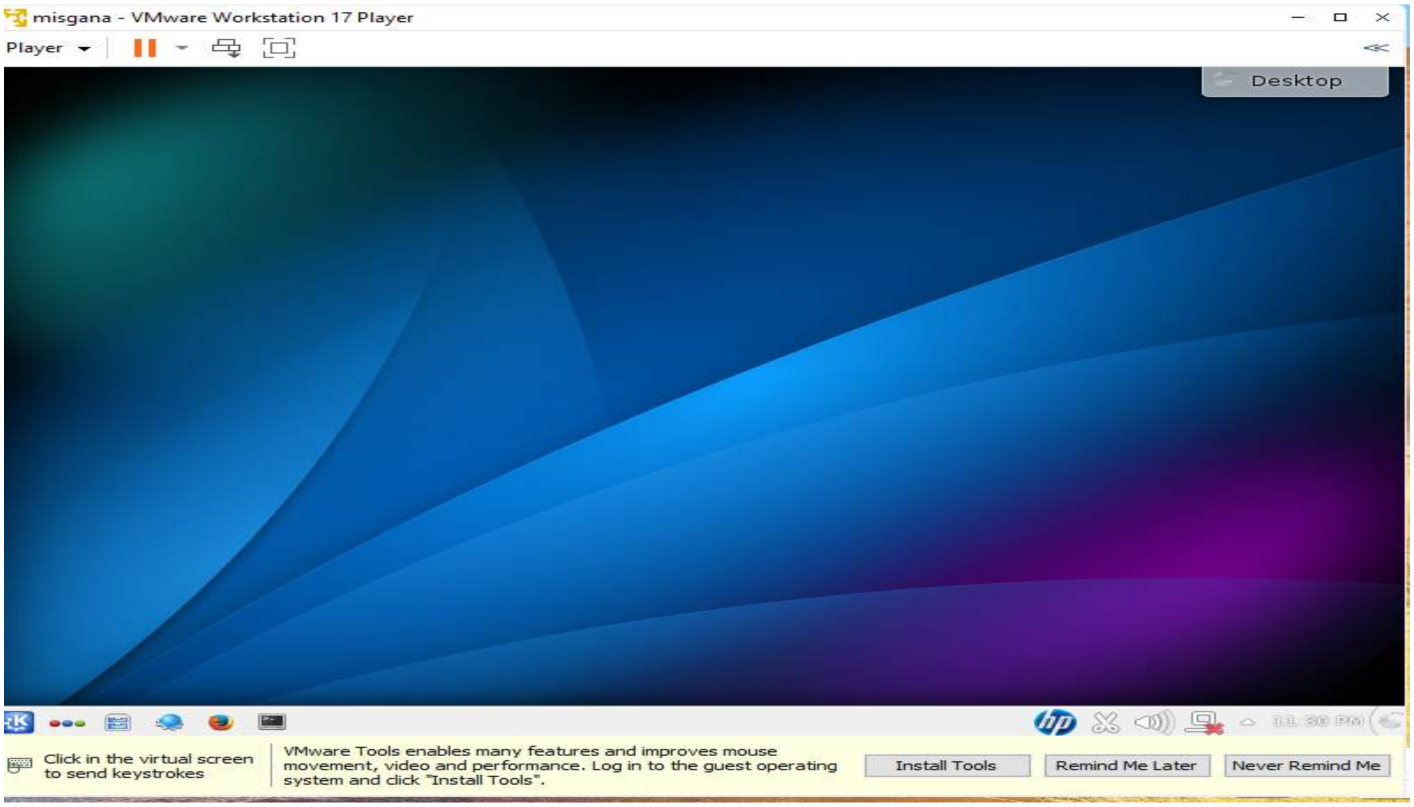
```
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Version:      @(#)inittab          2.04    17/05/93      MvS
#                                    2.10    02/10/95      PV
#                                    3.00    02/06/1999    PV
#                                    4.00    04/10/2002    PV
#                                   13.37    2011-03-25    PJV
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
# Modified by:  Patrick J. Volkerding, <volkerdi@slackware.com>
#

# These are the default runlevels in Slackware:
#   0 = halt
#   1 = single user mode
#   2 = unused (but configured the same as runlevel 3)
#   3 = multiuser mode (default Slackware runlevel)
#   4 = X11/Wayland with SDDM/KDM/GDM/XDM (session managers)
#   5 = unused (but configured the same as runlevel 3)
#   6 = reboot

# Default runlevel. (Do not set to 0 or 6)
id:4:initdefault:

# System initialization (runs when system boots).
si:S:sysinit:/etc/rc.d/rc.S

# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K

# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M

# What to do at the "Three Finger Salute".
ca::ctrlaltdel:/sbin/shutdown -t5 -r now

# Runlevel 0 halts the system.
l0:0:wait:/etc/rc.d/rc.0

# Runlevel 6 reboots the system.
l6:6:wait:/etc/rc.d/rc.6
```

