7 Under the hood

7.1 Introduction

MX Linux ultimately inherits its fundamental design from Unix, an operating system that has been around in various forms since 1970, much earlier than MS-Windows. From that Linux was developed, from which Debian develops its distribution. The base operating system is the topic of this section. Users coming from Microsoft Windows typically find a lot of unfamiliar concepts, and get frustrated trying to do things the way they are accustomed to doing them.

This section will give you a conceptual overview of some basic aspects of MX Linux, and how they differ from other systems to help ease your transition.

Links

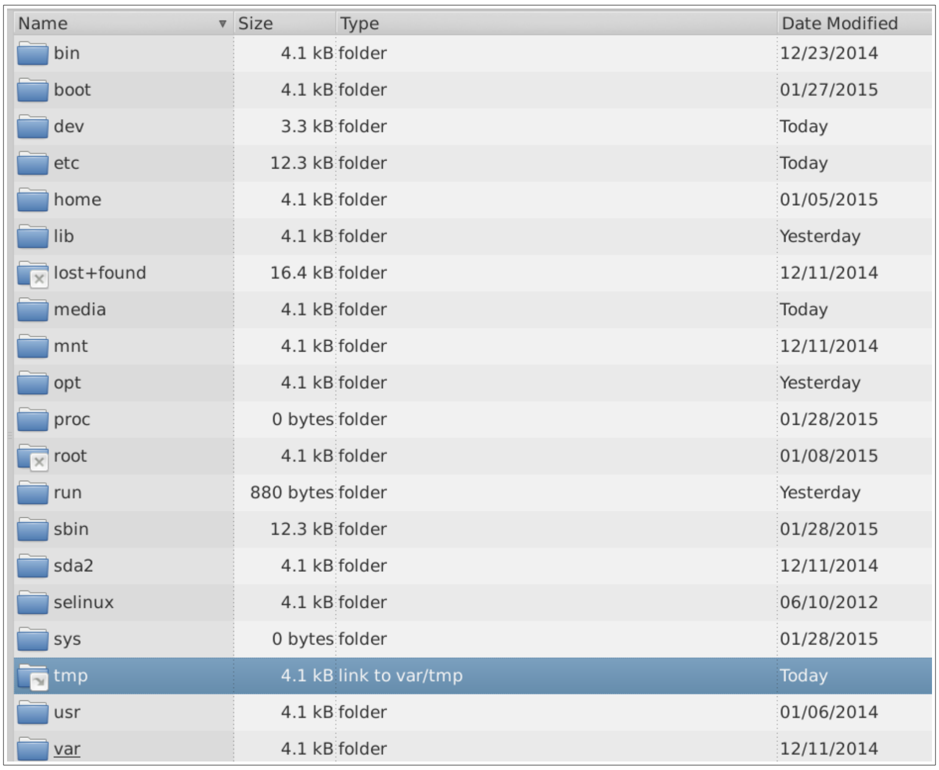
- Wikipedia: Unix
- Linux Home Page
- Wikipedia Debian

7.2 The file system structure

There are two basic uses of the term "file system". The first is the Operating System's Filesystem. This refers to the files and their organization that the operating system uses to keep track of all the hardware and software resources it has as its disposal while running.
The other use of the term file system refers to the Disk Filesystem, designed for the storage and retrieval of files on a data storage device, most commonly a disc drive. The Disk Filesystem is set when the disk partition is first formatted, prior to writing any data on the partition.

The Operating System's Filesystem

One of the first problems many new Linux users struggle with is how the file system works. If you have been looking around your MX Linux system trying to find the C:\ drive or D:\ drive, for instance, you are searching in vain: MX Linux handles hard drives and other storage media differently from Windows. Rather than having a separate file system tree on every device, MX Linux has a single file system tree (called the /root/ of the file system) which is marked "/" and contains every attached device. When a storage device is added to the system, its file system is attached to a directory or subdirectory of the file system; this is called mounting a drive or device. If you open Thunar and click on File System in the upper left pane, you will notice a number of directories with names based on the Unix Filesystem Hierarchy Standard.



