SYSTEM RESCUE A STEP BY STEP GUIDE TO GETTING STARTED

Learn how to administrate and rescue computers using SystemRescue and free software

20220205

Table of Contents

Preface

SystemRescue is a Linux-based toolkit for computer administration and recovery. It is available for free. It runs from removable devices such as USB memory sticks, so it can be used on computers on which the operating system has stopped working. It allows users to perform many essential operations, such as recovering data files, partitioning disks, backing up and restoring system disks, resetting system passwords and testing the memory. This book is an introduction to SystemRescue. It helps new users get started with SystemRescue, and it provides step-by-step guidance for performing common operations.

What this book covers

Chapters 1 to 5 go through the basics. We start with an overview of SystemRescue, followed by guidance for downloading and installing it on a removable boot device. We continue with instructions to start the system using the most relevant boot parameters, navigate through the system and connect to the network.

Chapters 6 to 9 give you the essential knowledge and commands required to access and administrate disks from SystemRescue. You learn how disks are structured, how to identify file systems and access their contents and which disk utilities are available.

Chapters 10 to 14 provide guidance for performing common operations based on the background in the previous chapters. These guides show you how to recover data files from a computer on which the operating system is broken, how to clone disks, how to backup and restore disk contents using image files, how to use chroot to fix Linux systems and how to reset a forgotten local Windows password.

It is highly recommended that you read the first nine chapters in the normal order, unless you already have the skills, as each chapter provides essential information needed to understand what follows.

Chapters 10 to 14 can be read in any order, depending on your needs and interests. Ideally, you should read them all to have a good understanding of what SystemRescue can help you do. If you are in a hurry, you can skip some of these chapters and focus on what is most relevant to you. These later chapters make constant use of the knowledge from the first part of the book. Hence you are invited to refer to earlier chapters when necessary.

The technical details provided in this book are accurate at the time of publication. Please keep in mind that software changes quickly, so some details in this book might be different from what you find on your computer. What is most important is that you understand the general principles. These should remain valid for the long term.

Who should read this book

This book is targeted at users who are new to SystemRescue and want to perform computer maintenance operations using free software. You need good computer skills, and you should not be afraid to use command-line interfaces. Familiarity with Linux commands helps, but it is not required to use this book, as new commands are introduced with examples.

Use utilities with caution

It is important to understand well how the utilities mentioned in this book should be used, as they can cause damage if they are misused. Pay attention to details, as small mistakes can have a big impact. Most commands in the examples refer to resources such as disk names which are specific to a particular situation. So, make sure you adapt the commands in the examples to your situation before you execute them. To minimize risks, it is highly recommended that you regularly back up all essential data, especially before you perform sensitive operations.

Disclaimer

Licence

Project website

SystemRescue can be found on the official project website: https://www.system-rescue.org.

Chapter 1: Introduction to SystemRescue

Project objective

SystemRescue, formerly known as SystemRescueCd, is a Linux distribution intended to be used for administrating or recovering computer systems and data. It is often run from removable media such as USB memory sticks so it can be used on computers where the operating system has stopped working. It comes with many programs to carry out system administration tasks. It is based on Linux, but it also operates on Windows computers as it supports all mainstream file systems. SystemRescue is completely free to use.

Here are some examples of what you can do with SystemRescue:

- You might need to repartition disks when you purchase a new disk, reinstall the operating system, or just want to modify how the disk space is allocated.
- You might have to recover data files stored on a local disk if the operating system has stopped working. Important data files should be transferred to an external disk or another computer over a network before the disk is erased and the system reinstalled.
- Creating a physical copy of an internal disk allows you to restore all its contents in a single operation if the computer is affected by a critical software issue or disk failure. Having such a physical copy is very useful if your operating system stops working after a faulty system update, because of malware such as a virus, or if the primary disk must be replaced because of a hardware failure.
- Resetting the password of a local user account on either a

Linux or a Windows system allows you to restore access to the system if you forget your password.

- You can use SystemRescue to fix system or configuration changes which are keeping Linux from starting.
- You can run a memory test when you experience random problems or crashes and you want to know if this is caused by damaged memory modules.

SystemRescue is based on Linux, so it can naturally be used to administrate Linux systems. But it also works with Windows computers, as it can access data stored on FAT and NTFS file systems. It supports a wide range of hardware devices and file systems. This is essential as you can use the system and access your data only if your hardware is properly supported. Also, the Linux ecosystem provides lots of free utilities. SystemRescue gives you access to many such administration and recovery programs. It also comes with general applications such as text editors and web browsers. However, software which is not related to system administration such as word processors and music players is not included in SystemRescue so it stays as compact as possible.

The SystemRescue way

SystemRescue comes with a lot of tools so you can administrate computer systems and recover data files. But it will not resolve problems automatically for you. To use it effectively, you must know which programs to use and and how to use them. There are many ways to learn how to use these programs. This book helps you get started with SystemRescue, but you can do much more than what is documented here. Most programs included in SystemRescue come with offline reference documentation, which can be accessed from the system without having to be connected to the internet. You can also find good tutorials on the internet to learn how to use a specific program to solve a particular issue.

The large choice of programs allows you to perform some operations in multiple ways so you can decide which approach is best for you. Some utilities are controlled through graphical interfaces but most are based on command lines. In general, graphical tools are easier to use, but command-line programs give you more control and flexibility. For example, you can take advantage of Linux shell pipelines to combine multiple commands to perform operations more effectively. You can even automate tasks which you must execute regularly by writing shell scripts or using Perl, Python or Ruby, which are all included.

If you have advanced needs, it is possible to boot SystemRescue from the network instead of using removable devices. This is particularly useful when you have to manage multiple computers, as you can deploy SystemRescue on a central server. This way, you can update SystemRescue more easily, and you can manage the configuration and possible customisations in one place.

Upstream distribution

Since version 6.0, SystemRescue is based on Arch Linux, which is a popular Linux distribution. Hence most components included in SystemRescue are provided exactly as packaged by Arch Linux. This upstream project is developed by a large number of users. That community builds all components from source code, applies patches to fix bugs, keeps the software up to date, and makes sure all packages are consistent and work well together. The Arch Linux project also provides very good documentation, so it is a good place to find detailed information about specific programs in SystemRescue.

The official Arch Linux project is currently available only for the 64-bit PC architecture (amd64/x86_64). It used to be provided for 32-bit PCs (i686) as well, but the popularity of that architecture has decreased so support was discontinued in 2017. Arch Linux 32 is a derivative project which continues to distribute 32-bit packages.

This allows a 32-bit version of SystemRescue to be provided. Arch Linux 32 is not as active as the main project, hence 32-bit versions of packages are often released days or weeks after the 64-bit versions. It is important to keep in mind that the number of people who use 32-bit hardware and software keeps decreasing. Hence fewer users are reporting issues that specifically affect 32-bit programs, and fewer developers are interested in maintaining those programs. As 64-bit hardware has been available for a very long time, it is strongly recommended that you use the 64-bit edition of SystemRescue if you can.

Chapter 2: Preparing the Boot Device

Downloading the ISO image

The first thing to do is to download SystemRescue. To get a genuine copy, make sure you download it from the official website: https://www.system-rescue.org/.

The main file to download is the ISO image file. This one file contains everything you need to use SystemRescue. The ISO image is quite large, so you will need a decent internet connection. It is strongly recommended that you download the 64-bit edition (amd64/x86_64) unless your hardware supports only 32-bit software.

You should get the latest version available unless there is a particular reason to use an older version. The instructions below are based on version 7.00, but the process should be very similar in later versions.

Verifying the integrity of the ISO image

The download page also provides checksums so you can verify the integrity of the ISO image file you downloaded. Each checksum file is named after a specific version of the ISO image. These checksums are sequences of numbers and letters derived from the contents of the original ISO image file. Any changes in the file would cause a checksum to be completely different. Hence you can make sure your download is strictly identical to the original ISO image by calculating the checksum of your copy and comparing it to the checksum provided on the website.

For example, here is the SHA256 checksum that corresponds to version 7.00 for the amd64 architecture. You can find it in a file named `systemrescue-7.00-amd64.iso.sha256`:

```
ab7fc8a073f09ad7b093ba42bf7a49681954e5567df182766a0343aeeba1251c
```

Verifying a checksum does not take long, and it gives you a chance to immediately fix the problem if you get an invalid copy of the ISO image. This will help you avoid unexpected issues when you use SystemRescue. The website provides multiple types of checksums. They are all effective and widely used. You need just one checksum to verify the integrity of the download. Make sure the checksum file corresponds to the correct version and architecture by comparing the names of the files.

To calculate a SHA256 checksum on Linux, use the `sha256sum` command followed by the path to your copy of the ISO image in a terminal:

```
[user@localhost]$ sha256sum systemrescue-7.00-amd64.iso
ab7fc8a073f09ad7b093ba42bf7a49681954e5567df182766a0343aeeba1251c  systemrescue-
7.00-amd64.iso
```

On Windows you can launch a command prompt from the menu, go to the download folder using `cd`, and then use the `certutil` command as in the following example:

```
C:\> cd \temp
C:\temp> certutil -hashfile systemrescue-7.00-amd64.iso sha256
SHA256 hash of systemrescue-7.00-amd64.iso:
ab7fc8a073f09ad7b093ba42bf7a49681954e5567df182766a0343aeeba1251c
CertUtil: -hashfile command completed successfully.
```

After you calculate the checksum of your copy of the ISO image, you should compare it with the contents of the checksum file that you downloaded. Something has gone wrong if the checksum of your copy of the ISO image is different from the expected checksum. There could have been a corruption during the download, the file could have been downloaded partially, or it might have been written on a damaged disk. In these cases, you can try again or you can download the ISO image to a different computer so you get a valid file.

If you do not want to verify the checksum of the ISO image at this stage you can specify a particular boot option when you start SystemRescue. That way the checksum of the system image is verified during initialisation. This can be a good alternative, as it takes less effort but it verifies the integrity only of the main system file. This is documented in the next chapter.

Choosing the right boot device

The next step is to install SystemRescue on a bootable device. There are two popular options.

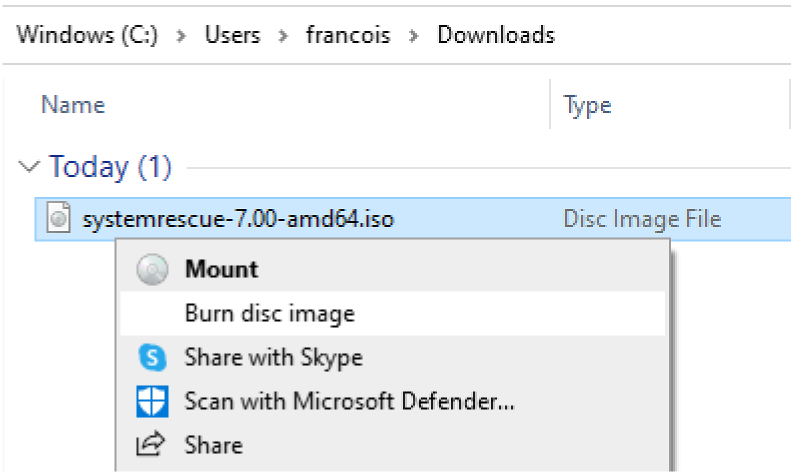
Originally SystemRescue was designed to run from CD-ROM media, but you can use a DVD-ROM because it has more capacity. Most computers that still have optical drives should be able to read DVD-ROMs. However, many newer computers, especially compact laptops, do not have optical drives. Therefore, memory sticks are the best type of removable devices to run SystemRescue.

All data on the target device will be deleted during the installation, so you must backup your data before you use that device for SystemRescue. You may also prefer to purchase a new device. Make sure the size of the boot device is bigger than the size of the ISO image. Ideally, it should be a good quality device to avoid problems. It should also be fast, so the system can start quickly. Hence USB 3 memory sticks are recommended over older models.

Preparing a bootable DVD-ROM

The first option is to install SystemRescue on a DVD-ROM. It is very important to burn the ISO file in the right way. Do not add the ISO file to a disc as an ordinary data file. The ISO file is an image of the whole device, so it must be written in such a particular way.

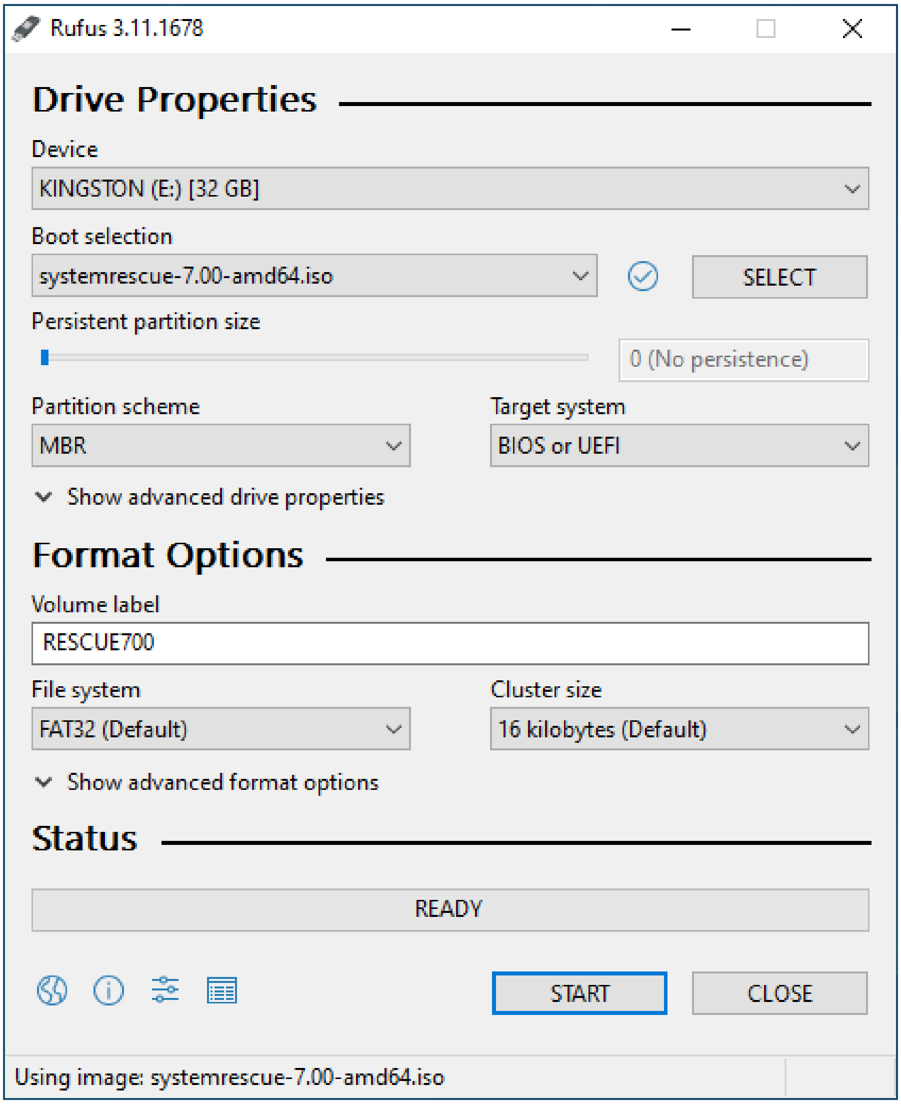
On Windows 10, this can be done easily by clicking on the ISO file (in Windows explorer) with the right button of the mouse. A contextual menu should show an entry to `Burn disc image`. This is the simplest solution, but you can use alternative programs if you prefer.


Windows (C:) › Users › francois › Downloads
Name
Type
Today (1)
systemrescue-7.00-amd64.iso
Disc Image File
Mount
Burn disc image
Share with Skype
Scan with Microsoft Defender...
Share

On Linux, ISO images can be written to DVDs using programs such as K3B, Brasero and XfBurn or by using command-line tools.

Preparing a memory stick on Windows

You need a specific program to install SystemRescue on a memory stick. Rufus is highly recommended for this task. You can download it for free from https://rufus.ie/.


Rufus 3.11.1678

Drive Properties

Device: KINGSTON (E:) [32 GB]

Boot selection: systemrescue-7.00-amd64.iso

Persistent partition size: 0 (No persistence)

Partition scheme: MBR

Target system: BIOS or UEFI

Format Options

Volume label: RESCUE700

File system: FAT32 (Default)

Cluster size: 16 kilobytes (Default)

Status

READY

Using image: systemrescue-7.00-amd64.iso

Once you download Rufus, execute it on your computer then insert the memory stick. At the top of the screen, a drop-down menu will list all removable devices on the computer. They should be easy to identify thanks to their volume names and sizes. Select the relevant memory stick. Next, select the ISO image that you downloaded. When you select these two items, Rufus refreshes the screen to show options that are relevant in the current situation.