Step 18.finally we get the login screen.



VMware Tools enables many features and improves mouse movement, video and performance. Log in to the guest operating system and click "Install Tools".

And then we get the interface with konsole terminal for implementation of system call, kickoff application e.t.c icons


misgana - VMware Workstation 17 Player
Player
Desktop
Click in the virtual screen to send keystrokes
VMware Tools enables many features and improves mouse movement, video and performance. Log in to the guest operating system and click "Install Tools".
Install Tools
Remind Me Later
Never Remind Me

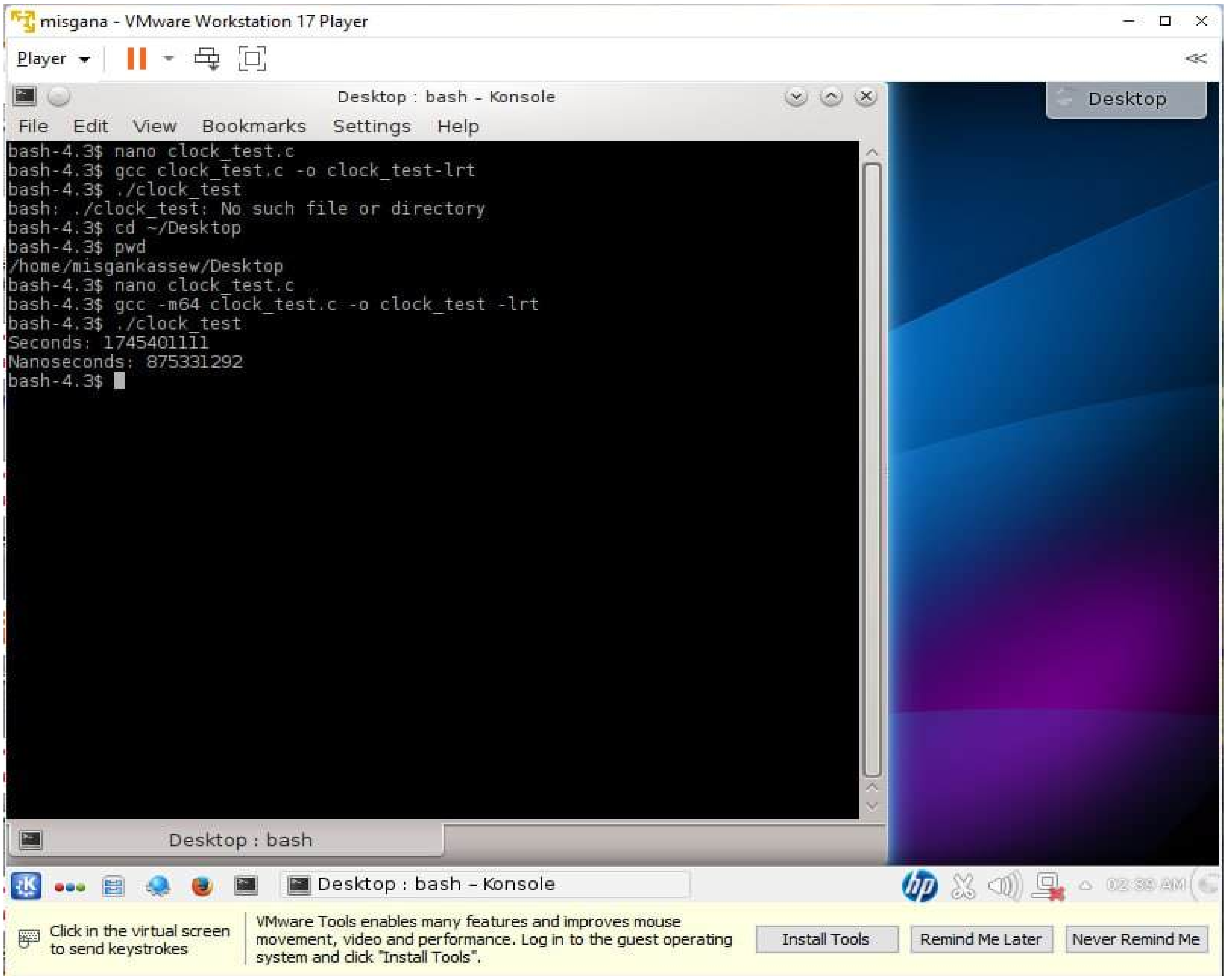
Issues (Problems Faced) and their solutions