Name	Size	Type	Date Modified
bin	4.1 kB	folder	12/23/2014
boot	4.1 kB	folder	01/27/2015
dev	3.3 kB	folder	Today
etc	12.3 kB	folder	Today
home	4.1 kB	folder	01/05/2015
lib	4.1 kB	folder	Yesterday
lost+found	16.4 kB	folder	12/11/2014
media	4.1 kB	folder	Today
mnt	4.1 kB	folder	12/11/2014
opt	4.1 kB	folder	Yesterday
proc	0 bytes	folder	01/28/2015
root	4.1 kB	folder	01/08/2015
run	880 bytes	folder	Yesterday
sbin	12.3 kB	folder	01/28/2015
sda2	4.1 kB	folder	12/11/2014
selinux	4.1 kB	folder	06/10/2012
sys	0 bytes	folder	01/28/2015
tmp	4.1 kB	link to var/tmp	Today
usr	4.1 kB	folder	01/06/2014
var	4.1 kB	folder	12/11/2014

Figure 7-1: The MX filesystem viewed in Thunar

Here is a simple description of the major directories in MX Linux along with an example of when users commonly work with files in those directories:

- /bin
  - This directory contains binary program files which are used by the system during startup, but which also may be required by user actions once the system is fully up and running.
  - Example: Many basic command-line programs, such as the Bash shell, and utilities like /dd/, /grep/, /ls/, and /mount/ are located here, in addition to programs only used by the OS.
- /boot
  - As you might guess, files that Linux needs to boot are located here. The Linux kernel, the core of the Linux operating system, is kept here, as are bootloaders such as GRUB.
  - Example: no file here is commonly accessed by users.
- /dev
  - In this directory are special files that link to the various input/output devices on the system.
  - Example: no file here is commonly accessed directly by users, except in CLI mounting commands.
- /etc This directory contains configuration files for the system such as well as application configuration files.
  - Example: The file /etc/fstab specifies mount points for additional filesystems on devices, partitions, etc. that can be configured for your optimal use.

    - Example: display problems sometimes involve editing the file /etc/X11/xorg.conf.
- /home
    - Here the user’s personal directories (data and settings) reside. If there is more than one user, a separate subdirectory is set up for each. No user (except root) can read another user’s home directory. The user’s directory contains both hidden (where the filename is preceded by a dot) and visible files; hidden files can be revealed by clicking View >Show Hidden Files in Thunar.
    - Example: users typically organize their own files at first by using default directories such as Documents, Music, etc.
    - Example: your Firefox profile is located in the hidden directory .mozilla/firefox/
- /lib
    - This directory contains shared object libraries (analogous to Windows DLL’s) that are required at boot time. In particular, kernel modules will be found here, under /lib/modules.
    - Example: no file here is commonly accessed by users.

- /media
    - Files for removable media such as CDroms, floppy drives, and USB memory sticks are installed here when the media are automounted.
    - Example: After dynamically mounting a peripheral device like a flash drive, you may access it here.
- /mnt
    - Physical storage devices must be mounted here before they can be accessed. After drives or partitions are defined in the file /etc/fstab, then their file system is mounted here.
    - Example: Users can access drives and partitions mounted here.
- /opt
    - This is the intended location of major third-party application subsystems installed by the user.
    - Example: if you install Google Earth, this is where it will be installed. Some distros also place user-installed programs in /usr subdirectories.
- /proc
    - The location for process and system information
    - Example: no file here is commonly accessed by users
- /root
    - This is the home directory for the root user (administrator). Note that this is not the same as “/” the file system root.
    - Example: no file here is commonly accessed by users, but files saved while logged in as the root user may be saved here.