It is recommended that you select `MBR` as the partition scheme and

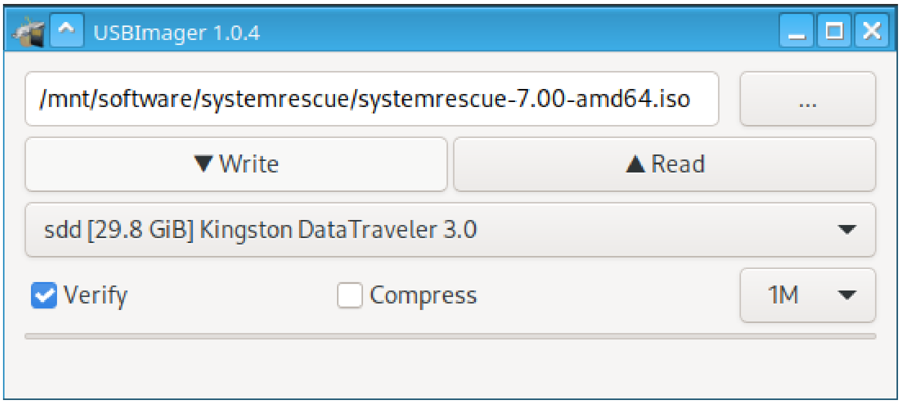
`BIOS or UEFI` as the target system so your memory stick can be used on any type of PC.

Next, look at the box where the volume name of the memory stick can be edited. Rufus automatically detects the name from the ISO image and reuses it by default. It is essential not to change the original volume name as it is used during the boot process to identify the device that contains SystemRescue files.

The device should be initialised with a `FAT32` file system and the default cluster size. Other options may be available depending on the version of Rufus. In general, default choices should be fine. Once you are ready, click on `Start`. Rufus will ask if the image should be written in `ISO mode` or `DD mode`. Both options will provide a usable boot device. It is recommended that you choose `ISO mode` so the memory stick is prepared with a writable file system using the full size of the device. This allows you to use the remaining space on the memory stick to store additional files, and it also allows SystemRescue files such as boot menu entries to be modified. In `DD mode`, the memory stick will be bootable but the remaining disk space will not be accessible for storing additional files.

Preparing a memory stick on Linux

The recommended utility for installing SystemRescue on a memory stick from Linux is `usbimager`. It is very compact, and it comes with minimal dependencies. It is available for free at https://gitlab.com/bztsrc/usbimager, and it can be executed without installation. Multiple download formats are available. The X11 version is the best choice for most users. Once you download the ZIP file, extract the files in it. Then execute the program from the disk.



/mnt/software/systemrescue/systemrescue-7.00-amd64.iso

sdd [29.8 GiB] Kingston DataTraveler 3.0

[✓] Verify

[ ] Compress

This program is very simple to use. Select the ISO image and the target memory stick, using the graphical interface, then click on the `Write` button to start the installation. This is like performing the installation using Rufus in DD mode, so the memory stick will be bootable. However, the additional disk capacity will not be available for storing additional files.

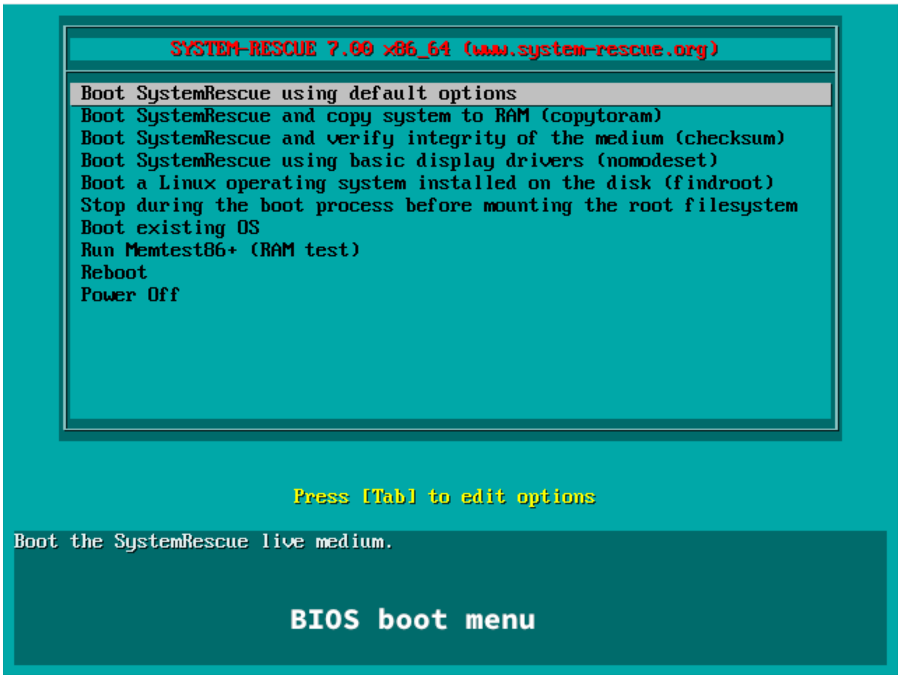
Chapter 3: Starting SystemRescue

Booting from the removable device

After you prepare the SystemRescue boot device, it is time to use it. You should make sure the computer starts from the boot device you prepared. You can do this by pressing the correct key when you start the computer. You can boot from a specific removable boot device temporarily, or you can modify the boot order in the firmware settings so the removable boot device you prepared is used before other boot devices. Ideally, the computer should always try to boot from a removable device before it looks for the internal disk, so you don't worry about the boot priority each time you want to boot from a removable device. A message should appear just after you start the computer to indicate which keys to press to make these changes. In general, it is either a function key such as `F1` or `F2`, or a key like `Delete` or `Enter`.

Boot menus

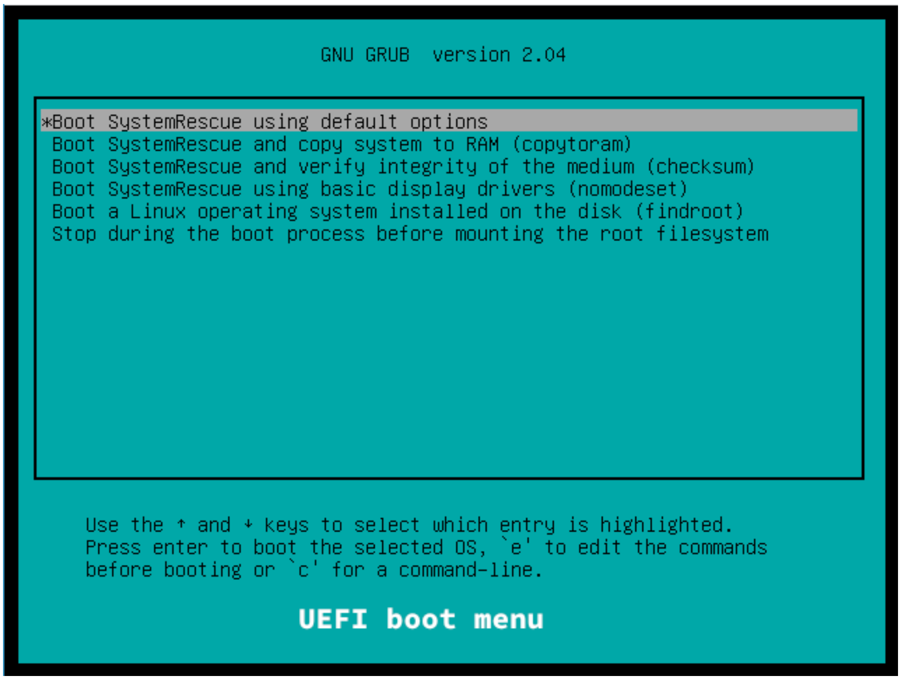
Once the system has started from the removable boot device, you should see the SystemRescue boot menu with a blue background. There are two types of boot menus. The BIOS menu appears if the computer is based on a legacy BIOS firmware or if it is using UEFI firmware configured to run in legacy mode:



```
SYSTEM-RESCUE 7.00 x86_64 (www.system-rescue.org)

Boot SystemRescue using default options
Boot SystemRescue and copy system to RAM (copytoram)
Boot SystemRescue and verify integrity of the medium (checksum)
Boot SystemRescue using basic display drivers (nomodeset)
Boot a Linux operating system installed on the disk (findroot)
Stop during the boot process before mounting the root filesystem
Boot existing OS
Run Memtest86+ (RAM test)
Reboot
Power Off

Press [Tab] to edit options

Boot the SystemRescue live medium.
```

BIOS boot menu

The UEFI boot menu appears if you are using a recent computer configured to boot in UEFI mode:



```
GNU GRUB  version 2.04

*Boot SystemRescue using default options
 Boot SystemRescue and copy system to RAM (copytoram)
 Boot SystemRescue and verify integrity of the medium (checksum)
 Boot SystemRescue using basic display drivers (nomodeset)
 Boot a Linux operating system installed on the disk (findroot)
 Stop during the boot process before mounting the root filesystem

   Use the ↑ and ↓ keys to select which entry is highlighted.
   Press enter to boot the selected OS, `e' to edit the commands
   before booting or `c' for a command-line.
```

UEFI boot menu

The two types of boot menus are similar but there are some differences. The first few entries are the most useful ones. They allow you to boot the main SystemRescue operating system in different ways. If you started in Legacy BIOS mode, you will also see an entry to run `Memtest86+`. This standalone program performs extended memory tests. It is particularly useful if you experience stability issues with your computer, such as crashes, and you want to know if this is caused by defective memory modules. Unfortunately, this program is not available in UEFI mode. The following sections provide information about the main boot entries.

Boot using the default options

The first choice in the menu causes SystemRescue to boot with reasonable options so it is most likely to work for everyone. As this is the default choice, it will be activated if you press `Enter` when it is highlighted or if you let the timeout expire. The configuration associated with this entry causes the system to run from the boot device you prepared. The system start-up will be quick, but the removable boot device will be required throughout the session. You cannot remove it until you shut down or reboot the computer.

Boot and copy the system to memory (copytoram)

A popular alternative is to boot and copy the system to memory (RAM). This causes the whole system to be copied from the boot device to the memory during the boot process. This takes additional time to start, and it requires more RAM. However, once it is fully initialised the system will run entirely from memory. It should be more reactive, and it will not rely on the boot device after the initialisation is complete. Therefore, the boot device can be removed when the system is ready, and it won’t be affected.

This way to boot the system requires more resources. You must have enough RAM to store a copy of the system and to execute applications. This should not be a problem unless the amount of memory available on the computer is less than twice the size of the ISO image.

Running the system from memory is useful if you work with multiple removable devices and you do not have sufficient drives or ports to insert them all at the same time. It is also essential if you are going to make changes on the device from which SystemRescue is being executed. For example, you may be running SystemRescue from an external disk, and you may want to use SystemRescue to repartition the device that it is on. Such an operation would cause the system to crash if it affects the SystemRescue installation on the device. This problem can be avoided by copying the system to memory during the system initialisation so it does not depend on files from the device after initialisation.

Running the system from memory is also more reliable than running it from a removable device. Unfortunately, read or write errors happen rather frequently with optical discs and USB memory sticks. A read error may happen while a program attempts to access files located on bad sectors. This might cause the application or the system to crash while you are in the middle of an important operation. In general, RAM is much more reliable, so this scenario is less likely to happen if you are running the system from memory.

Just remember that running SystemRescue from memory is strongly recommended if you have enough resources on your computer, especially if you plan to work from this system for an extended period or if you must perform a critical operation. The additional boot time can be minimized by using a fast boot device, such as a USB 3 memory stick.

Boot and verify the system integrity (checksum)

The boot menus also have an entry to check the integrity of the system during the boot process.

Unexpected problems are likely to happen if SystemRescue has been installed from a corrupted copy of the ISO image, or if it has been installed on a device which has bad sectors. These types of corruptions can cause the system or applications to crash, or to fail with unexpected errors.

The integrity of the ISO image can be verified manually using checksum files, as explained in the previous chapter. This entry in the boot menu provides an alternative way to verify the system's integrity. When you boot using this entry, the system calculates the checksum of the image located on the boot device and it compares it with the expected checksum. The boot process stops and shows an error message if it detects any corruption.

It is recommended that you use this option at least the first time you start SystemRescue from a new device, or when you experience stability problems. Such issues can also occur if the computer memory is defective. Running `Memtest86+` or any other memory tester can help determine if some memory modules must be replaced.

Boot entries implementations

Each entry in the boot menu correspond to a particular boot command line which controls how SystemRescue starts and behaves. These command lines start with essential parameters such as references to files which are loaded in memory. Optional parameters can be added at the end of a line to change the way the system is executed.

The main boot entries described above all load the same system image files but they use different boot options. For example, the entry which causes the system to be copied to memory is the same as the default entry except that the keyword `copytoram` has been added to the end of the command line. Also, the boot command line which triggers a verification of the system's integrity is similar to the default entry, but with the keyword `checksum` added to the end of the command line.

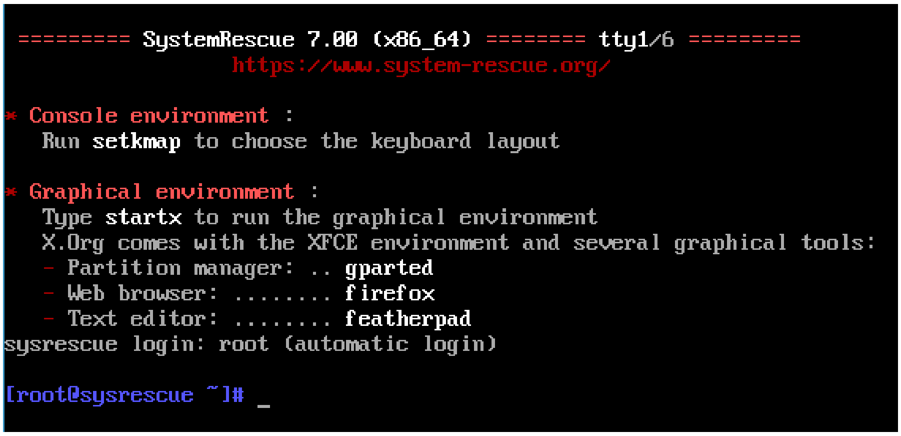
The boot menus offer only a small number of pre-defined choices that should satisfy common needs. But sometimes you might want to start the system with a different combination of options. For example, you might want to enable the options `copytoram` and `checksum` at the same time so the system runs from memory and it also verifies its integrity. Fortunately, the boot command lines that correspond to the entries in the boot menu can be edited before they are used, so you can control how the system will be executed.

The way you edit and execute boot command lines depends on the type of boot menu you are using. Instructions appear on the screen to remind you how you can edit boot command lines. Entries in the BIOS boot menu can be edited by pressing `Tab` and executed by pressing `Enter`. In the UEFI boot menu, entries are modified by pressing `e` and they are executed by pressing `F10`.

This chapter describes only the most important boot parameters. There are many others. There is a list of the most useful parameters in Appendix 1: Boot Parameters and you can get more details on the official project website. You may also want to make your changes to these entries persistent so the system always boots using the options you prefer. Then you won't have to edit these entries each time. If you boot SystemRescue from a writable device such as a memory stick, you can edit text files which define these boot entries. Please refer to Appendix 2: How to Customise Boot Entries for more details.

Booting the system

When you activate an entry from the boot menu, the system starts, and this console prompt appears when the system is ready:



```
 ========= SystemRescue 7.00 (x86_64) ======== tty1/6 =========
                  https://www.system-rescue.org/

* Console environment :
   Run setkmap to choose the keyboard layout

* Graphical environment :
   Type startx to run the graphical environment
   X.Org comes with the XFCE environment and several graphical tools:
   - Partition manager: .. gparted
   - Web browser: ........ firefox
   - Text editor: ........ featherpad
sysrescue login: root (automatic login)

[root@sysrescue ~]# _
```

If you don't see this screen, you should try to identify error messages, and verify the integrity of the system files using the corresponding entry from the boot menu. Also, you can test your computer memory using `Memtest86+` (if you use the legacy BIOS mode) or any other memory tester.

You do not need a username or password to start using SystemRescue. The session automatically runs as the `root` user account, as this is required to perform most maintenance operations. The system won't stop you from doing risky activities such as erasing a disk, so you must be sure that what you are doing is right. Always think twice before you execute sensitive or destructive commands.

Configuring the keyboard layout

The first thing to do when you see the console prompt is to configure the keyboard so it matches your layout. As explained on the screen you should type `setkmap` and press `Enter`. It will display a list of supported keyboard layouts. Select the appropriate choice and, ideally, note its exact name for future references.