✓ During I implement a system call I was challenged by the text shown in the image(no such file or directory) and that is appeared during I run it.

✓ After that I solve it step by step by identifying :

1.by writing pwd frist I check the location that was in /home/misganakassew/Desktop and it was correct.

2. And then I identify that what bit of operating system I use, and I get this that I use 64-bit operating system instead of 32-bit operating system so I have to write a command like this gcc -m64 clock_test.c -o clock_test -lrt instead of gcc clock_test.c -o clock_test -lrt. And finally I run it successfully by writing./clock _test command as shown in the image .



```
bash-4.3$ nano clock_test.c
bash-4.3$ gcc clock_test.c -o clock_test-lrt
bash-4.3$ ./clock_test
bash: ./clock_test: No such file or directory
bash-4.3$ cd ~/Desktop
bash-4.3$ pwd
/home/misgankassew/Desktop
bash-4.3$ nano clock_test.c
bash-4.3$ gcc -m64 clock_test.c -o clock_test -lrt
bash-4.3$ ./clock_test
Seconds: 1745401111
Nanoseconds: 875331292
bash-4.3$ 
```

Click in the virtual screen to send keystrokes

VMware Tools enables many features and improves mouse movement, video and performance. Log in to the guest operating system and click "Install Tools".

✓ an other problem was internet challenging and I solve this problem by using data cable.