    - Programs are installed here if they are required by the system startup scripts but will not normally be run by users, other than root—in other words, system administration utilities.
    - Example: no file here is commonly accessed by users, but this is where files like modprobe and ifconfig are located.
- /tmp
    - This is the location of temporary files produced by programs—such as compilers—as they run. In general, these are short-term temporary files, of use to a program only while it is running.
    - Example: no file here is commonly accessed by users.
- /usr
    - This directory contains many things for user applications, and is analogous in some ways to the Windows directory "Program Files".
    - Example: many executables are located in (/usr/bin)
    - Example: documentation (/usr/docs) and configuration files, graphics and icons are in (/usr/share).
    - Example: many files specific to MX Linux are located in /usr/local/
- /var
    - This directory contains files that are constantly changing while Linux is running, e.g. logs, system mail and queued processes.
    - Example: you can look in /var/log/ when trying to determine what happened during a process

The Disk Filesystem

The disk file system is something about which the average user does not need to be much concerned. The default disk file system used by MX Linux is called ext4, a version of the ext2 file system that is journaled —i.e., it writes changes to a log before enacting them, rendering it more robust. The file system ext4 is set during installation when your hard drive is formatted.

By and large, ext4 has more years on its track record than any of its rivals, and combines stability and speed; for these reasons, we do not recommend installing MX Linux onto a different disk file system unless you are well-educated in the differences. However, MX Linux can read and write to many other formatted disk filesystems, and may even be installed on some of them, if for some reason one of them is preferred over ext4.

Links

- Wikipedia Filesystem
- Wikipedia. Comparison of filesystems
- Wikipedia Ext4
- Log files in the /var/log/ directory

7.3 Permissions

MX Linux is an account-based operating system. This means that no program can run without a user account to run under, and any running program is thereby limited by the permissions granted to the user who started it.

> NOTE: Much of the security and stability that Linux is known for hinges on the proper use of limited user accounts, and the protection provided by default file and directory permissions. For this reason, you should operate as root only for a procedure that requires it. Never log into MX Linux as root to run the computer for normal activities–running a web browser as root user, for instance, is one of the few ways you could get a virus on a Linux system!

Basic information

The default file permissions structure in Linux is fairly simple, but more than adequate for most situations. For each file or folder, there are three permissions that can be granted, and three entities (owner/creator, group, others/world) to which they are granted. The permissions are:

- Read permission means that data can be read from the file; it also means the file can be copied. If you don’t have read permission for a directory you can’t even see the names of files listed in it.
- Write permission means that the file or folder can be changed, appended, or deleted. For directories, it specifies whether a user can write to files in the directory.
- Execute permission means whether or not the user can run the file as a script or program. For directories, it determines whether or not the user can enter and make it the current working directory. Every file and folder acquires a single user designated as its owner when it is created on the system. (Note that if you move a file from another partition where it has a different owner, it will keep the original owner; but if you copy and paste it, it will be assigned to you.) It also has a single group designated as its group, by default the group to which the owner belongs. The permissions you grant to others affect everyone who isn’t the owner or in the owning group.

> NOTE: For advanced users, there are additional special attributes beyond read/write/execute that can be set: sticky bit, SUID, and SGID. For more information, see Links section below.

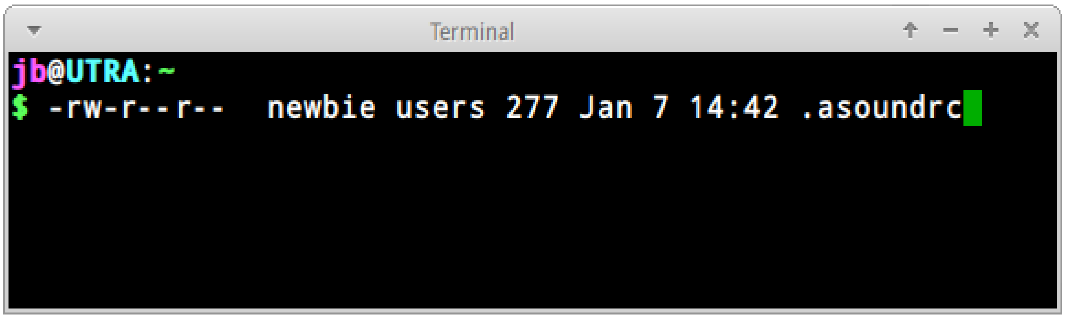
Viewing, setting and changing permissions

There are many tools available in MX Linux to view and manage permissions.