The next time you start the system, you can directly specify the name of the layout on the command to not have to go through the list. For example, if you are based in France, you can just type `setkmap fr-latin1` and press `Enter`. Or you can add an option such as `setkmap=fr-latin1` to the system boot command line so the keyboard is automatically configured for your country when the system starts.

When you work in console mode, you can run only applications that run in text mode. Lots of powerful programs can be run this way. You have access to multiple virtual consoles, and you can switch between them by pressing `Alt+F1`, `Alt+F2`, `Alt+F3`, and so on. These allow you to run multiple programs in parallel.

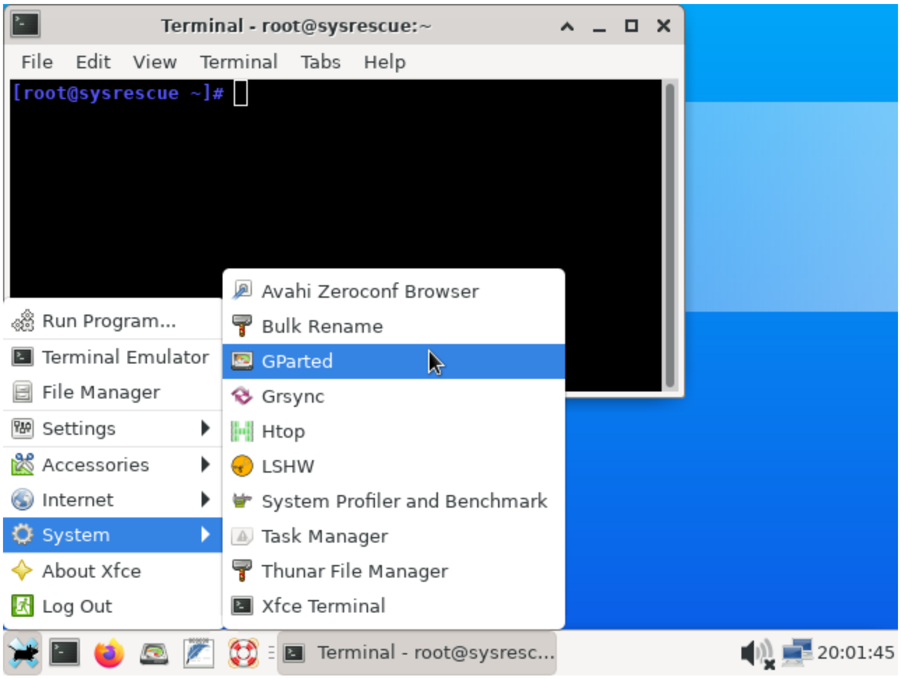
Starting the graphical environment

You can continue working in the text environment if it suits you, but most users prefer to switch to a graphical environment at this stage. Hence, it is highly recommended that you type `startx` and press `Enter` as the instructions on the screen suggest. After a few seconds, you should see a graphical desktop on your screen. If you have problems getting to the graphical environment, try to restart the system using the boot entry that uses basic display drivers.

Chapter 4: System Overview

Graphical environment

When you run the `startx` command in the initial console, the graphical environment appears. It is based on the lightweight XFCE desktop environment which comes with a traditional user interface.



```
[root@sysrescue ~]# 
```

The most important applications can be started directly from icons at the bottom of the screen. The XFCE icon in the bottom left corner gives you access to the application menu where you can find all other available graphical applications as well as settings such as the screen resolution. This menu also provides access to the `Log Out` entry, which you use to shutdown the computer properly at the end of your session. The network and the sound can be configured using icons located next to the clock on the right side of the screen.

Applications overview

Here is an overview of the key graphical applications that can be started from icons in the panel:

- The terminal is required to execute command-line programs and applications which run in text mode. It is used extensively in the following chapters.
- The web browser gives access to internet websites so you can search for tutorials for performing a particular operation and online documentation in general
- The graphical partition editor allows you to manipulate partitions on your disks. There is more information about how to use it in Chapter 9: Partitioning Disks
- The graphical text editor lets you easily modify configuration files or write personal notes.
- The help icon provides an offline inventory which lists all programs that come with SystemRescue sorted by category.

The application menu gives access to many more programs. It is recommended that you review all the categories in the menu to know which graphical applications are available. Here is an overview of what you will find:

- There are utilities which provide detailed information about your hardware.
- Graphical file managers allow you to list, copy, move, rename or delete files.
- Other programs allow you to connect to other computers over the network using protocols such as VNC and RDP.
- Different types of text editors are included to suit everyone’s preferences, and hexadecimal editors are provided to view or edit binary files.

- Some general tools such as a calculator can be helpful for example if you must calculate disk sizes to perform changes.

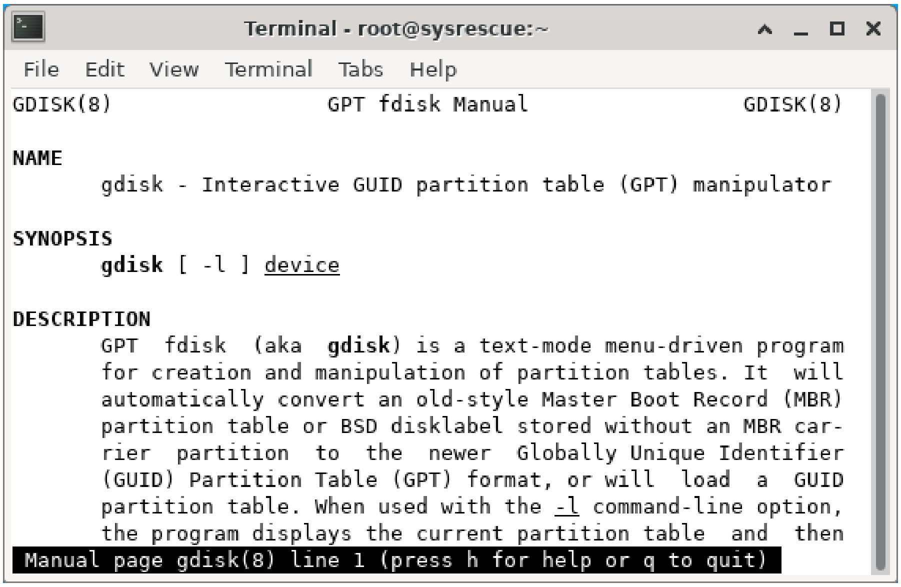
Documentation and user guides

As mentioned before, SystemRescue provides many programs to administrate your system, but you are in charge of the whole process and you must know which tools to use and how to use them.

The following chapters provide step-by-step instructions for performing some common operations. But the tools included in SystemRescue can do much more, and you need to know how to get started.

The program inventory, which is accessible from the help icon on the panel, is a good starting point. It provides a list of available administration tools. As most programs must be executed as command lines from the terminal, they do not have corresponding icons in the application menu. You can find which programs correspond to your needs using this inventory.

Command-line programs normally come with an offline manual. These can be accessed using the `man` command in the terminal. For example, `man gdisk` displays the manual for the `gdisk` program, as shown on the following screenshot. You can scroll using the arrows keys, or Page-up/Page-down, and press `q` to quit.


Terminal - root@sysrescue:~
File Edit View Terminal Tabs Help

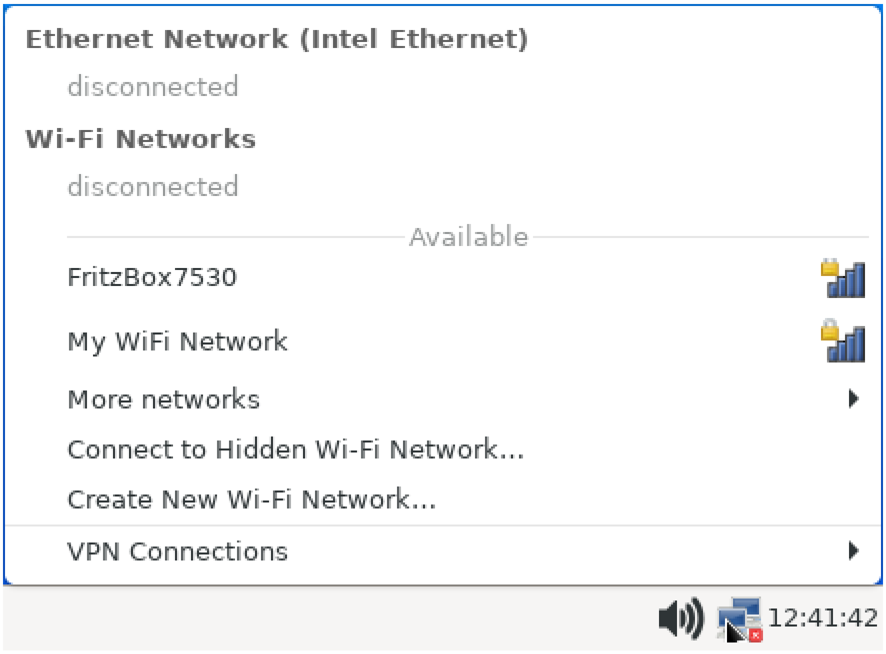
```
GDISK(8)                  GPT fdisk Manual                  GDISK(8)

NAME
       gdisk - Interactive GUID partition table (GPT) manipulator

SYNOPSIS
       gdisk [ -l ] device

DESCRIPTION
       GPT  fdisk  (aka  gdisk) is a text-mode menu-driven program
       for creation and manipulation of partition tables. It  will
       automatically convert an old-style Master Boot Record (MBR)
       partition table or BSD disklabel stored without an MBR car-
       rier  partition  to  the  newer  Globally Unique Identifier
       (GUID) Partition Table (GPT) format, or will  load  a  GUID
       partition table. When used with the -l command-line option,
       the program displays the current partition table  and  then
 Manual page gdisk(8) line 1 (press h for help or q to quit)
```

These manuals provide a description of the application and an exhaustive list of the parameters that the command supports. They are very useful as reference documentation. To get started with new programs, it is recommended that you do some research on the internet to find guides that are more accessible.

Chapter 5: Network Connectivity

Configuring the network

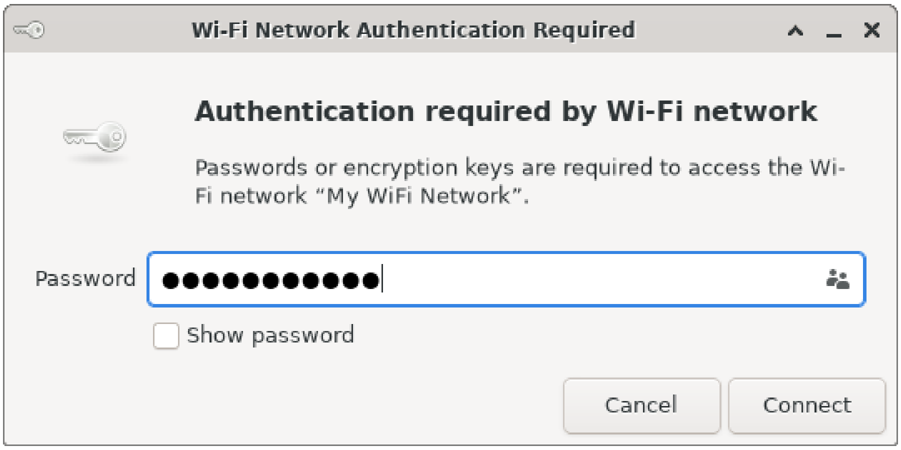
You will sometimes need to connect to networks while you work from SystemRescue, either to get documentation from the internet or to transfer data to or from other computers. The network configuration is controlled by the NetworkManager service by default. It comes with user-friendly tools to configure both Ethernet and Wi-Fi network adapters from either the graphical interface or using commands in the terminal. The graphical configuration tool is accessible from an icon located next to the clock. When you click the icon with the left button of your mouse, it shows a menu like the following one:



You should see one entry for each Ethernet interface and a list of

Wi-Fi networks if your computer supports them. If you don't see what you expected, it probably means there is no driver for your network adapter. It is more likely not to be supported if your hardware is very recent or if you use an old version of SystemRescue. Linux drivers are regularly added and updated, so you should upgrade to the latest version if your hardware is not supported.

To connect to a Wi-Fi network, just click on its name in the menu. If the network is encrypted, it will ask you for a password:


Wi-Fi Network Authentication Required

Authentication required by Wi-Fi network

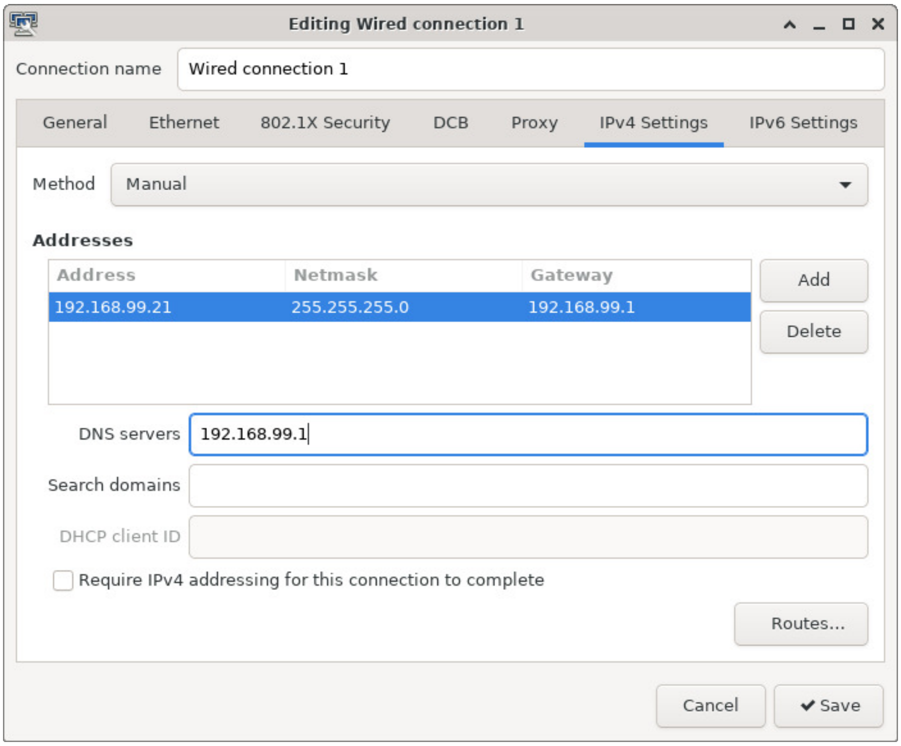
Passwords or encryption keys are required to access the Wi-Fi network "My WiFi Network".

Password ●●●●●●●●●●●●

Show password

If the network uses dynamic addresses, the connectivity should be configured automatically. This is the case with most home and corporate networks nowadays. In general, networks are configured with DHCP to provide a dynamic IP address, DNS servers, and a default gateway so connected devices can easily access the internet.

If your network does not use such a dynamic configuration, you must configure static addresses manually. To do this, go to the graphical interface and double click on a particular connection. Then choose either `IPv4 settings` or `IPv6 settings` and select the `Manual` method instead of `Automatic (DHCP)`.


Editing Wired connection 1

Connection name: Wired connection 1

General | Ethernet | 802.1X Security | DCB | Proxy | IPv4 Settings | IPv6 Settings

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.99.21	255.255.255.0	192.168.99.1

DNS servers: 192.168.99.1

Search domains:

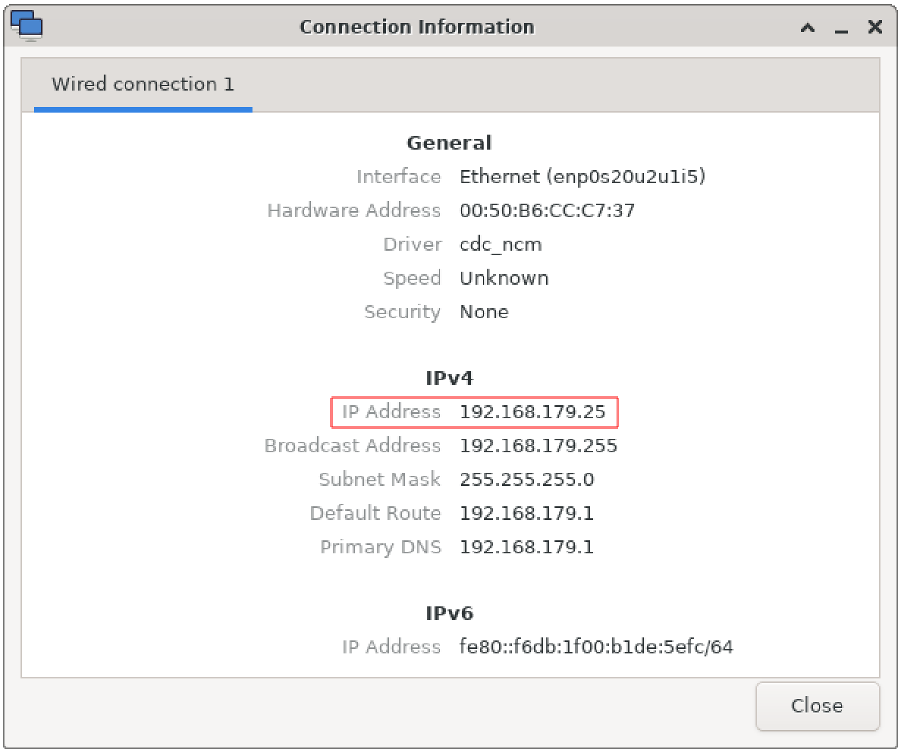
DHCP client ID:

☐ Require IPv4 addressing for this connection to complete

If you prefer, you can also configure the network using command-line tools. NetworkManager can be controlled using both `nmcli` and `nmtui` in a terminal. Also, traditional Linux network commands such as `ifconfig` and `ip` are available.