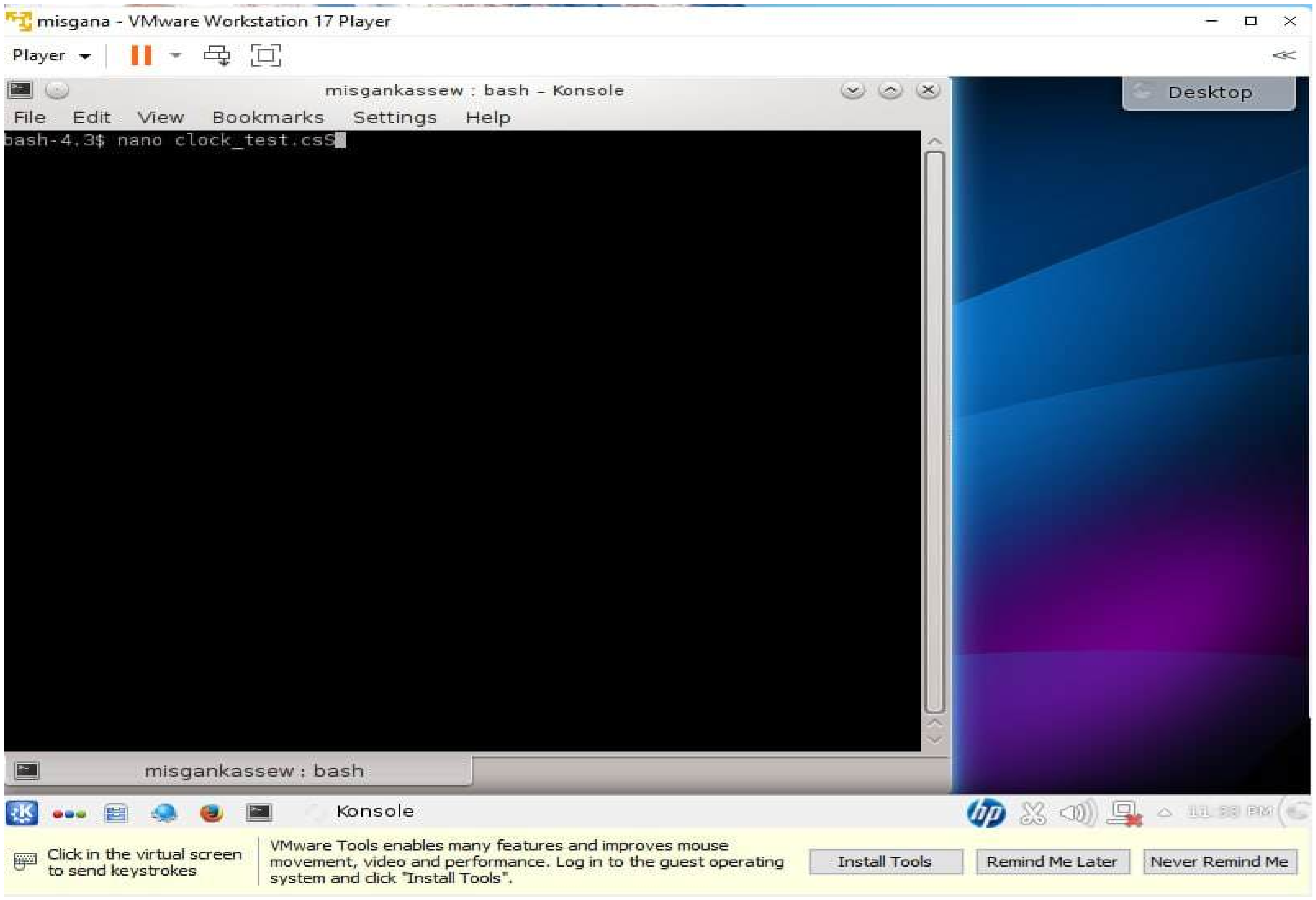
✓ there was also delaying and stacking of the computer and I try to solve it by refreshing timely.

System Call Implementation

Implement and test the clock_gettime() system call to retrieve the current system time (seconds and nanoseconds).