- GUI
    - Thunar To view or change a file’s permissions, right-click the file and select Properties. Click the Permissions tab. Here you can set the permissions granted to the owner, group, and others using the pull-down menus. For some files (like scripts , for instance), you need to check the box to make them executable, and for folders you can check a box to limit the deletion of files inside it to the owners.
    - NOTE: you must be operating as root to change the permission of a file or directory whose owner is root. On bigger folders you MUST REFRESH your Thunar window or else the permissions will show incorrectly, even though the permissions have

actually changed. Just hit F5 to refresh the window or else you will see the original permissions.
    - MX User Manager is an easy way to change permissions by associating a user with specific groups.

- CLI
    - Internal partitions. By default, the root/superuser password is required to mount internal partitions. To change this behavior, consult the MX/antiX Wiki.
    - New external partitions. Formatting a new partition with ext4 requires root permissions, which can lead to unexpected or undesired result of the regular user not being able to write any files to the partition. To change this behavior, consult the MX/antiX Wiki.
    - Manual operations. Although MX User Manager covers most daily situations, sometimes it can be preferable to deal with the command line. Basic permissions are represented by r (read), w (write) and x (execute); a dash indicates no permissions. To view permissions of a file on the command line, type this: ls -l NameofFile. The -l switch will cause the file to be listed in long format, displaying its permissions among other information.



```
jb@UTRA:~
$ -rw-r--r--   newbie users 277 Jan 7 14:42 .asoundrc
```

Figure 7-2: Viewing file permissions

The characters right after the opening dash (indicating it is a regular file) contains the three permissions (read/write/execute) for owner, group and others: 9 characters total. Here it shows that the owner has read and write but not execute (rw-), but the group and others can only read. The owner in this case is specified to be “newbie” who belongs to the group “users”.
If for some reason it was necessary to change the ownership of this file to root using the command line, you would use the chown command like in this example: chown root /home/newbie/.asoundrc. For details on using chown, as well as the more detailed chmod, see Links section.

Links

- MX/antiX Wiki: Permissions
- File Permissions

7.4 Configuration files

With only rare exceptions, program and system settings on MX Linux are stored in discrete plain text configuration files; there is no “Registry” which requires special tools to edit. Most configuration files are just simple lists of parameters and values which are read by programs when they launch to determine their behavior.

7.4.1 User config files

Files that hold individual user settings (such as high scores for your games, or the layout of your desktop) are stored within a Users home directory, typically as a hidden file or directory, and can only be edited by that user or by root. These personal configuration files are actually less often edited directly than system files because most of the user configuration is done graphically through the applications themselves. When you open an application and click Edit > Preferences, for example, your selections are written to a (usually hidden) configuration file in your user directory. Likewise in Firefox, when you type about:config in the address bar, you are editing the hidden configuration files.

7.4.2 System config files

Files that hold system-wide configurations or defaults (such as the file that determines which services automatically launch during boot up) are largely stored in the /etc/ directory and are only editable by root. Most of these files are never touched directly by regular users, such as these for instance:

- /etc/rc.d/rc5.d — Contains files to control runlevel 5 into which MX Linux boots after login.
- /etc/sysconfig/keyboard — Used to configure the keyboard.
- /etc/network/interfaces — Defines internet interfaces on the system.

Some configuration files can contain just a few lines, or even be empty, while others may be quite long. The important point is that if you are looking for a configuration file for an application or process, head for the /etc directory and look around. Caution: because these files affect the whole system, 1) back up any file you intend to edit (easiest in Thunar: copy and paste back in, adding BAK at the end of the file name), and 2) be very careful!