State of the network connection

To check the network connection, click the NetworkManager icon using the right button of your mouse and select `Connection information` from the menu. It shows information that includes the IP address your computer is currently using. You will need this address if you want to connect from another computer, so make a record of it.



General

Interface	Ethernet (enp0s20u2u1i5)
Hardware Address	00:50:B6:CC:C7:37
Driver	cdc_ncm
Speed	Unknown
Security	None

IPv4

IP Address	192.168.179.25
Broadcast Address	192.168.179.255
Subnet Mask	255.255.255.0
Default Route	192.168.179.1
Primary DNS	192.168.179.1

IPv6

IP Address	fe80::f6db:1f00:b1de:5efc/64

How to manage the firewall

By default, the firewall is enabled in SystemRescue, and it blocks only incoming connections. This prevents sensitive information stored on the computer from being exposed unintentionally. It allows you to establish outbound connections, so you can visit internet websites and connect to other computers from SystemRescue. Therefore, you won't need to change the firewall configuration in most cases.

You will have to intervene when you need to receive incoming connections on a computer running SystemRescue, for example for transferring data from a damaged computer as explained in Chapter 11: How to Recover Files Over a Network. In situations like these, you must authorise incoming connections either by adding a new rule in the firewall configuration or by stopping the firewall completely.

In SystemRescue, the firewall is controlled by two services: `iptables` and `ip6tables` which manage IPv4 and IPv6 respectively. You can check if the firewall is active by looking at the state of these services using the following commands in a terminal:

```
[root@sysrescue ~]# systemctl is-active iptables
active
[root@sysrescue ~]# systemctl is-active ip6tables
active
```

The most secure way to authorise new incoming connections is to create a rule which authorises only the connections you need. Then the system blocks all other incoming connections. For example, you might want to let another computer, which uses address `192.168.99.21`, connect to the SSH service running in SystemRescue. The SSH service uses TCP port 22 by default. Hence you would authorise such a connection using the following command in a terminal:

```
[root@sysrescue ~]# iptables -I INPUT -p tcp -m tcp --dport 22 -s 192.168.99.21
-j ACCEPT
```

If you are connected to a trusted network, it can be easier to completely stop the firewall, so all incoming connections are allowed. You can do this by stopping both services using the following command:

```
[root@sysrescue ~]# systemctl stop iptables ip6tables
```

You can then confirm that the two services have been stopped:

```
[root@sysrescue ~]# systemctl is-active iptables
inactive
[root@sysrescue ~]# systemctl is-active ip6tables
inactive
```

You can also use the following iptables commands if you want to check what firewall rules and policies are in effect:

```
[root@sysrescue ~]# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
[root@sysrescue ~]# ip6tables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

You should focus on the target associated with the `INPUT` policy in the commands output. The target will be `ACCEPT` when the firewall authorises all inbound connections. When the target is `DROP` the firewall is not allowing new connections, so it needs specific rules for such connections to be authorised.

Chapter 6: Disks Structures

Many operations performed with SystemRescue involve manipulating disks. This includes disk partitioning, creating or restoring backups of an operating system and recovering data files. It is important to have a good understanding of how disks are structured so you can perform these operations successfully. This chapter provides a bottom-up overview of how disks are structured. It applies to devices such as SSD disks, magnetic hard disks and USB memory sticks.

Disk devices

From the point of view of an operating system, a disk is simply a long sequence of bytes, which are grouped by sectors of typically 512 or 4096 bytes. For example, a 500GB disk is just a sequence of 500 billion bytes and it corresponds to around 1 billion sectors on a disk that uses 512 bytes per sector. A lot must happen to turn this unstructured sequence of bytes into files and folders that organise the storage of data.

Partition tables

Disks are often divided into partitions so the space can be used as if there were multiple smaller disks. For example, a 500GB disk could be partitioned into two partitions: a 200GB partition can be used to store operating system files and a 300GB partition can be allocated to user data.

The first few sectors of a physical disk often contain a partition table. This small structure defines a list of partitions on the disk and their attributes such as their start locations and sizes. For example here is a partition table on a 500GB disk as shown by the fdisk utility:

```
[root@sysrescue ~]# fdisk --list /dev/sdc
Disk /dev/sdc: 465.76 GiB, 500107862016 bytes, 976773168 sectors
Disk model: Samsung SSD 850
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: C616E018-6795-40A1-AFC1-5C91D7DE6C41

Device         Start       End   Sectors   Size Type
/dev/sdc1      20480   8407039   8386560     4G EFI System
/dev/sdc2    8407040  19402751  10995712   5.2G Linux filesystem
/dev/sdc3   19402752 871915519 852512768 406.5G Linux filesystem
/dev/sdc4  871915520 871948287     32768    16M Microsoft reserved
/dev/sdc5  871948288 955873279  83924992    40G Microsoft basic data
/dev/sdc6  955873280 960067583   4194304     2G Microsoft basic data
/dev/sdc7  960067584 976773119  16705536     8G Linux filesystem
```

In this example, you can see the disk uses sectors of 512 bytes each, and the partition table contains seven entries. Each entry defines where a partition starts and ends in terms of sectors, which are numbered from the beginning of the disk.

There are different types of partition tables. The most common types are MBR/MSDOS and GPT. MBR/MSDOS partition tables are old, and they are quite limited as they support only four primary partitions and each entry is limited to around 4 billion sectors. This limits the size of a partition to 2TB on disks using sectors of 512 bytes. GPT partition tables have been introduced more recently to remove these limitations. They also provide support for additional attributes, so each partition can have a name. Disks based on GPT also have a second copy of the partition table located at the end of the disk. This means it can be recovered if the first copy is overwritten or damaged.

Keep in mind that most disks contain a partition table even if all the space is allocated to a single partition. Also, some operating systems use small partitions which are not always visible in the file manager. This may give the impression the disk contains a single partition when there are more than one. You can check how a disk is partitioned by running tools such as GParted or fdisk from SystemRescue.

File systems

A partition on a disk is a subset of a disk, hence it is also a sequence of bytes. File systems turn these sequences of bytes into logical structures which allow data to be stored in files that are organised in directories. A file system manages all these structures including each file's contents and attributes such as timestamps and permissions.

Each operating system uses a different type of file system. Some provide only very basic features and others have advanced features such as transparent compression and snapshots. For example, MS-DOS and early versions of Windows use FAT file systems which are very limited. Recent versions of Windows use the new technology file system (NTFS) which removed old restrictions and introduced features such as support for access controls and transparent compression. On Linux, the most common native file systems are currently EXT4, XFS and BTRFS. Fortunately Linux also supports FAT and NTFS file systems, so SystemRescue can access files used by Windows.

Chapter 7: Identifying Disks

Disk names on Linux

Many administrative tasks involve reading and writing to and from disks, so it is important to know how to access these devices from SystemRescue. The operating system allocates a device name to each disk. You use these names to run disk-related operations.

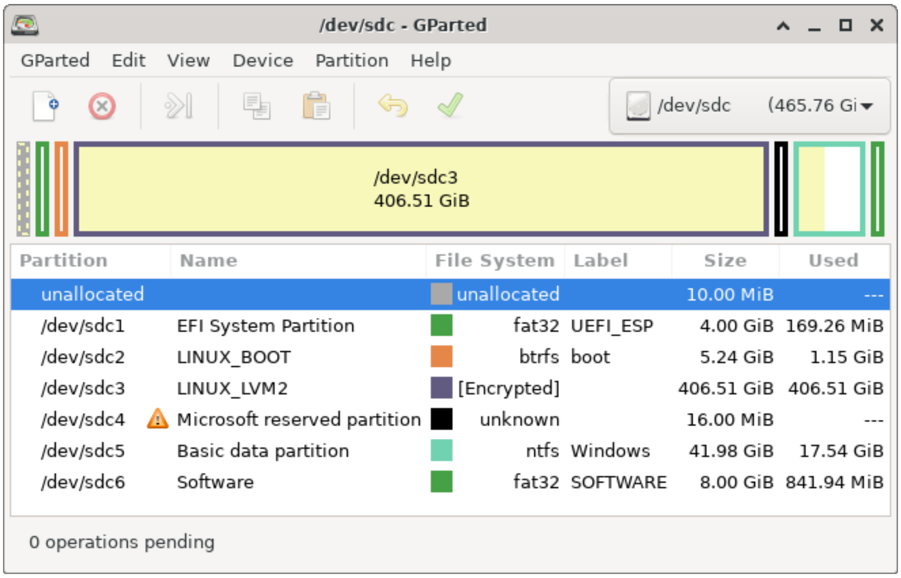
Different types of disks use different naming schemes. SATA disks are allocated names such as `/dev/sda` for the first device, `/dev/sdb` for the second such disk, and so on. NVMe disks get names such as `/dev/nvme0n1`, then `/dev/nvme0n2`, etc.

The partitions on a disk are identified by a number that follows the name of the disk. If the name of the disk ends with a number, the letter `p` is used to separate the partition number from the name of the disk. Here are a few examples:

- `/dev/sda1` is the first partition on disk `/dev/sda`
- `/dev/nvme0n1p1` is the first partition on disk `/dev/nvme0n1`
- `/dev/nvme0n1p2` is the second partition on disk `/dev/nvme0n1`

Identifying disk devices

SystemRescue comes with GParted. It is a graphical partition editor that allows you to identify disks and partitions easily. The main window shows the contents of one disk at a time. If you have multiple disks on your computer, you can switch between them using the drop-down list in the top right corner of the window.



Partition	Name	File System	Label	Size	Used
unallocated		unallocated		10.00 MiB	---
/dev/sdc1	EFI System Partition	fat32	UEFI_ESP	4.00 GiB	169.26 MiB
/dev/sdc2	LINUX_BOOT	btrfs	boot	5.24 GiB	1.15 GiB
/dev/sdc3	LINUX_LVM2	[Encrypted]		406.51 GiB	406.51 GiB
/dev/sdc4	⚠ Microsoft reserved partition	unknown		16.00 MiB	---
/dev/sdc5	Basic data partition	ntfs	Windows	41.98 GiB	17.54 GiB
/dev/sdc6	Software	fat32	SOFTWARE	8.00 GiB	841.94 MiB

0 operations pending

GParted shows disk layouts in two ways. At the top of the window is a graphical view of the partitions. The bottom shows the same partitions as a list with more details. The screenshot above shows an example of a SATA disk named /dev/sdc which contains six partitions.

You can also use the lsblk command in a terminal to identify disks. In the example below this command is used with additional options to specify which attributes to display for each device:

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,model
PATH          SIZE FSTYPE      LABEL     MODEL
/dev/sdc    465.8G                       Samsung_SSD_850_EVO_mSATA
/dev/sdc1      4G vfat        UEFI_ESP
/dev/sdc2    5.2G btrfs       boot
/dev/sdc3 406.5G crypto_LUKS
/dev/sdc4     16M
/dev/sdc5     42G ntfs        Windows
/dev/sdc6      8G vfat        SOFTWARE
```

The lsblk command shows the most important attributes for each disk and partition. As you can see in the example above it provides the model of the disk, the type and label of each file system, and the size of each device. These help you to identify your disks and find which device name to specify when you perform an operation involving disks or partitions. This command is used extensively in the following chapters.

Chapter 8: Accessing File Systems

Once you have identified your disks, you might want to get access to the files they contain. To make files stored on a device accessible, you must mount the device on a directory. The directory is the location from which files become visible. SystemRescue supports all mainstream file systems, including EXT4, XFS and BTRFS, which are used on Linux systems, and FAT and NTFS, which are used on Windows.

Unlike most Linux distributions, SystemRescue does not automatically mount file systems. It takes more effort to mount file systems manually, but this is required so you can control which file systems are being accessed. SystemRescue is often used to create physical copies of disks, as explained in Chapter 12: How to Create Disk Clones and Images, so it is essential that file systems stored on these disks are not mounted during such operations. Manually mounting and unmounting file systems is easy once you are familiar with the commands involved.

Mounting a file system

You can mount a file system by running two simple commands in a terminal. As an example, we are going to mount a device named `/dev/sdc6` on a new directory named `software` so files can be accessed from this location. This new directory is called a mount point as it is where the device will be mounted. You can give this directory any name you like, but avoid spaces as these are used as separators in command lines.

First, you create a mount point. It is recommended that you create this new directory as a sub-directory of `/mnt` to keep things tidy. This new directory can be created with the `mkdir` command. The best approach is to run the command with the full path to the new directory as well as the option `-p` so it also creates parent directories (ie `/mnt`) if they do not already exist:

```
[root@sysrescue ~]# mkdir -p /mnt/software
```

The second step is to mount the device on the new mount point. You do this with the `mount` command, which requires two parameters. The first parameter is the name of the device which contains files you want to access. In the following example, the name is `/dev/sdc6`. The second parameter is the directory where the device will be mounted, `/mnt/software` in the example. The `mount` command also accepts optional parameters, and we are going to use the `-o ro` option to specify that the file system must be mounted in read-only mode. We are also adding option `-v` to increase the verbosity. Because of this, the command shows a message to confirm that it was successfully executed.

```
[root@sysrescue ~]# mount /dev/sdc6 /mnt/software -o ro -v
mount: /dev/sdc6 mounted on /mnt/software.
```

It is recommended that you mount the device in read-only mode if you do not plan to make any change to the files. This way, you are sure that these files cannot be deleted or modified by mistake. Once the device is mounted, use the `df` command, which stands for disk free. It shows how much space is in use on this device:

```
[root@sysrescue ~]# df -hT /mnt/software
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/sdc6      vfat  8.0G  826M  7.2G  11% /mnt/software
```

Navigating through file systems

You can explore the contents of your mounted file systems using applications such as `Thunar`, which is a graphical file manager accessible from the main menu, or by using Midnight Commander which is introduced below.

Here is how you show the contents of a mounted file system from the command line in the terminal. The `ls` command shows the files and directories which are present in a particular location. It is recommended that you use this command with the `-lh` options to get a more readable list. Specify the full path to the mount point. Here is an example of what it shows:

```
[root@sysrescue ~]# ls -lh /mnt/software
total 16K
drwxr-xr-x 2 root root 4.0K Sep  6 16:11 drivers
drwxr-xr-x 2 root root 4.0K Sep  6 16:11 systemrescue
drwxr-xr-x 2 root root 4.0K Sep  6 16:07 temp
drwxr-xr-x 2 root root 4.0K Sep  6 16:10 utilities
```

Another way to show the contents of the device is to change the current directory using `cd` and then run `ls` without providing any location:

```
[root@sysrescue ~]# cd /mnt/software
[root@sysrescue /mnt/software]# ls -lh
total 16K
drwxr-xr-x 2 root root 4.0K Sep  6 16:11 drivers
drwxr-xr-x 2 root root 4.0K Sep  6 16:11 systemrescue
drwxr-xr-x 2 root root 4.0K Sep  6 16:07 temp
drwxr-xr-x 2 root root 4.0K Sep  6 16:10 utilities
```

Access control

Linux and Windows operating systems have implemented access control on all modern file systems so access to files and directories can be restricted to specific users. Access restrictions prevent ordinary users from damaging files that belong to the operating system. In most organisations, these restrictions are also used to control which people can access sensitive files.

The default session in SystemRescue uses the `root` user account, which comes with all administrative rights. Hence the system will not stop you from accessing files on local disks even if these files have access restrictions. You can use the unrestricted access in SystemRescue to copy files which were restricted by other operating systems installed on the computer.

Accessing files

Here are some examples of operations that you can perform once a device has been mounted:

- If the operating system on your computer does not start, you can use SystemRescue to transfer important data files from an internal disk to an external disk before you erase the internal disk and reinstall the operating system. Chapter 10: How to Recover Files to Another Disk explains how to recover data files this way.
- You might have modified a system configuration file incorrectly causing the system to stop working. You can use SystemRescue to access such system configuration files to revert the change and fix the system.
- You may want to backup or restore a Linux operating system by copying all its system files using commands such as `rsync`.