Steps that we followed

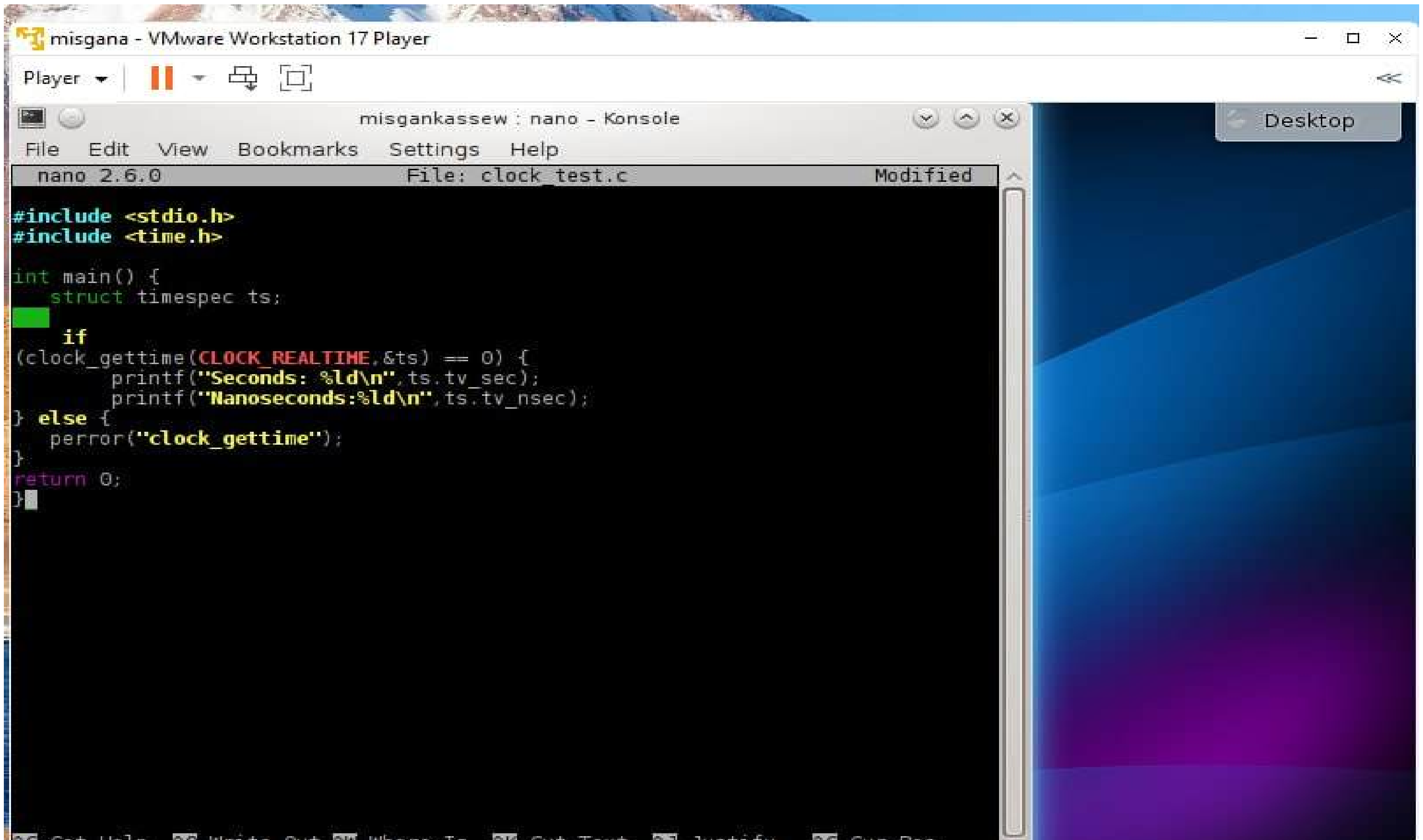
Step 1. Opened Terminal and create a source file/ a C file named clock_test.c and then press enter.



```
bash-4.3$ nano clock_test.csS
```

misgana - VMware Workstation 17 Player
Player
misgankassew : bash - Konsole
File Edit View Bookmarks Settings Help
Desktop
misgankassew : bash
Konsole
Click in the virtual screen to send keystrokes
VMware Tools enables many features and improves mouse movement, video and performance. Log in to the guest operating system and click "Install Tools".
Install Tools
Remind Me Later
Never Remind Me