7.4.3 Example

Sound problems can be solved with a number of graphical and command-line tools, but once in a while a user needs to edit directly the system-wide configuration file. For many systems, this will be /etc/modprobe.d/snd-hda-intel.conf. It is a simple file whose top paragraph looks like this:

```
# some chips require that the model be set manually
# for example asus g71 series may need model=g71v
options snd-hda-intel model=auto
```

To try to get sound, you might decide to substitute the exact information about the sound model in place of the word “auto.”. To find out your sound model, you could open a terminal and type:

```
lspci | grep Audio
```

The output will depend on the system, but it will take the following form:

```
00:05.0 Audio device: nVidia Corporation MCP61 High Definition Audio (rev a2)
```

Now you can plug that information back into the configuration file:

```
# some chips require that the model be set manually
# for example asus g71 series may need model=g71v
options snd-hda-intel model=nvidia
```

You would save the file, reboot the machine, and hopefully your sound should be working. You could also try more precision by using model=nvidia mcp61 instead, if the first did not work.
Links

- Understanding Linux Configuration Files
- File Permissions

7.5 Runlevels

MX Linux boots up by executing the program init. After completing the boot process, init executes all startup scripts in a directory specified by the default runlevel (this runlevel is given by the entry for id in /etc/inittab). Like most other Linux versions, MX Linux has 7 runlevels:

Table 10: Runlevels in MX Linux

Runlevel	Comment
0	Halt the system
1	Single-user mode: provides a root console without logon. Useful if you lose your root password
2	Multiuser with no network
3	Console logon, no X (i.e. no GUI)
4	Not used/custom
5	Default GUI logon
6	Reboot the system

MX Linux defaults to runlevel 5, therefore any init scripts set up in the level 5 config file will run at boot.

Use

Understanding runlevels can be handy. When users have a problem with X Window Manager, for instance, they can not correct it on the default runlevel 5, because X is running on that level. But they can get to runlevel 3 to work on the problem in one of two ways.

- From the Desktop: press Ctrl-Alt-F1 to get out of X. To actually drop to runlevel 3, become root and type telinit 3; this will stop all the other services still operating on runlevel 5.
- From the GRUB menu: press e (for edit) when you see the GRUB screen. On the subsequent screen, add a space and the number 3 at the end of the line that starts with “linux” located one above the lowest line (the actual boot command). Press F-10 to boot.
-

Once the cursor is at a prompt, login with your normal username and password. If necessary, you can also login as “root” and provide the administrative password. Useful commands when you are looking at the prompt on runlevel 3 include:

Table 11: Common runlevel 3 commands

Command	Comment
runlevel	Returns the number of the runlevel you are on.
halt	Run as root. Shuts the machine down. If that does not work on your system, try poweroff.
reboot	Run as root. Reboots the machine.
<application>	Runs the application, as long as it is not graphical. For instance, you can use the command nano to edit text files, but not leafpad.
Ctrl-Alt-F7	If you used Ctrl-Alt-F1 to drop out from a running desktop but did not continue down to runlevel 3, this command brings you back to your desktop.
telinit 5	Run as root. If you are on runlevel 3, enter this command to get to the login manager lightdm.

NOTE: these commands may change in the future if MX Linux switches to a new system manager.

Links

- Wikipedia: Runlevel
- The Linux Information Project: Runlevel Definition

7.6 The kernel

7.6.1 Introduction

To provide a background, here is a simplified diagram and description of the kernel’s position in a Linux OS, borrowed from Anatomy of the Linux kernel.