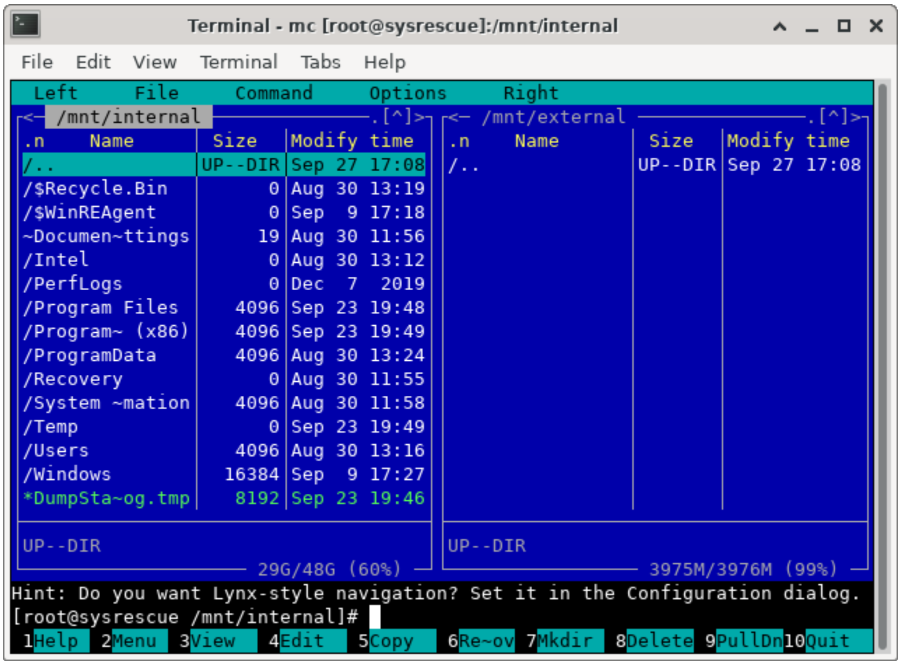
There are many ways to manipulate files in SystemRescue:

- To modify some files, for example, to fix a system configuration file, use any editor that comes with SystemRescue. Some editors are graphical, and others, such as `vim` and `nano`, work in the terminal.
- The graphical file manager is accessible from the main menu. You can use it to copy, move, rename or delete files. You access files by typing the full path to the mount point in the address bar at the top or by navigating using the sidebar. You can also perform these operations with commands such as `cp` to copy, `rm` to remove, `mv` to move or rename. Please refer to the corresponding reference documentation for more details.
- Midnight Commander, which is introduced next, is a very convenient way to work on files if you are not familiar with command lines.

Midnight Commander

This powerful tool lets you view, edit, copy, move, rename or delete files and directories. You start Midnight Commander by executing `mc` in a terminal. By default, it comes with two panes so you can easily copy or move files across two locations. When you want to switch from one pane to the other, press `Tab`. To navigate, use the arrow keys and `Enter`. To return to the parent directory, select the `..` entry at the top and press `Enter`.

The bottom of the application shows actions which can be triggered by pressing the corresponding function keys on the keyboard. For example, use `F3` to view the file which is currently selected, `F4` to edit it, `F5` to copy it, and `F6` to move it. To create a new directory use `F7`, and use `F8` to delete the selection. To quit press `F10`.


Terminal - mc [root@sysrescue]:/mnt/internal
File Edit View Terminal Tabs Help
Left File Command Options Right

```
<- /mnt/internal ---------------.[^]>  <- /mnt/external ----------------.[^]>
.n      Name       Size    Modify time   .n      Name           Size    Modify time
/..              UP--DIR Sep 27 17:08    /..                  UP--DIR Sep 27 17:08
/$Recycle.Bin          0 Aug 30 13:19
/$WinREAgent           0 Sep  9 17:18
~Documen~ttings       19 Aug 30 11:56
/Intel                 0 Aug 30 13:12
/PerfLogs              0 Dec  7  2019
/Program Files      4096 Sep 23 19:48
/Program~ (x86)     4096 Sep 23 19:49
/ProgramData        4096 Aug 30 13:24
/Recovery              0 Aug 30 11:55
/System ~mation     4096 Aug 30 11:58
/Temp                  0 Sep 23 19:49
/Users              4096 Aug 30 13:16
/Windows           16384 Sep  9 17:27
*DumpSta~og.tmp     8192 Sep 23 19:46

UP--DIR                                  UP--DIR
             29G/48G (60%)                          3975M/3976M (99%)
Hint: Do you want Lynx-style navigation? Set it in the Configuration dialog.
[root@sysrescue /mnt/internal]#
 1Help  2Menu  3View  4Edit  5Copy  6Re~ov 7Mkdir  8Delete 9PullDn10Quit
```

Unmounting file systems

When you have finished working on files, you are encouraged to execute the `sync` command in a terminal. This causes the system to flush any pending change to the disk.

Devices must be unmounted properly to make sure the file system structure remains consistent. To unmount a file system, use the `umount` command in a terminal. You must specify either the name of the device or the corresponding mount point. As an example, let's consider a device named `/dev/sdc6` which was mounted on `/mnt/software`. This device can be unmounted using either of these commands:

```
[root@sysrescue ~]# umount /dev/sdc6
[root@sysrescue ~]# umount /mnt/software
```

The `umount` command fails if any resource on the devices is still in use.

```
[root@sysrescue /mnt/software]# umount /mnt/software
umount: /mnt/software: target is busy.
```

If you get such an error, you must close all applications which are accessing files or directories on this device, including terminals where the current directory belongs to the device.

Also, when you have finished your operations, shut down SystemRescue properly by clicking `Log Out` and then `Shutdown` in the main menu.

Chapter 9: Partitioning Disks

Overview

SystemRescue comes with many utilities for partitioning disks. They allow you to create, resize, move and delete partitions and the corresponding file systems.

There are many reasons to change the way disks are partitioned. You must normally create partitions and file systems before you can use a new disk. Also, if you copy an old disk to a larger new one, you must change the disk layout so the additional disk space can be used. In this case, you can either create new partitions or resize existing ones to allocate the extra space.

At times, you might want to replace a large partition with multiple smaller ones. For example, many new computers have disks where most of the space is allocated to a single partition. This default layout is simple, but it can be sub-optimal. It is often a good idea to store the operating system files and personal data on different partitions. This allows you to perform operations on the system partition without affecting your personal data files. If you create an image of the disk as described in Chapter 12: How to Create Disk Clones and Images and your operating system stops working, you can restore the system from the disk image. If your data are located on a separate partition, you could restore the system without affecting your data.

You must be very cautious when you work on disk partitions, as these operations can cause you to lose data if things go wrong. It is highly recommended that you back up your data before you start. Also, you can create a physical copy of an entire disk as explained in Chapter 12: How to Create Disk Clones and Images. This allows you to restore the entire disk contents if there is any issue during the operation.

Partitions and file systems

It is essential to understand the difference between partitions and file systems. A partition is just a subset of a physical disk. It is implemented as an entry in the partition table which indicates its location on the disk and its attributes. In general partitions contains file systems which determines how files and directories are organised. A particular file system can make use of all the space allocated to the corresponding partition. In the current chapter, the term "volume" is used to refer to the combination of a partition and the file system it contains.

Most disk partitioning activities involve manipulating both a partition and the corresponding file system. It is essential that the two operations are consistent. For example, you may have a 500GB disk which contains just a single volume where your operating system is installed. If the file system on this partition contains a lot of free space you might want to shrink this volume to 100GB to make space for another volume of 400GB where you could store data files. Shrinking the current 500GB volume involves two operations:

- First, you shrink the file system to 100GB so the file system structures and data are moved to the first 100GB in this partition. As a result, you have a 500GB partition which contains a 100GB file system.
- Next, you shrink the partition to 100GB, by modifying the entry in the partition table, to make space for other partitions.

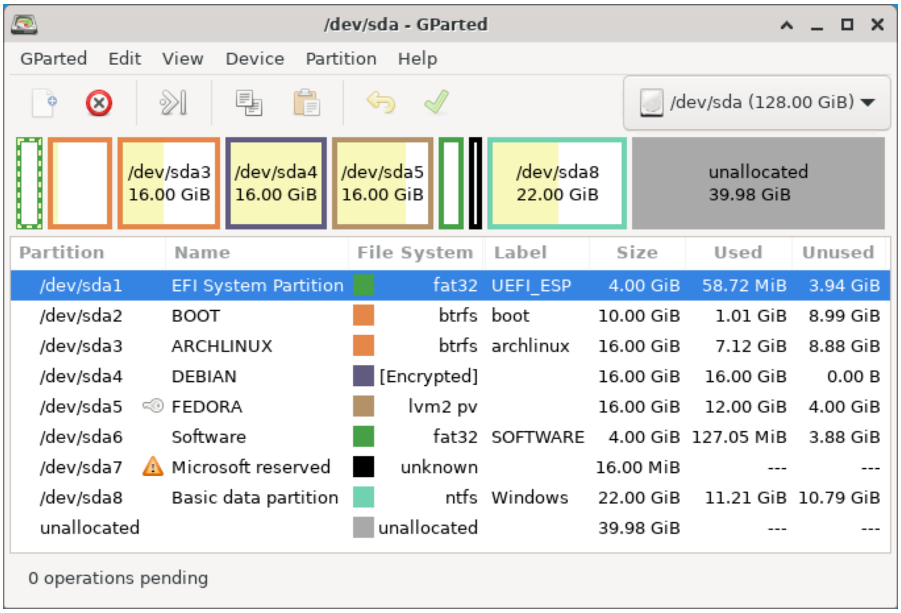
It is important to apply these two operations in the right order so the file system is never larger than the partition it belongs to. Shrinking the partition to 100GB while the file system is still using 500GB would cause the file system to be truncated and it would become unusable. Growing a volume involves doing the two operations in the opposite order: you must grow the partition before you grow its file system.

There are three categories of utilities involved in disk partitioning:

- Programs such as fdisk and gdisk manipulate partitions by working on the partition table, but they do not operate at the file system level.
- Each type of file system comes with its own set of utilities for performing administration tasks. These include creating new file systems and resizing existing ones. These tools do not perform any change on partition tables.
- Tools such as GParted operate on both partitions and file systems and make disk partitioning much easier.

Graphical partitioning tool

GParted is the best tool for most disk partitioning activities. It has a graphical user interface and it is very user friendly. It modifies both partition tables and the corresponding file systems so both structures are modified in a consistent way during an operation.



Partition	Name	File System	Label	Size	Used	Unused
/dev/sda1	EFI System Partition	🟩 fat32	UEFI_ESP	4.00 GiB	58.72 MiB	3.94 GiB
/dev/sda2	BOOT	🟧 btrfs	boot	10.00 GiB	1.01 GiB	8.99 GiB
/dev/sda3	ARCHLINUX	🟧 btrfs	archlinux	16.00 GiB	7.12 GiB	8.88 GiB
/dev/sda4	DEBIAN	🟪 [Encrypted]		16.00 GiB	16.00 GiB	0.00 B
/dev/sda5	FEDORA	🟫 lvm2 pv		16.00 GiB	12.00 GiB	4.00 GiB
/dev/sda6	Software	🟩 fat32	SOFTWARE	4.00 GiB	127.05 MiB	3.88 GiB
/dev/sda7	⚠ Microsoft reserved	⬛ unknown		16.00 MiB	---	---
/dev/sda8	Basic data partition	🟩 ntfs	Windows	22.00 GiB	11.21 GiB	10.79 GiB
unallocated		⬜ unallocated		39.98 GiB	---	---

0 operations pending

It supports all mainstream types of partition tables and file systems. It relies on file system specific utilities to perform operations on file systems, so these have to be installed. SystemRescue comes with all common file system tools.

GParted shows a graphical representation of the disk space so it is very easy to visualise how the disk space is allocated. If your computer has multiple disks, use the drop-down list at the top of the window to switch from one disk to another. Partitions and their corresponding file systems can easily be created, deleted, moved or resized by clicking on actions in the menus. Such operations are not applied immediately. The graphical representation of the disk is updated so you can visualise what effect pending operations would have. Once you are satisfied with the new disk layout you must apply pending operations, either from the menus or by clicking on the green tick in the toolbar.

If you are setting up a new disk, or if you want to reinitialise an existing disk, you can create a new partition table. This action replaces all existing partitions and causes all data on the disk to be lost. This operation can be triggered from the main menu, where you select the type of partition table you want to use. Unlike other operations, this change takes effect as soon as you validate the operation.

Advanced partition manipulation tools

You can also use partition manipulation tools such as fdisk and gdisk. These allow you to create, modify or delete entries in a partition table. Old versions of fdisk did not support GPT partition tables but this has now been implemented. The gdisk utility has the same features as fdisk for GPT partition tables. It can convert an MBR/MSDOS partition table to GPT. These two programs can be executed in a terminal and they are controlled by selecting options interactively from a list of choices. The cfdisk and cgdisk programs provide the same types of features with an alternative user interface in which partitions and commands are selected using arrow keys.

You can use these partition manipulation tools to perform operations that are not supported in GParted. This includes growing a file system which GParted does not support. You would have to grow the partition using one of these tools and then you grow the corresponding file system using its specific utilities.

Please use the `man` command in a terminal to get more details about how these programs operate. For example type `man fdisk` in a terminal to get the fdisk manual page.

File system utilities

SystemRescue comes with utilities to administrate all mainstream file systems such as EXT4, XFS, BTRFS, FAT and NTFS. These tools allow you to perform the following types of operations:

- All mainstream file systems have their own `mkfs` utilities to create a new file system on a partition or disk device (`mkfs.ext4`, `mkfs.xfs`, `mkfs.btrfs`, `mkfs.fat`, `mkfs.ntfs`)
- The `fsck` utilities allow you to verify the structure of a file system and fix errors (`fsck.ext4`, `fsck.xfs`, `fsck.btrfs`, `fsck.fat`, `ntfsfix`)
- Most file systems can be resized but the corresponding utilities have different names. EXT4 are resized using `resize2fs`. XFS file systems can be extended using `xfs_growfs`. BTRFS file systems can be managed using the general `btrfs` utility which provides a sub-command to perform a resizing operation. NTFS file systems can be resized using `ntfsresize`.
- Most file systems have additional commands to perform various operations such as enabling or disabling features and

changing the volume name. Here are examples: `tune2fs`, `xfs_admin`, `btrfs`.

File systems are resized either online (when they are mounted) or offline (when they are not mounted). Each type of file system has its own capabilities. For example, EXT4 can be extended either online or offline but it can be shrunk only offline. XFS file systems can be extended online or offline but they cannot be shrunk. BTRFS file systems can be extended or shrunk, but those operations must be run online. NTFS file systems must be resized offline with the current Linux utilities.

Chapter 10: How to Recover Files to Another Disk

SystemRescue can help you recover data files stored on a local disk when you have lost access to them. This can be essential, for example if the operating system installed on your computer has stopped working and it cannot restart. If the problem is caused by a software issue, such as a defective operating system update, you might have to reinstall the system. This often involves erasing the disk, hence losing data files stored on the disk. In such a situation, you should copy all your important data files to a safe place before you reinstall the system. The challenge is to copy your data files when the operating system is unusable. Fortunately, SystemRescue can help you access your data files when your ordinary operating system is broken.

There are multiple ways to copy your data files to a safe place. This chapter guides you through the process of copying files from one disk to another. In a typical situation, you would copy files located on an internal disk to an external disk. As an alternative, the next chapter shows you how to transfer files to another computer over a network.

The following instructions come with examples of commands applied to a real situation. In this example, we recover data files stored on a Windows NTFS file system and copy them to an external 1TB disk. This is just an example, and these instructions apply to similar situations, as long as the data are stored on file systems that SystemRescue can read. It is important to understand the general process and modify the commands so they match your particular situation.

Starting SystemRescue

If you are following these instructions, the operating system on your computer is probably not in working condition. So, the first thing to do is to boot SystemRescue. You then go to the graphical desktop and open a terminal to be ready to execute commands.