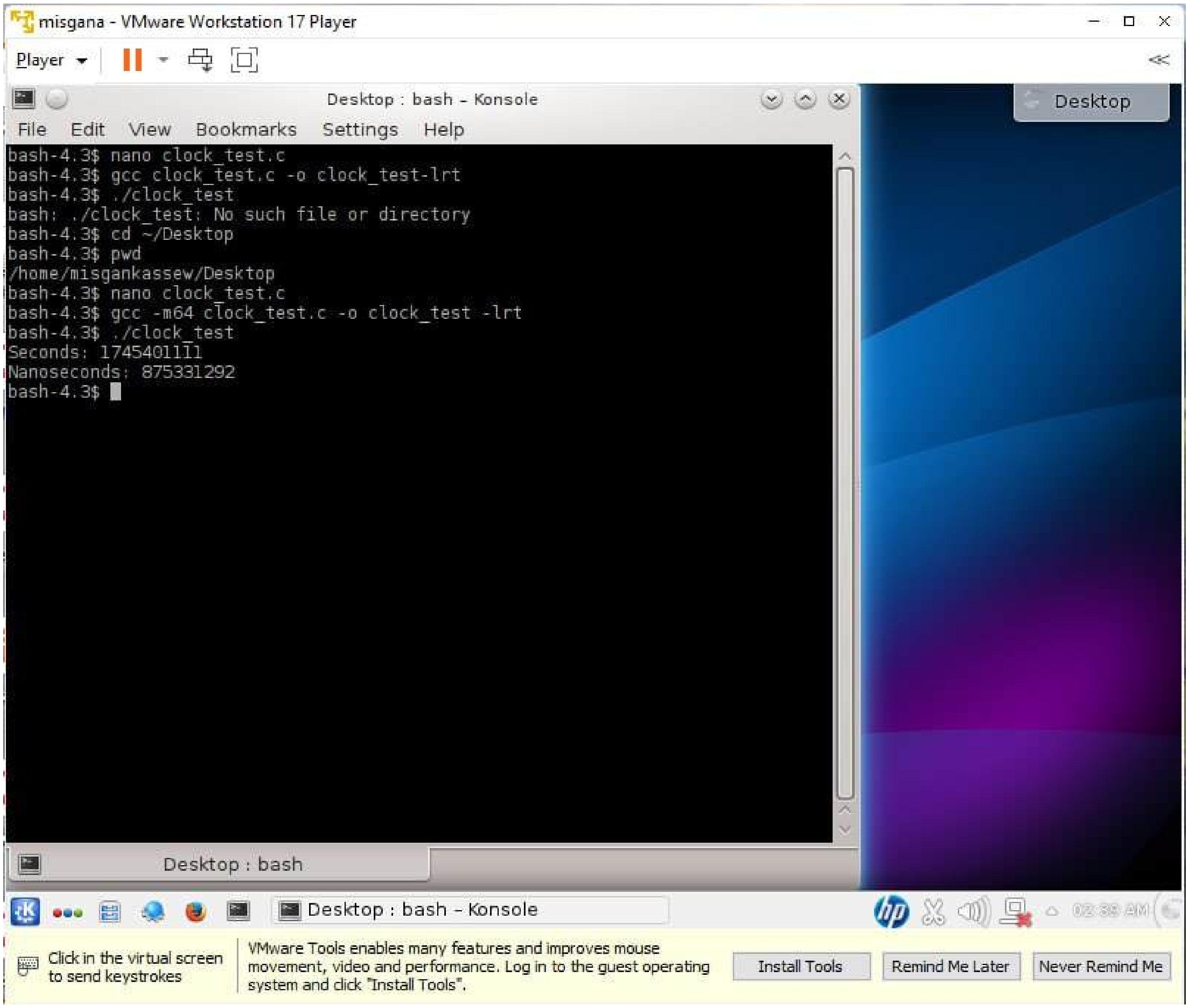
Step 2.write a code to implement a given system call and press ctrl + O to save and then press enter and then press Ctrl +X to exit.



```
 nano 2.6.0                   File: clock_test.c                    Modified

#include <stdio.h>
#include <time.h>

int main() {
   struct timespec ts;

    if
(clock_gettime(CLOCK_REALTIME,&ts) == 0) {
        printf("Seconds: %ld\n",ts.tv_sec);
        printf("Nanoseconds:%ld\n",ts.tv_nsec);
} else {
   perror("clock_gettime");
}
return 0;
}
```

misgana - VMware Workstation 17 Player
Player
misgankassew : nano - Konsole
File Edit View Bookmarks Settings Help
Desktop

Step 3. Finally compile and run the program and see the results of second and nanosecond.