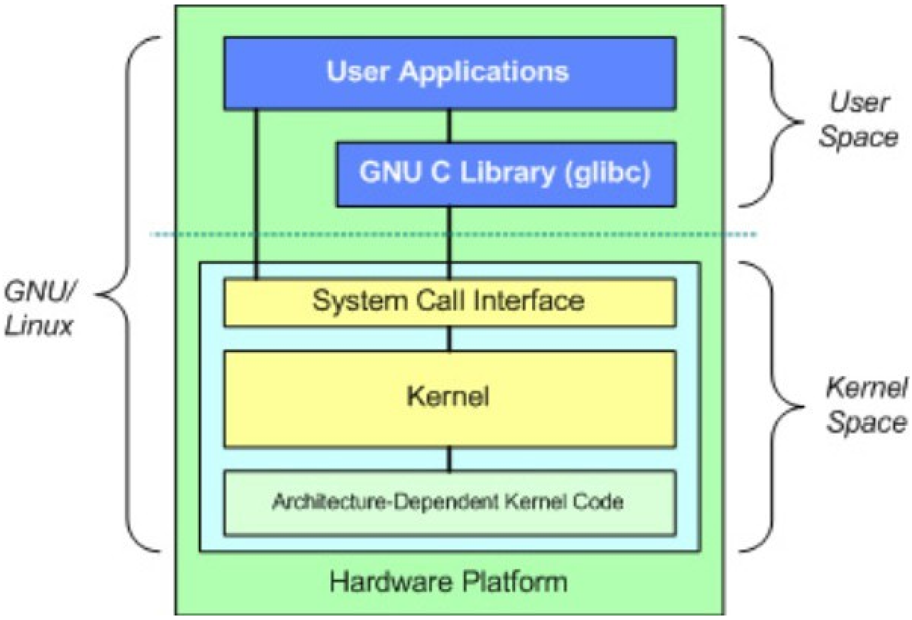


Figure 7-3: Diagram of the Linux kernel

At the top is the user space, or application space. This is where the user applications are executed. Below the user space is the kernel space. Here, the Linux kernel exists. There is also the GNU C Library (glibc). This provides the system call interface that connects to the kernel and provides the mechanism to transition between the user-space application and the kernel. This is important because the kernel and user application occupy different protected address spaces. And while each user-space process occupies its own virtual address space, the kernel occupies a single address space.

7.6.2 Upgrading/Downgrading

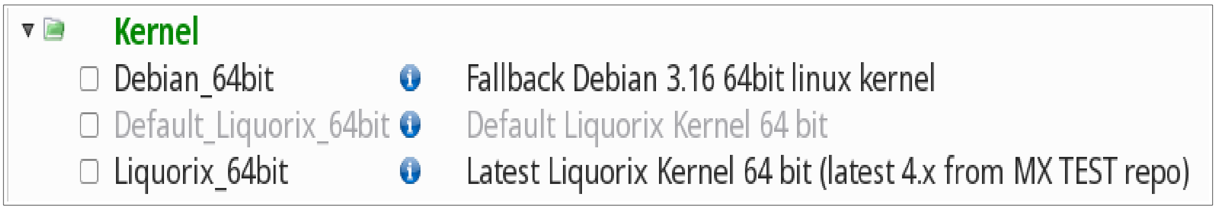
Basic

Unlike other software on your system, the kernel is not upgraded automatically except below the minor revision level (indicated by the third number in the kernel name). Before you change your current kernel, you would do well to ask yourself some questions: Why do I want to upgrade the kernel? Is there a driver I need for new hardware, for instance? Am I aware that I might have problems of one kind or another?

MX Linux provides an easy method of upgrading/downgrading the default kernel: open MX Package Installer and click on the “Kernel” category. There you will see two active entries surrounding the greyed-out default kernel:

- A “fallback” kernel that represents a reasonable choice for dropping down a level
- The latest available kernel in the same series as the default kernel

Once you check and install the new kernel, reboot and make sure the new kernel is highlighted; if not, click on the options line and select what you want.



▾ Kernel

☐ Debian_64bit ⓘ Fallback Debian 3.16 64bit linux kernel

☐ Default_Liquorix_64bit ⓘ Default Liquorix Kernel 64 bit

☐ Liquorix_64bit ⓘ Latest Liquorix Kernel 64 bit (latest 4.x from MX TEST repo)

Figure 7-4: Kernel options in MX Package Installer for 64bit architecture

Advanced

Here is a basic approach for manually upgrading the Linux kernel on your system.

- First, find out what you currently have installed. Open a terminal and enter inxi -S. For instance, a user of MX 15 64 bit version would see something like this:

  ```
  $ inxi -S
  System: Host: UTRA Kernel: 4.2-3.dmz.3-liquorix-amd64 x86_64 (64 bit)
  Desktop: Xfce 4.12.2 Distro: MX-16_x64-mx Metamorphosis
  ```

  Be sure to write down the name of the kernel from the output of that command.
- Second, select and install a new kernel. Open Synaptic, search on linux-image and look for a higher kernel number that matches the architecture (e.g., 686) and processor (e.g., PAE) that you already have, unless you have a good reason to change. Install the one you want or need in the usual manner.