Identifying the disks

This operation involves two file systems. The first file system is where the files to be recovered are currently stored, and it is likely to be on an internal disk. The second file system is where these data files must be copied to, and it could be an external disk. These two file systems can be identified using `lsblk` in a terminal:

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,model
PATH          SIZE FSTYPE      LABEL     MODEL
/dev/sdc    465.8G                       Samsung_SSD_850_EVO_mSATA
/dev/sdc1       4G vfat        UEFI_ESP
/dev/sdc2     5.2G btrfs       boot
/dev/sdc3   406.5G crypto_LUKS
/dev/sdc4      16M
/dev/sdc5      48G ntfs        Windows
/dev/sdc6       2G vfat        SOFTWARE
/dev/sdd    931.5G                       SABRENT
/dev/sdd1   931.5G ext4        external
```

In this example, we copy data from the Windows partition identified as `/dev/sdc5` (the fifth partition on the internal disk) to `/dev/sdd1` which is a unique partition on the external 1TB disk.

Mounting the disks

Next, these two file systems must be mounted. To do this, you must create two mount points. The names of these directories are not important as long as they are the same in every step. Here, we call the mount points `internal` and `external` as the original data files are located on an internal disk and we want to copy them to an external disk. Run these two commands in a terminal to create the two mount points:

```
[root@sysrescue ~]# mkdir -p /mnt/internal
[root@sysrescue ~]# mkdir -p /mnt/external
```

The file system that contains the original data files will be mounted in read-only mode because write access is not required, and this reduces the risk of accidentally deleting these data files.

```
[root@sysrescue ~]# mount /dev/sdc5 /mnt/internal -o ro
```

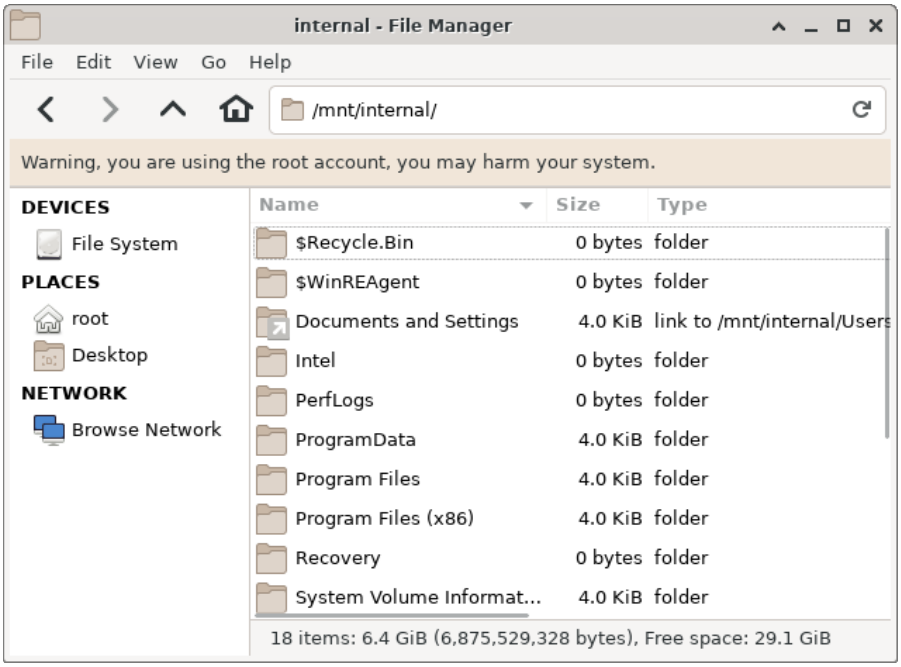
The destination file system must be mounted in read-write mode so data can be written:

```
[root@sysrescue ~]# mount /dev/sdd1 /mnt/external -o rw
```

Copying the data files

At this stage, you can access the original data from `/mnt/internal` and the destination from `/mnt/external`. You can use any application you like to copy files between these two locations. The simplest methods are to use either Midnight Commander, which was introduced in Chapter 8: Accessing File Systems or to use the graphical file manager as follows.

Thunar is a user-friendly graphical file manager. You execute it by clicking on `file manager` in the main application menu. Then you navigate to the mount point where the source files are located - `/mnt/internal` in the current example. You reach this location either using the sidebar or by typing the path of the mount point in the address bar at the top. Then select the files or directories to be copied and click on `Copy` in the menu. Then navigate to `/mnt/external` and select `Paste` in the menu.



Warning, you are using the root account, you may harm your system.

Name	Size	Type
$Recycle.Bin	0 bytes	folder
$WinREAgent	0 bytes	folder
Documents and Settings	4.0 KiB	link to /mnt/internal/Users
Intel	0 bytes	folder
PerfLogs	0 bytes	folder
ProgramData	4.0 KiB	folder
Program Files	4.0 KiB	folder
Program Files (x86)	4.0 KiB	folder
Recovery	0 bytes	folder
System Volume Informat...	4.0 KiB	folder

18 items: 6.4 GiB (6,875,529,328 bytes), Free space: 29.1 GiB

Unmounting the disks

When you have copied all your data files, run `sync` in the terminal to flush all pending operations to the disks. Then you unmount the two file systems using the `umount` commands as shown below:

```
[root@sysrescue ~]# sync
[root@sysrescue ~]# umount /mnt/internal
[root@sysrescue ~]# umount /mnt/external
```

Finally, you shut down SystemRescue properly to make sure the disk structures remain consistent.

Data verification

After you shut down SystemRescue, you should verify that your data have been copied successfully before you perform any operation on the original disk that might destroy data. If you have copied the data to an external disk, you might want to copy the files to another computer and make sure all the files you expect have been copied successfully.

Chapter 11: How to Recover Files Over a Network

The previous chapter provided guidance for recovering data files stored on a computer with a damaged operating system by copying them to another disk. This chapter also is focused on data recovery, but this time files are copied over a network. Files will be copied from a computer running SystemRescue to another computer on the network, so you don't need an external disk for storing a copy of the original files. The two computers must be able to communicate with each other. This is normally not a problem if they are connected to the same home or corporate network by either Ethernet or Wi-Fi.

Let's say the operating system on your primary computer stopped working and you want to transfer your data to another computer that works fine. You will use SystemRescue on the primary computer to access the data stored on the local disk. You will connect the secondary computer to the primary one and download the files. You will transfer the data using the SFTP protocol (SSH File Transfer Protocol) which is secure and easy to use. The computer running SystemRescue acts as a server to share the data over SFTP. The secondary computer acts as an SFTP client that establishes the connection to the primary computer to download the data over the network.

The first few steps are similar to those in the previous chapter as the procedure to access the data located on the internal disk is the same.

Identifying the disk

Start SystemRescue on the primary computer and open a terminal in the graphical environment. Then, identify the name of the device where the data you want to recover are stored, using the `lsblk` command:

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,model
PATH          SIZE FSTYPE      LABEL     MODEL
/dev/sdc   465.8G                        Samsung_SSD_850_EVO_mSATA
/dev/sdc1      4G vfat        UEFI_ESP
/dev/sdc2    5.2G btrfs       boot
/dev/sdc3  406.5G crypto_LUKS
/dev/sdc4     16M
/dev/sdc5     48G ntfs        Windows
/dev/sdc6      2G vfat        SOFTWARE
```

We are going to copy data from the Windows disk identified as `/dev/sdc5` in this example. As usual you should modify these details so they match your situation.

Mounting the disk

The disk that contains the data to be transferred must be mounted so it is accessible. Create the mount point using the `mkdir` command in a terminal. Again we are calling this mount point `internal` as it is assumed the original files are located on an internal disk.

```
[root@sysrescue ~]# mkdir -p /mnt/internal
```

The file system should be mounted in read-only mode as write access is not required, and this reduces the risk of accidentally deleting these data files:

```
[root@sysrescue ~]# mount /dev/sdc5 /mnt/internal -o ro
```

Connecting to the network

Both computers must be connected to the network so data can be transferred. On SystemRescue use the NetworkManager to configure the connection to either an Ethernet or Wi-Fi network. It is accessible from its icon in the notification area next to the clock. You can also use an alternative method such as commands if you prefer. Once you are connected to the network, get the current IP address of this computer so the client knows where to connect in a later step. You get this address by clicking on the NetworkManager icon with the right button of the mouse and selecting `Connection Information` in the menu. If necessary, please refer to Chapter 5: Network Connectivity to get more details about the network configuration.

Starting the SFTP service

The computer running SystemRescue works as an SFTP server to allow the secondary computer to access the data. SystemRescue comes with OpenSSH, which can be used as an SFTP server. It runs as a service in the background, so there is no graphical interface. It can be configured using text files, but the default configuration does not require any change so files can be transferred as documented here.

This service is enabled by default, but you should double check that it is still enabled on the version you are using by running the following command in the terminal:

```
[root@sysrescue ~]# systemctl is-active sshd
active
```

If the service was not active, you should start it using the following command:

```
[root@sysrescue ~]# systemctl restart sshd
```

Disabling the firewall

To increase security, SystemRescue comes with the firewall enabled by default. The firewall blocks incoming connections. In this situation, the secondary computer must be able to connect to the SFTP server running in SystemRescue, so the firewall must be configured or disabled to allow the client to connect.

If you are connected to a trusted network, the simplest solution is to stop the firewall services in SystemRescue using the following command.

```
[root@sysrescue ~]# systemctl stop iptables ip6tables
```

Then check that the firewall services have been stopped successfully:

```
[root@sysrescue ~]# systemctl is-active iptables
inactive
[root@sysrescue ~]# systemctl is-active ip6tables
inactive
```

As an alternative, you could add a rule to the firewall so only the secondary computer is allowed to connect. If necessary, please refer to Chapter 5: Network Connectivity to get more details.

Setting the root password

The SFTP client must authenticate using a username and a password so it can access the data from the SFTP server. The system `root` account is used so the client can access files that are visible in SystemRescue. By default, authentication as the `root` account in SystemRescue is not allowed. A password must be set to allow the client to authenticate. Create a password to use in the following steps. Only the password of the `root` user account on the live system is changed. Passwords used by operating systems on the disk will not be affected by this operation.

Set the password of the `root` account by typing `passwd root` in a terminal in SystemRescue. You will be asked to type the password twice (or multiple times in case of errors). For security reasons, the password does not appear on the screen as you type it.

```
[root@sysrescue ~]# passwd root
New password:
Retype new password:
passwd: password updated successfully
```

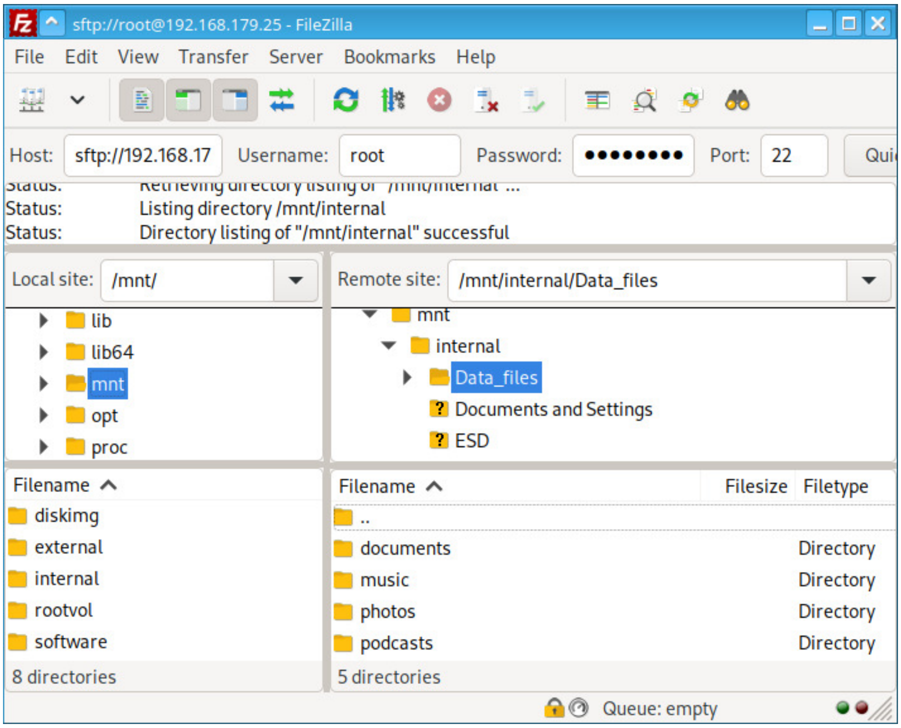
Installing an SFTP client

The client computer must execute an SFTP client to connect to the SFTP server. You can use any SFTP client you like. The recommended program is FileZilla client, as it is a good quality program and it is available for Linux, Windows and MacOS. It can be downloaded for free from the official project website: https://filezilla-project.org/. If you prefer an alternative, please make sure it supports the SFTP protocol, as it is very different from the FTP and FTPS protocols, even though they are all used to transfer files.

An alternative to SFTP is to use rsync. It is a very powerful program and can work with OpenSSH to transfer files over a network. It is widely available on Linux distributions. Using rsync requires more effort at first, so FileZilla client is used in the current example.

Copying data using FileZilla

Now run the FileZilla client on the secondary computer to connect to the SFTP server running in SystemRescue.



sftp://root@192.168.179.25 - FileZilla

Host: sftp://192.168.17 Username: root Password: ●●●●●●●●● Port: 22

```
Status:     Retrieving directory listing of "/mnt/internal"...
Status:     Listing directory /mnt/internal
Status:     Directory listing of "/mnt/internal" successful
```

Local site: /mnt/

- lib
- lib64
- mnt
- opt
- proc

Filename
diskimg
external
internal
rootvol
software

8 directories

Remote site: /mnt/internal/Data_files

- mnt
  - internal
    - Data_files
    - Documents and Settings
    - ESD

Filename	Filesize	Filetype
..		
documents		Directory
music		Directory
photos		Directory
podcasts		Directory

5 directories

Queue: empty

You must first provide the connection details using the quick connect bar at the top of the window.

- The `Host` is the IP address of the computer running SystemRescue.
- The `Username` is `root` as this is the system user account you are using to authenticate on the SFTP server.
- The `Password` is the password you created in a previous step.
- The `Port` should be `22` as it is the default port for the SSH/SFTP service.

When you are ready, press the `Quickconnect` button. If the connection is successful you will see files and folders appear. The screen is divided into two panes. The first pane represents local files and folders from the computer running the FileZilla client. The other pane shows remote files on the primary computer running as an SFTP server in SystemRescue.

On both sides you must navigate to the relevant folders. On the remote side, find the files that you want to copy by navigating to the mount point such as `/mnt/internal`. On the local side, find the folder where you want the remote files to be copied. When you have found these two locations, you can copy the remote files to the local folder by using a simple drag and drop. Select files and folders on the remote side and drag them to the relevant folder on the local side. The copy of the files should start immediately. You can follow the progress by looking at the list of files at the bottom of the screen.

When the operation is complete, use your favourite file manager on the client computer to make sure all files that you expect have been successfully transferred.

Troubleshooting

If the connection is not successful, read the messages that appear just below the connection bar on the FileZilla client. Error messages provide the information you need to understand what has gone wrong. It is important to differentiate connectivity issues from authentication issues to focus on the cause of the problem.

A connectivity problem occurs when the client cannot communicate with the server. This can happen if the network is not configured properly. For example, one of the two computers might not be connected to the network, the two computers might not be on the same network, or the wrong IP address was used on the SFTP client. Connectivity issues can also happen when network traffic is blocked by a firewall on either computer or by network equipment between them.

For authentication problems, the SFTP client and server applications can communicate over the network, but the SFTP server rejects the connection request from the client. This can happen if the username or password is incorrect or when the SSH/SFTP server is configured to restrict incoming connections.

You should also check the OpenSSH service logs by running `journalctl -f --unit=sshd` in a terminal in SystemRescue before you try to connect again from the client. If you see authentication errors, it means there is no connectivity problem and you should check the username and the password. If you don’t see any error message, it is an indication that the server has not received any connection request from the client, so you must fix the connectivity problem by checking the most likely causes mentioned above.

Chapter 12: How to Create Disk Clones and Images

This chapter presents the steps to follow to create physical copies of disks. A disk may be copied to another disk or to an image file that can be restored in the future.

There are various reasons for creating copies of disks. First, you can create a copy of a disk to have a backup. An internal disk typically stores lots of contents such as the operating system, some applications, their configurations and some data files. If anything goes wrong, you would be able to restore all these contents from the backup you created. Having such a backup helps in case of software problems, for example, if the system stops working after a faulty system update or because of malware. Having a copy would also be useful if the disk must be replaced because of a hardware fault or as part of an upgrade.

Copying disks is also useful if you have installed the operating system and application on a computer and you want to duplicate this installation to similar computers without having to repeat the same installation procedure multiple times. Hence there are several use cases for copying disks.

Logical and physical approaches

There are multiple valid ways to copy disks. This chapter focuses on creating and restoring physical copies. Before we dive into the details, it is important to understand how the physical approach differs from a logical approach.