misgana - VMware Workstation 17 Player
Player
Desktop : bash – Konsole
File Edit View Bookmarks Settings Help
Desktop
Desktop : bash
Desktop : bash – Konsole
Click in the virtual screen to send keystrokes
Install Tools
Remind Me Later
Never Remind Me

```
bash-4.3$ nano clock_test.c
bash-4.3$ gcc clock_test.c -o clock_test-lrt
bash-4.3$ ./clock_test
bash: ./clock_test: No such file or directory
bash-4.3$ cd ~/Desktop
bash-4.3$ pwd
/home/misgankassew/Desktop
bash-4.3$ nano clock_test.c
bash-4.3$ gcc -m64 clock_test.c -o clock_test -lrt
bash-4.3$ ./clock_test
Seconds: 1745401111
Nanoseconds: 875331292
bash-4.3$
```

VMware Tools enables many features and improves mouse movement, video and performance. Log in to the guest operating system and click "Install Tools".

Further on System Call

A system call is a programmed request from a user-space application to the operating system's kernel to perform a privileged operation. Since user applications are restricted from directly accessing hardware and critical system resources, they rely on system calls to interact with these resources securely. Examples include file manipulation, process control, memory management, and time handling.

System calls serve as the interface between user programs and the kernel, allowing the user to request services like reading from a file, allocating memory, or retrieving the system time.

The clock_gettime() System Call

The clock_gettime() system call is used to retrieve the current time of a specified clock, most commonly the real-time clock. It returns the time in two components:

Seconds: The whole seconds part of the current time.

Nanoseconds: The fractional part of a second, offering high precision.

This system call is essential in systems that require precise timing, such as performance profilers, schedulers, or real-time applications.

The Purpose of Implementing clock_gettime() system call

The purpose of implementing the clock_gettime() system call is to:

-Understand how time is managed in the Linux kernel.

-Gain practical experience in kernel development.

-Enable applications to access accurate and high-resolution system time.

-This implementation also strengthens the understanding of how user-space and kernel-space communicate.

Filesystem Support

A filesystem is the method and data structure an operating system uses to manage files on a disk or partition. It determines how data is stored, organized, retrieved, and secured. Different filesystems have different strengths in terms of speed, reliability, fault tolerance, and support for large files or disks.

In Linux, several filesystems are supported natively or via modules, each offering unique features. Choosing the right one depends on the intended use of the system, hardware specifications, performance needs, and reliability requirements.

Filesystem Used in This Installation: ext4

For this Slackware 14.2 installation, the chosen and configured filesystem is ext4 (Fourth Extended Filesystem). Ext4 is widely used as the default filesystem in most modern Linux distributions, including Ubuntu, Debian, Fedora, and Arch. Slackware also supports ext4 fully, and it's ideal for both desktop and server environments.

Why ext4 Was Chosen?

Stability: Ext4 is mature and well-tested in production environments.

Journaling: It provides a journaling feature that helps protect against data corruption during unexpected shutdowns or crashes.

Performance: Supports faster file access through techniques like extents and delayed allocation.

Large Volume and File Size Support: Ext4 supports individual file sizes up to 16 TiB and volumes up to 1 EiB.

Backward Compatibility: Ext4 can mount ext2 and ext3 filesystems, making migration easy.

Wide Tool Support: Almost all Linux recovery, backup, and repair tools support ext4.

Advantage and Disadvantage

Advantages of Slackware Linux in VMware:

Stability and Reliability: Slackware is famous for its rock-solid performance, making it ideal for long-term use and learning environments.

Manual Control: Users learn deeply by manually managing partitions, services, bootloaders, and configurations.

Minimalism: Slackware avoids unnecessary software bloat, leaving only what the user explicitly installs and configures.

Performance: Because it runs fewer background services by default, Slackware can be lightweight and fast, even on older hardware.

Education-Focused: Excellent for learning shell commands, scripting, system boot procedures, package management, and networking.

Disadvantage of Slackware Linux OS

Steep Learning Curve: New users may find it difficult due to the lack of GUI installation and dependency resolution tools.

Limited Automation: Tasks like system updates and package installation often require manual input and careful handling.