- Third, install the linux-headers package that match the new kernel you selected. There are two methods of doing this.
    - Look carefully at the Synaptic entries beginning linux-headers and match the kernel.
    - Alternatively, you can install the headers more easily after rebooting into the new kernel by typing the following code in a root terminal:

      ```
      apt-get install linux-headers-$(uname -r) 5
      ```

      Headers will also be installed if you use a command such as m-a prepare.
- When you reboot, you will automatically boot into the highest available kernel. If it doesn't work, you have the option to return to what you were using: reboot, and when you see the GRUB screen highlight Advanced Options for whatever partition you want to boot into, then select the kernel and press enter. 10,00

7.6.3 Kernel upgrade and drivers

Dynamic Kernel Module Support(DKMS) automatically recompiles all DKMS driver modules when a new kernel version is installed. This allows drivers and devices outside of the mainline kernel to continue working after a Linux kernel upgrade. The exception concerns proprietary graphics drivers (Section 3.3.2).

- NVidia drivers
    - If installed with sgfxi, they must be rebuilt with sgfxi, see Section 6.5.3
    - If installed with the MX Nvidia driver installer or via synaptic/apt-get, the kernel modules may need to be rebuilt. Re-running MX Nvidia driver installer from the menu should offer to reinstall and rebuild the modules. If your reboot gets stuck at a console prompt, try "sudo ddm-mx -i nvidia" to reinstall and rebuild the driver modules.
- AMD/ATI fglrx drivers
    - If you installed a proprietary ATI driver and upgraded the kernel, re-run MX AMD/ATI graphics installer from the menu.
    - If you boot to a command line, run sudo ddm-mx -i ati.
- Intel drivers
    - You may need to upgrade the driver, depending on the kernel you select for upgrade target.

7.6.4 More options

Other considerations and choices exist with respect to kernels:

- Other pre-rolled kernels exist such as the Liquorix kernel, which is a version of the Zen kernel and is intended to provide a better desktop use experience in terms of responsiveness, even under heavy loads such as during gaming, plus low latency (important for audio work). MX Linux updates the Liquorix kernels frequently, so it is most easily installed through the MX Package Installer, in the Kernel section.
- Distros (e.g., MX's code-parent antiX) often roll their own.
- Knowledgeable individuals may compile a specific kernel for particular hardware.

7.6.5 Links

- Wikipedia: Linux kernel
- Anatomy of the Linux kernel
- Linux kernel archives
- Interactive map of Linux kernel

7.6.6 Kernel panic and recovery

A kernel panic is a relatively rare action taken by the MX Linux system when it detects an internal fatal error from which it can not safely recover. It can be caused by a number of different factors that range from hardware problems to a bug in the system itself. When you get a kernel panic, try rebooting with the MX Linux LiveMedium, which will overcome temporarily any software problems and hopefully allow you to see and offload your data. If that doesn't work, then unplug all unnecessary hardware and try again.

Your first concern is to access and secure your data. Hopefully, you have it backed up somewhere. If not, you can use one of the data recovery programs such ddrescue that is supplied with MX Linux. Your last resort is to take your hard drive to a professional recovery business.

There are a number of steps you might have to take to recover a functional MX Linux system once you have your data safe, although ultimately you may have to reinstall using the LiveMedium. Depending on the type of failure, the following steps may be undertaken:

1. Remove packages that broke the system.
2. Reinstall the graphic driver.
3. Reinstall GRUB using MX Boot Repair.
4. Reset the root password.
5. Reinstall MX Linux, selecting the check box on Screen 2 to keep /home so that your personal configurations will not be lost.

Be sure to ask on the Forum if you have any questions about these procedures.

Links

- GNU C Library Home Page
- Ddrescue