A logical copy is created by analysing the disk structures on the original disk and then recreating these structures and the data on the destination disk. As explained in Chapter 6: Disks Structures disks normally contain a partition table and multiple partitions which contain file systems. Copying disks logically requires programs to copy the partition table and each file system. It requires multiple operations, and each program involved must properly support a particular type of partition table or file system so the copy is successful. For example, the operating system installed on the disk may rely on file system attributes (such as a volume name or unique identifier) so it can recognize the file system where it is installed. Also, the boot loader which starts the system may rely on having boot sectors in specific locations on the disk. A logical copy of a disk will not be usable if any of these structures or attributes have not been properly recreated.

A physical copy is an exact copy of an entire disk. A disk is a long sequence of bytes. A physical copy does not involve complicated operations. All that is required is to copy each byte in the sequence. To perform a physical copy, the program does not need to understand how the disk is structured, so it does not need to support a particular type of partition table or file system. A physical copy has the same size as the original disk, as it is an exact copy.

There are also intermediate approaches between a strict physical copy and a logical copy. For example, some disk cloning utilities can copy disks at a block level and skip sections that are not in use.

This chapter focuses on creating strict physical copies of disks, as this approach has important advantages:

- It does not require any sophisticated program to create such a copy, as the utilities involved do not need support for particular disk structures.
- The copy is strictly identical to the original, so there is no risk of making a copy where some specific structures have not been properly preserved.
- An entire disk can be copied to or from an image file in a single

operation using a simple command.

This approach also has disadvantages:

- It does not differentiate sections that contain data from those that do not, so everything is copied. Copying sections of the disk which are not in use causes the operation to take more time. Also, the copy uses more space if the destination is an image file, compared to logical copies where empty sections are skipped.
- It is impossible to copy a disk to a smaller disk, as the data at the end of the disk would be lost.

This physical approach is suitable for copying internal disks that contain an operating system, as preserving disk structures is important to keep systems working. Other approaches are more relevant for copying data files.

Exclusive access

Important: A disk must not be in use when it is being copied. Nothing apart from the program performing the copy should be accessing the disk while it is being copied. File systems located on the disk must not be mounted during the operation as files and internal structures could be modified. This is why file systems are not mounted automatically in SystemRescue. That approach gives you more control over which disks are being mounted so you can create or restore physical copies of your disks in a safe way.

Please note that nothing prevents you from copying a disk while it is in use. This would not necessarily cause any error, but the copy would be inconsistent and you would probably not notice the problem until you attempt to restore the copy.

SystemRescue must be run either from a removable medium such as a USB memory stick or from a copy in memory while you perform such a copy. This way, it does not rely on the disk that is being copied during the operation. It is recommended that you run SystemRescue from a copy in memory if you have enough resources. This can be achieved by selecting the entry to boot and copy the system to memory in the boot menu, as explained in Chapter 3: Starting SystemRescue.

The dd command

Physical disks, partitions and files are all seen by the operating system as sequences of bytes. This makes it possible to copy a physical disk or a partition to a file for backup and recovery purposes. The system reads or writes data by sectors that are typically 512 or 4096 bytes, depending on the type of disk. Physical copies are created by using the `dd` program, which is widely available on Linux systems. It must be run from the command line in a terminal. This program reads blocks of bytes sequentially from the input, and writes these blocks to the output. Both the input and output can be a file or a device such as a hard drive.

Here are three typical use cases:

- If you provide the name of a disk as the input and an ordinary file as the output, this effectively creates a new output file as an exact copy of the entire disk. The new file is an image of the disk, and it can be used as a backup that can be restored in the future.
- If you provide an image file as the input and the name of a disk as the output, then the disk is restored from the image file previously created.
- If you provide the names of two disks as input and output, the contents of the disk specified as input are copied to the disk specified as output. The destination disk must be at least as large as the source disk so all the data can fit.

The standard `dd` program works well when the devices do not have any bad sectors. In case of read or write errors while accessing the disk, it is recommended that you use GNU `ddrescue` instead of the standard `dd` command as `ddrescue` has been designed to cope with such situations. It works quite differently from `dd`, and this is outside the scope of the current use case, so it is not treated here. To understand how to use it, you must do some additional research.

Creating an image file of a disk

Here are the steps for copying a disk to an image file. As an example, we are going to copy an internal 500GB disk where the operating system is installed. The copy will be written to an external 1TB disk. The file system on the destination disk must have enough free space to store the image file, which will be as big as the disk being copied.

First, boot SystemRescue so the disk being copied is not in use during the operation. Then start the graphical environment and open a terminal to be ready to execute commands.

Next, you identify the two disks involved in the operation using the `lsblk` command in a terminal. In the current example, the system sees the disk to be copied as `/dev/sdc`, and it contains six partitions. The copy will be written to a file stored in a file system on `/dev/sdd1` which is the first partition of the external disk.

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,model
PATH         SIZE FSTYPE      LABEL     MODEL
/dev/sdc   465.8G                       Samsung_SSD_850_EVO_mSATA
/dev/sdc1     4G vfat        UEFI_ESP
/dev/sdc2   5.2G btrfs       boot
/dev/sdc3 406.5G crypto_LUKS
/dev/sdc4    16M
/dev/sdc5    48G ntfs        Windows
/dev/sdc6     2G vfat        SOFTWARE
/dev/sdd   931.5G                       SABRENT
/dev/sdd1 931.5G ext4        external
```

If your external disk does not contain a file system, please refer to Chapter 9: Partitioning Disks to understand how to initialize the disk. The file system where the image file will be stored must be able to store large files. Modern file systems such as EXT4, XFS, BTRFS and NTFS can all cope with large files. However, FAT16 and FAT32 file systems are very limited and unreliable, so you should not store such image files on them.

Now you should create a new mount point which we call `external` so you can mount the external disk, as shown in the following two commands in a terminal:

```
[root@sysrescue ~]# mkdir -p /mnt/external
[root@sysrescue ~]# mount -v /dev/sdd1 /mnt/external
mount: /dev/sdd1 mounted on /mnt/external.
```

You can use the `df` command to check the type of file system it uses and to make sure it has enough space to store the image file:

```
[root@sysrescue ~]# df -hT /mnt/external
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/sdd1      ext4  916G   72M  870G   1% /mnt/external
```

Before you start the copy, make sure the disk being copied is not mounted so the copy is consistent. You do this by showing the mount points for this device using `lsblk` with the following options:

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,mountpoint /dev/sdc
PATH          SIZE FSTYPE      LABEL     MOUNTPOINT
/dev/sdc   465.8G
/dev/sdc1      4G vfat        UEFI_ESP
/dev/sdc2    5.2G btrfs       boot
/dev/sdc3  406.5G crypto_LUKS
/dev/sdc4     16M
/dev/sdc5     48G ntfs        Windows
/dev/sdc6      2G vfat        SOFTWARE  /mnt/software
```

In the example above `lsblk` shows that the sixth partition of the disk is mounted on `/mnt/software`, so this must be unmounted before you start copying `/dev/sdc`:

```
[root@sysrescue ~]# umount /dev/sdc6
```

You should now be ready to perform the copy. In a typical situation, the `dd` command needs two arguments: `if` and `of`. They represent the input and output files or devices. The input is the name of the device that corresponds to the entire internal disk. In the current example, it is `/dev/sdc`. The output is the name of the new image file located on the external disk. As the external disk is mounted on `/mnt/external` the full path of the image file could be something such as `/mnt/external/disk-image`. It is recommended that you give a quite specific name to your image file and that this name includes a date so you can identify which disk or computer it corresponds to and when it was created. Also, you should use the `dd` command with the `status=progress` option so you can follow the progress of the operation. Here is the full command that performs the copy in the current example:

```
[root@sysrescue ~]# dd if=/dev/sdc of=/mnt/external/disk-image status=progress
976773168+0 records in
976773168+0 records out
500107862016 bytes (500 GB, 466 GiB) copied, 11040.3 s, 45.3 MB/s
```

It is important to be very cautious when you run this command as it can be destructive if it is used incorrectly. If you provide the wrong output as the `of` parameter all data on the output file or device will be overwritten.

This command is likely to take a long time to complete as disks are often quite large. You must make sure there is no error message on the program output as this might be an indication that the image has not been created successfully.

The example above shows what you can expect to see when the command is successful. It shows the number of bytes which have been copied, the size in human-readable units and the number of blocks it corresponds to.

When the operation is complete, you should check that the size of the new image file is identical to the size of the disk you copied. You can use the `blockdev` command to get the exact size of the disk in bytes, and then the `du` command to show the exact size of the image file:

```
[root@sysrescue ~]# blockdev --getsize64 /dev/sdc
500107862016

[root@sysrescue ~]# du -b /mnt/external/disk-image
500107862016    /mnt/external/disk-image
```

After you have checked that everything is correct, run `sync` and unmount the external disk to make sure all write operations have been completed before you disconnect the external disk.

```
[root@sysrescue ~]# sync
[root@sysrescue ~]# umount /mnt/external
```

Restoring a disk from an image file

In the previous section, we created a copy of a disk as an image file that can be used as a backup of the entire disk. In this section, we do the exact opposite: we restore an image file to a disk. Restoring a disk from a backup is useful when an issue affects your system and it cannot be resolved easily. This can happen ,for example, if an operating system update was faulty and the system has become unusable. You may also want to copy the contents of an image file to a new disk if it is replacing an old disk that failed or as part of a hardware upgrade.

The operation in this section is destructive. All the disk contents will be replaced by the contents of the image file. So you must be sure that all important data on the target disk have been copied to a safe place before you start this operation. If you must recover data files from a disk, you can use the instructions from Chapter 10: How to Recover Files to Another Disk or Chapter 11: How to Recover Files Over a Network.

As usual, you start by booting SystemRescue so you can operate on the internal disk without it being used by the operating system. Use the `lsblk` command to identify the name of the internal disk which must be restored (`/dev/sdc` in the current example) and the name of the external disk that contains the image file (`/dev/sdd1` in this example).

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,mountpoint
PATH         SIZE FSTYPE      LABEL     MOUNTPOINT
/dev/sdc   465.8G
/dev/sdc1      4G vfat        UEFI_ESP
/dev/sdc2    5.2G btrfs       boot
/dev/sdc3  406.5G crypto_LUKS
/dev/sdc4     16M
/dev/sdc5     48G ntfs        Windows
/dev/sdc6      2G vfat        SOFTWARE
/dev/sdd   931.5G
/dev/sdd1  931.5G ext4        external
```

The file system which contains the image file must be mounted on a new mount point, which we call `external`:

```
[root@sysrescue ~]# mkdir -p /mnt/external
[root@sysrescue ~]# mount /dev/sdd1 /mnt/external
```

Pay attention to all the details shown by the `lsblk` command, such as the disk size, partitions and labels to be sure you recognise the correct disk. This is a destructive operation, as all data on the target disk will be overwritten. Also, make sure the disk being overwritten does not contain any mounted file system, as the `dd` program must have exclusive access to the disk during the operation. If necessary, you must unmount these before you continue.

You should check that the target disk is at least as big as the disk image, otherwise, some data will be missing on the target disk. As before, use the `blockdev` and `du` commands to check the sizes of the disk and image file in bytes:

```
[root@sysrescue ~]# du -b /mnt/external/disk-image
500107862016      /mnt/external/disk-image

[root@sysrescue ~]# blockdev --getsize64 /dev/sdc
500107862016
```

Finally, run the `dd` command to copy the contents of the image file to the disk. In this case, the input specified using `if` is the path to the image file, and the output specified with `of` is the target disk where the image must be restored. Make sure that you have specified the correct output before you press `Enter`, as all data on this device will be overwritten.

```
[root@sysrescue ~]# dd if=/mnt/external/disk-image of=/dev/sdc status=progress
```

It is important that you read the messages printed on the screen to make sure there is no error during the operation.

After the operation, you can use the `lsblk` command again to see if the disk contains the partitions you expected.

You may have restored an image to a disk that is bigger than the image file. In that case, the additional space on the new disk is unallocated. You can use partitioning tools to grow existing partitions and file systems or you can create new partitions to take advantage of this space. To know more about disk partitioning, read Chapter 9: Partitioning Disks.

Accessing a single partition in an image file

In the previous example, the entire disk was restored from an image file. Sometimes you want to restore only a single partition from an image file. For example, your operating system might have stopped working. In that case, you want to restore the contents of the partition where the operating system is installed from an image file created when the operating system was working correctly. In such a case, you do not want other partitions to be affected by this operation as they may contain data files which you want to preserve.

The image file was created as a copy of an entire disk, so it contains the contents of each partition at the time the image file was created. It is possible to copy a subset of the image to restore the contents of an individual partition. This operation might not work if the partitioning of the disk has been changed since the image was created.

Please follow the instructions from the previous section to identify the disks involved and to mount the disk that contains the image file that you want to partially restore.

Here, an extra step is required to access the contents that correspond to a specific partition in the image file. In theory, it is possible to manually analyse the partition table and calculate where the contents of a specific partition start and end in the image file, but this is risky.

To solve this problem, we get the system to create a virtual disk device from the image file so the image can be treated as an ordinary disk. As the image is a copy of a disk, it also contains a partition table. The system analyses it and creates virtual devices that correspond to each partition in the image file. This way the contents of the partition you want to restore from the image are easily accessible.

The `losetup` command lets you create a virtual disk device based on the contents of an image file. This type of virtual device is called a loop device. Each time data are read from the virtual device, the system actually gets the contents from the image file associated with the loop device. The device will be created in read-only mode to avoid accidental changes in the image file. Here is the full command that creates a virtual loop device from the image file located on the external disk:

```
[root@sysrescue ~]# losetup --read-only -P --find --show /mnt/external/disk-
image
/dev/loop1
```

The command prints the name of the newly created virtual device, such as `/dev/loop1` in this example. The system may use other devices, such as `/dev/loop2`, depending on the circumstances. You can use `lsblk` to show the list of partitions which this new virtual disk contains. The partitions listed should be identical to those on the disk at the time you created the image file. Do not forget to replace `/dev/loop1` with the name that is relevant to your situation:

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label /dev/loop1
PATH            SIZE FSTYPE      LABEL
/dev/loop1   465.8G
/dev/loop1p1     4G vfat        UEFI_ESP
/dev/loop1p2   5.2G btrfs       boot
/dev/loop1p3 406.5G crypto_LUKS
/dev/loop1p4    16M
/dev/loop1p5    48G ntfs        Windows
/dev/loop1p6     2G vfat        SOFTWARE
```

In this example, we can see the virtual device contains six partitions. We restore the contents of the fifth partition from the image file as this is the partition that contains the operating system that must be restored. The device which corresponds to this partition in the image file is `/dev/loop1p5` in our example. You can confirm that the size of this partition in the image file is identical to the size of the fifth partition on the target disk:

```
[root@sysrescue ~]# blockdev --getsize64 /dev/loop1p5
51521781760

[root@sysrescue ~]# blockdev --getsize64 /dev/sdc5
51521781760
```

Now you can use the `dd` command to restore the contents of the fifth partition from the partition on the virtual device. In the current example, the input is `/dev/loop1p5` and the output is `/dev/sdc5`. As usual, please double check that you are using the right input and output devices in your situation, as all contents on the output device will be overwritten.

```
[root@sysrescue ~]# dd if=/dev/loop1p5 of=/dev/sdc5 status=progress
```

At the end of the operation, the contents of this partition should be identical to the contents of this partition at the time the image was created, and other partitions should not have been affected by this operation. Now you can detach the virtual loop device before you can unmount the external disk, using the following commands:

```
[root@sysrescue ~]# losetup --detach /dev/loop1
[root@sysrescue ~]# umount /mnt/external
```

These virtual loop devices could also be mounted if you want to restore individual files from the image, instead of restoring an entire partition. You can refer to Chapter 10: How to Recover Files to Another Disk for more details about how to copy files from one file system to another.

Alternative approaches

The instructions above were focused on physical copies. This approach was chosen because it is simple and it covers many essential use cases. The process could be improved in many ways. For example, physical images could have been compressed to save storage space. But this would have made it impossible to use virtual loop devices to directly access partitions inside the image. Also, compression would cause large parts of the image to become unusable in case the image file becomes corrupted. So there are good reasons to keep images files uncompressed, and that should not be a problem considering large external disks are quite affordable.

There are also many alternative approaches for copying disks, and different people have different needs and preferences. Other tools such as `partclone` are included in SystemRescue so you can consider them to see if they suit you better.

In any case, it is recommended that you create an image of the internal disk when the operating system and applications have all been installed and configured. Being able to restore a working backup of a disk can save a lot of time and effort when something goes wrong.

Chapter 13: How to Fix Linux Systems Using chroot

SystemRescue comes with chroot, which can help fix many issues affecting Linux operating systems. It is particularly useful in the following situations:

- The Linux system can boot, but the password to authenticate on the system has been lost.
- The boot loader, such as Grub, which starts Linux has stopped working and must be reinstalled.
- The system boot process fails because of issues related to the Linux kernel or the initramfs image.