Software Compatibility: Some modern applications may require additional effort to install or configure due to Slackware’s strict adherence to simplicity.

Documentation Dependency: A lot of reliance on user manuals, official documentation, and community forums is required to solve problems.

Future Outlook / Recommendation

After successful installation, try installing a graphical desktop environment like XFCE, KDE, or Fluxbox to improve usability.

Learn to manage packages using pkgtool, slackpkg, or sbopkg, which provide structured methods for software management.

Explore the process of compiling and installing a custom Linux kernel to tailor the OS to specific

needs.

Practice writing shell scripts for automation of system tasks and experiment with cron jobs and system logging.

For advanced learning, integrate Slackware into a virtual network with other OSs in VMware to simulate server-client or multi-OS environments.

Use Slackware as a base for practicing system programming, performance analysis, or server setup (e.g., hosting a basic Apache server).

Virtualization in Modern Operating Systems

What is Virtualization?

Virtualization is a technology that allows multiple operating systems or applications to run simultaneously on a single physical machine by abstracting the hardware and providing virtual environments. It is the process of creating a virtual version of something—like a virtual machine (VM), storage device, or network resources—using software.

In the context of operating systems, virtualization allows a host machine to run several guest operating systems using tools like VMware Workstation, Oracle VirtualBox, or Hyper-V. These guest systems behave as if they are running on dedicated hardware, but in reality, they share the physical resources of the host.

Why Virtualization?

Virtualization plays a significant role in modern computing for the following reasons:

✓ It enables multiple virtual machines to run on a single physical server, maximizing CPU, memory, and storage usage.

✓ Each VM operates in its own isolated environment, reducing the risk of system crashes or

security breaches affecting the host or other VMs.

✓ Fewer physical servers are needed, which reduces hardware costs, energy consumption, and maintenance.

✓ Developers and students can test different operating systems, applications, and configurations without impacting their main systems.

✓ VMs can be paused, saved, copied, and moved between machines easily.

✓ Virtual machines can be backed up and restored quickly, improving business continuity and disaster recovery efforts.

How Virtualization Works?

Modern virtualization relies on hypervisors—software that creates and manages virtual machines by simulating hardware environments.

a) Types of Hypervisors

✓ Type 1 (Bare-metal)- Runs directly on physical hardware. Examples: VMware ESXi, Microsoft Hyper-V, Xen.

✓ Type 2 (Hosted)- Runs on a host operating system. Examples: VMware Workstation, Oracle VirtualBox.

b) VM Architecture

Each virtual machine contains:

✓ A guest operating system (e.g., Linux, Windows).

✓ Virtual hardware components (CPU, RAM, hard disk, network interface).

✓ Virtual BIOS/UEFI, bootloaders, and sometimes virtual drivers.

c) Virtualization Layers

✓ The host OS manages physical resources.

✓ The hypervisor intercepts hardware access requests from VMs and maps them to the host.

✓ The guest OS operates unaware that it is running in a virtual environment.

d) VMware Workstation Example

When Slackware Linux is installed inside VMware Workstation:

✓ VMware creates a virtual hard disk and simulates other hardware.

✓ Slackware runs on this virtual hardware as if it were physical.

✓ The user can interact with Slackware, install software, and even simulate networks—all within a virtual environment.

Conclusion

Through the installation of Slackware Linux on VMware Workstation, the concept of manual OS setup and configuration was explored in depth, reinforcing critical system administration skills. Working in a virtualized environment allowed for flexible testing and practical engagement without risk to the host system.

The study of virtualization provided insight into how operating systems are efficiently deployed and managed in modern infrastructure, from desktops to data centers. It also highlighted the role of tools like VMware in enabling parallel environments for learning and development.

Implementing the clock_gettime() system call gave hands-on experience with kernel-level operations, showcasing the interaction between user-space applications and system resources. This exercise contributed to a deeper understanding of timekeeping, process control, and the role of system APIs in programming.

Altogether, these three activities fostered both technical competence and conceptual clarity, forming a strong base for advanced learning in system programming, Linux administration, and virtual infrastructure management.