Why chroot might be needed

Before we dive into the details, it is important to understand why chroot may be required. Most Linux systems are started by a boot loader, such as Grub, which loads two files: the Linux kernel image file (often named vmlinuz) and the initramfs image file which provides additional kernel modules, initialisation programs and configuration files. Lots of things can go wrong with these two files to keep the system from booting properly. For example, the kernel might not be able to boot after a faulty update (this is quite rare). The system might also fail to boot if essential configuration files have been modified and those changes cause the start-up not to work with the new version of the initramfs file.

Such issues can often be resolved by running commands on the affected Linux system. For example, the kernel package could be updated to fix a bug that broke the system. Also, invalid configuration changes can be reverted, and a command can be executed to recreate the initramfs file using a new working configuration. As an example, the command which rebuilds an initramfs image is `update-initramfs` on Debian, `dracut` on Red Hat, and `mkinitcpio` on Arch Linux, so such commands are often distribution-specific. Also, these often rely on configuration files which are part of the broken system. Hence the commands to fix a broken Linux system often have to be executed from this specific system.

This problem is you need access to the broken system to run commands that can fix the issue. One way to sort this out is to boot SystemRescue, mount the root file system of the affected Linux system, and then use chroot to switch to the context of this broken system. This way, you can execute recovery commands as if you had been able to boot the broken system.

How chroot works

Chroot changes the effective root directory of the programs executed in the current terminal. In other words, these programs see the new directory as the root directory of the system. This allows you to run commands in the context of another Linux operating system as if the computer had directly started from that system. Chroot affects only programs executed in the current terminal, as it changes only the effective root directory of the current shell and all programs executed from that shell. All other programs continue to work in the original context.

For example, let's say your Linux system is installed on a local disk and you cannot open a session because you have forgotten your password. The procedure to follow is to boot SystemRescue, identify and mount the root file system of the local Linux system on a mount point such as `/mnt/linux`. Then use chroot to make this mount point the new effective root directory and execute the `passwd` command to reset the password in the chroot environment. The `passwd` command updates passwords stored in `/etc/shadow`. As the command runs in the chroot environment, the change actually takes place in `/mnt/linux/etc/shadow` which belongs to the affected Linux system. If the password command had been used outside of the chroot environment, it would instead have updated the password in `/etc/shadow` which is part of SystemRescue.

Many administration commands rely on virtual file systems such as `/dev`, `/proc` and `/sys`. Hence, these virtual file systems must be made available in the chroot environment for commands to work properly. The `arch-chroot` script takes care of setting up all these virtual file systems before it changes the effective root. This is why you should access chroot environments using `arch-chroot` instead of using the standard `chroot` command.

Detailed example

As an example, here are the steps to follow to change a password on a Linux operating system using chroot from SystemRescue. As usual, start by booting SystemRescue, then go to the graphical environment and launch a terminal.

The device where the non-working Linux system is installed must then be identified using `lsblk`:

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,model
PATH          SIZE FSTYPE      LABEL    MODEL
/dev/sdc    465.8G                      Samsung_SSD_850_EVO_mSATA
/dev/sdc1       4G vfat        UEFI_ESP
/dev/sdc2     5.2G btrfs       boot
/dev/sdc3   406.5G crypto_LUKS
/dev/sdc4      16M
/dev/sdc5      40G ntfs        Windows
/dev/sdc6       2G vfat        SOFTWARE
/dev/sdc7       8G btrfs       Fedora
```

In this example, the device corresponding to the Fedora Linux root file system has been identified as `/dev/sdc7`.

Next, create a mount point named `linux` and mount the Fedora Linux root file system there:

```
[root@sysrescue ~]# mkdir -p /mnt/linux
[root@sysrescue ~]# mount /dev/sdc7 /mnt/linux
```

Now you are ready to use `arch-chroot` to change the current context:

```
[root@sysrescue ~]# arch-chroot /mnt/linux
```

You can confirm that the context has been successfully changed as files belonging to Fedora are accessible without having to specify the mount point as a prefix.

```
[root@sysrescue /]# cat /etc/fedora-release
Fedora release 33 (Thirty Three)
```

However, the prompt still shows `sysrescue` because the hostname in memory has not changed. We can then run the commands to fix the Linux system files on the disk. In the current example we reset the password of the `root` user account:

```
[root@sysrescue /]# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

After all changes have been made, type `exit` in the current terminal to quit the chroot environment. Finally, you unmount the Linux file system. Please note that the recursive option is required so all additional file systems, which were mounted by `arch-chroot`, are unmounted automatically.

```
[root@sysrescue /]# exit
[root@sysrescue ~]# umount --recursive /mnt/linux
```

Conclusion

Chroot is very useful for accessing Linux operating systems installed on the disk from SystemRescue. It can be used in many situations to solve various types of problems affecting such systems. You must have a good understanding of a problem to be able to fix it. You should be able to determine the cause of the problem by carefully reading any error messages displayed during the boot process. You may have to do more research on the internet to understand how to fix a particular type of problem on a particular Linux distribution.

Please note: chroot will fail if you use a 32-bit version of SystemRescue to access a 64-bit system. But you could use chroot from a 64-bit version of SystemRescue to access a 32-bit system. This is one of the reasons it is recommended that you always use the 64-bit version of SystemRescue unless your hardware does not support it.

Chapter 14: How to Reset a Local Windows Password

This chapter guides you through the process of resetting a local user password on a Windows system. You must reset such a password if you have forgotten it and you cannot sign in. The current approach is applicable only for passwords stored on the local disk. It won't work if you sign in using an online Microsoft account.

SystemRescue comes with `chntpw`. This program can modify the `SAM` file, where Windows stores local passwords. You execute the following steps in a terminal from SystemRescue to access the Windows system disk. Then, you use `chntpw` to clear the password associated with your user account.

Identifying the Windows disk

The first step is to identify the partition where Windows is installed using `lsblk` in a terminal:

```
[root@sysrescue ~]# lsblk -o path,size,fstype,label,model
PATH        SIZE FSTYPE      LABEL    MODEL
/dev/sdc  465.8G                      Samsung_SSD_850_EVO_mSATA
/dev/sdc1     4G vfat        UEFI_ESP
/dev/sdc2   5.2G btrfs       boot
/dev/sdc3 406.5G crypto_LUKS
/dev/sdc4    16M
/dev/sdc5    40G ntfs        Windows
/dev/sdc6     2G vfat        SOFTWARE
/dev/sdc7     8G btrfs       Fedora
```

You should be able to identify the Windows partition as it should have an NTFS file system, and you may recognize its size and volume name. In the current example, Windows is installed on the device identified as /dev/sdc5.

Mounting the Windows disk

Next, create a mount point and name it windows:

```
[root@sysrescue ~]# mkdir -p /mnt/windows
```

The Windows file system must be mounted using the device name previously identified, such as /dev/sdc5 in this example. The full read-write access mode is used so the files where passwords are stored can be modified.

```
[root@sysrescue ~]# mount -v /dev/sdc5 /mnt/windows
mount: /dev/sdc5 mounted on /mnt/windows.
```

Make sure you can find the SAM file. The find command shows a list of all files with such a name on this file system:

```
[root@sysrescue ~]# find /mnt/windows -name SAM
/mnt/windows/Windows/System32/config/SAM
/mnt/windows/Windows/SysWOW64/LogFiles/SAM
```

We are interested in the SAM file located in the System32/config directory. If this file cannot be found, you might have used the wrong device. In that case, go back to the first step.

Clearing the password

Once the Windows file system and the SAM file are accessible, you are ready to manipulate local Windows accounts using the chntpw command. First, show the list of local user accounts by specifying option `-l` with this command followed by the path to the SAM file which was found in the previous step:

```
[root@sysrescue ~]# chntpw -l /mnt/windows/Windows/System32/config/SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive </mnt/windows/Windows/System32/config/SAM> name (from header):
<\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 686c <lh>
File size 65536 [10000] bytes, containing 7 pages (+ 1 headerpage)
Used for data: 318/32040 blocks/bytes, unused: 30/12792 blocks/bytes.

| RID -|---------- Username ------------| Admin? |- Lock? --|
| 01f4 | Administrator                  | ADMIN  | dis/lock |
| 01f7 | DefaultAccount                 |        | dis/lock |
| 03e9 | francois                       | ADMIN  |          |
| 01f5 | Guest                          |        | dis/lock |
| 01f8 | WDAGUtilityAccount             |        | dis/lock |
```

You should recognize the name of your local Windows user account in the list. Now let's use `chntpw` with the `-u` option to specify the name of the user account which needs a password reset, followed by the path to the `SAM` file:

```
[root@sysrescue ~]# chntpw -u francois /mnt/windows/Windows/System32/config/SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive </mnt/windows/Windows/System32/config/SAM> name (from header):
<\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 686c <lh>
File size 65536 [10000] bytes, containing 7 pages (+ 1 headerpage)
Used for data: 318/32040 blocks/bytes, unused: 30/12792 blocks/bytes.

================ USER EDIT ====================

RID     : 1001 [03e9]
Username: francois
fullname:
comment :
homedir :

00000220 = Administrators (which has 2 members)

Account bits: 0x0214 =
[ ] Disabled        | [ ] Homedir req.    | [X] Passwd not req. |
[ ] Temp. duplicate | [X] Normal account  | [ ] NMS account     |
[ ] Domain trust ac | [ ] Wks trust act.  | [ ] Srv trust act   |
[X] Pwd don't expir | [ ] Auto lockout    | [ ] (unknown 0x08)  |
[ ] (unknown 0x10)  | [ ] (unknown 0x20)  | [ ] (unknown 0x40)  |

Failed login count: 0, while max tries is: 0
Total  login count: 92

- - - - User Edit Menu:
 1 - Clear (blank) user password
(2 - Unlock and enable user account) [seems unlocked already]
 3 - Promote user (make user an administrator)
 4 - Add user to a group
 5 - Remove user from a group
 q - Quit editing user, back to user select
```

As explained in the instructions on the screen, you now press `1` + `Enter` to clear the user password, then `q` + `Enter` to quit the menu. Finally press `y` + `Enter` to confirm you want to save the modifications made in the `SAM` file.

Unmounting the Windows disk

When you have completed the changes, you should unmount the Windows disk before you can shut down or reboot the computer.

```
[root@sysrescue ~]# umount /mnt/windows
```

Setting a new password

After the reboot you should be able to sign in on Windows using your local account without any password. You can then press `Ctrl+Alt+Delete` and select `change a password` to set a new password on your account.

Appendix 1: Boot Parameters

Here are the most common boot parameters. Please refer to Chapter 3: Starting SystemRescue for more details about the main optional boot parameters.

Mandatory boot parameters

- **archisolabel=xxxx** defines the volume name of the file system from which SystemRescue loads its files during the initialisation.
- **archisobasedir=xxxx** defines the base directory where all SystemRescue files reside. This option allows you to install SystemRescue in a custom directory.

Optional boot parameters

- **copytoram** causes SystemRescue to be fully loaded into memory. This option is recommended unless your computer does not have enough memory.
- **checksum** triggers a verification of the integrity of the boot image. You should use this boot option if you get unexpected errors when running SystemRescue.
- **setkmap=xxxx** configures the keyboard layout using a specific keyboard map code (such as `de` for German keyboards). Run `setkmap` in a terminal to get a list of all supported layouts.
- **nofirewall** disables the firewall so you can establish connections to the SystemRescue live system from outside.
- **nomodeset** runs the system with a basic lower resolution display driver. Use this option if the default display settings are not working.

Appendix 2: How to Customise Boot Entries

It is possible to customise the SystemRescue entries in the boot menus if it has been installed on a writable file system such as FAT32 on a writable device. This is what you get if you install SystemRescue using Rufus in ISO mode as described in Chapter 2: Preparing the Boot Device. These customisations allow you to boot SystemRescue using your preferred boot options without having to modify these entries each time you use SystemRescue. For example, you can use this to automatically configure the keyboard layout when you start SystemRescue from the default entry in the menu. The two boot menus can be modified by editing the following two files which are located on the SystemRescue boot device: `sysresccd_sys.cfg` and `grubsrcd.cfg`. Please refer to Chapter 3: Starting SystemRescue for more details about how boot entries work and Appendix 1: Boot Parameters for a list of all common boot parameters.

BIOS boot menu

As an example, here is the definition of the first entry in the Legacy BIOS boot menu in `sysresccd_sys.cfg`:

```
LABEL sysresccd
TEXT HELP
Boot the SystemRescue live medium.
ENDTEXT
MENU LABEL Boot SystemRescue using default options
LINUX boot/x86_64/vmlinuz
INITRD boot/intel_ucode.img,boot/amd_ucode.img,boot/x86_64/sysresccd.img
APPEND archisobasedir=sysresccd archisolabel=RESCUE700
```

Customizations can be made on the line starting with the `APPEND` keyword. Let’s say you use a French keyboard and you always want to copy the contents of the system into memory during the boot. You could add the following two parameters at the end of this line:

```
setkmap=fr-latin1 copytoram
```

UEFI boot menu

Here is the definition of the default boot entry in the UEFI boot menu in `grubsrcd.cfg`:

```
menuentry "Boot SystemRescue using default options" {
        set gfxpayload=keep
        linux /sysresccd/boot/x86_64/vmlinuz archisobasedir=sysresccd
archisolabel=RESCUE700
        initrd /sysresccd/boot/intel_ucode.img /sysresccd/boot/amd_ucode.img
/sysresccd/boot/x86_64/sysresccd.img
```

This time, changes should be made on the line starting with `linux`. The path to the `vmlinuz` kernel image file must remain in its original location, just after the `linux` keyword. You can add more boot parameters at the end of the line.

Warning

Please make sure all parameters are separated by spaces and that you do not insert new line breaks. Also, make sure you do not change the value of the `archisolabel` parameter without understanding the implications. This value changes with each version of SystemRescue. It is used to identify the device where SystemRescue is installed during the boot process. Hence the value of this parameter must match the volume name of the device where it is installed. Otherwise, SystemRescue will fail to boot.

Appendix 3: Essential Commands

Here is a summary of essential commands which have been used throughout the various chapters. You can refer to this appendix to remind yourself which command to use for a particular use case. Please refer to the corresponding chapters if you need more context and details about what these commands do.

Documentation

Most commands come with a manual page which provides detailed documentation. These are accessible through the `man` command. For example, you can run `man dd` in a terminal to learn more about the `dd` command. Also most commands display a summary of each supported option when you execute the command followed by `--help` (eg: `lsblk --help`)

General commands

- `cd /path/to/directory` changes the current directory in the current shell
- `pwd` shows the path to the working directory in the current shell
- `ls -lh /path/to/directory` to show the contents of a directory on the file system
- `cat /path/to/file` shows the contents of a file
- `sha256sum /path/to/file` calculate the SHA256 checksum of a file
- `find /path/to/directory -name "filename"` searches for files with a particular name located in a particular directory

Controlling the firewall

- `systemctl is-active iptables ip6tables` shows if the firewall services are currently active
- `systemctl stop iptables ip6tables` stop the firewall services on the current system
- `iptables -S` and `ip6tables -S` shows the firewall policies and rules currently effective on the system

Identifying disks and partitions

- `lsblk -o path,size,fstype,label,model` shows the list of all disks and partitions as well as their main attributes.

Mounting a file system

- `mkdir -p /mnt/mount_point` creates a new directory which you can use as a mount point.
- `mount -v /dev/device_name /mnt/mount_point` mounts a file system on a directory. (Add `-o ro` at the end to mount it in read-only mode.)
- `umount /mnt/mount_point` unmounts the file system from a particular mount point.
- `df -hT /mnt/mount_point` shows how much disk space is available on a mounted file system.

Transferring files over the network

- `systemctl is-active sshd` shows if the sshd service is currently active.
- `systemctl restart sshd` restarts the sshd service on the current system.

- `systemctl stop sshd` stops the sshd service on the current system.
- `passwd root` changes the password of the root user account on the current system.
- `journalctl -f --unit=sshd` shows messages written by the sshd service.

Physical disk copies

- `blockdev --getsize64 /dev/device_name` shows the exact size of a disk or partition.
- `du -b /path/to/file` shows the exact size of a file.
- `dd if=input of=output status=progress` sequentially copies all data from the input to the output.
- `losetup --read-only -P --find --show /path/to/disk-image` creates a virtual loop device from a disk image file.
- `losetup -d /dev/loop_device` detaches a loop device from its image file.

Chroot

- `arch-chroot /mnt/mount_point` changes the effective root directory to a mounted file system.
- `umount --recursive /mnt/mount_point` unmounts a mounted file system and all recursive file systems.