

Linux

Faculty Orientation Program,
Savitribai Phule Pune University, Pune

Tushar B. Kute,
Data Scientist, MITU Research, Pune.

Contents

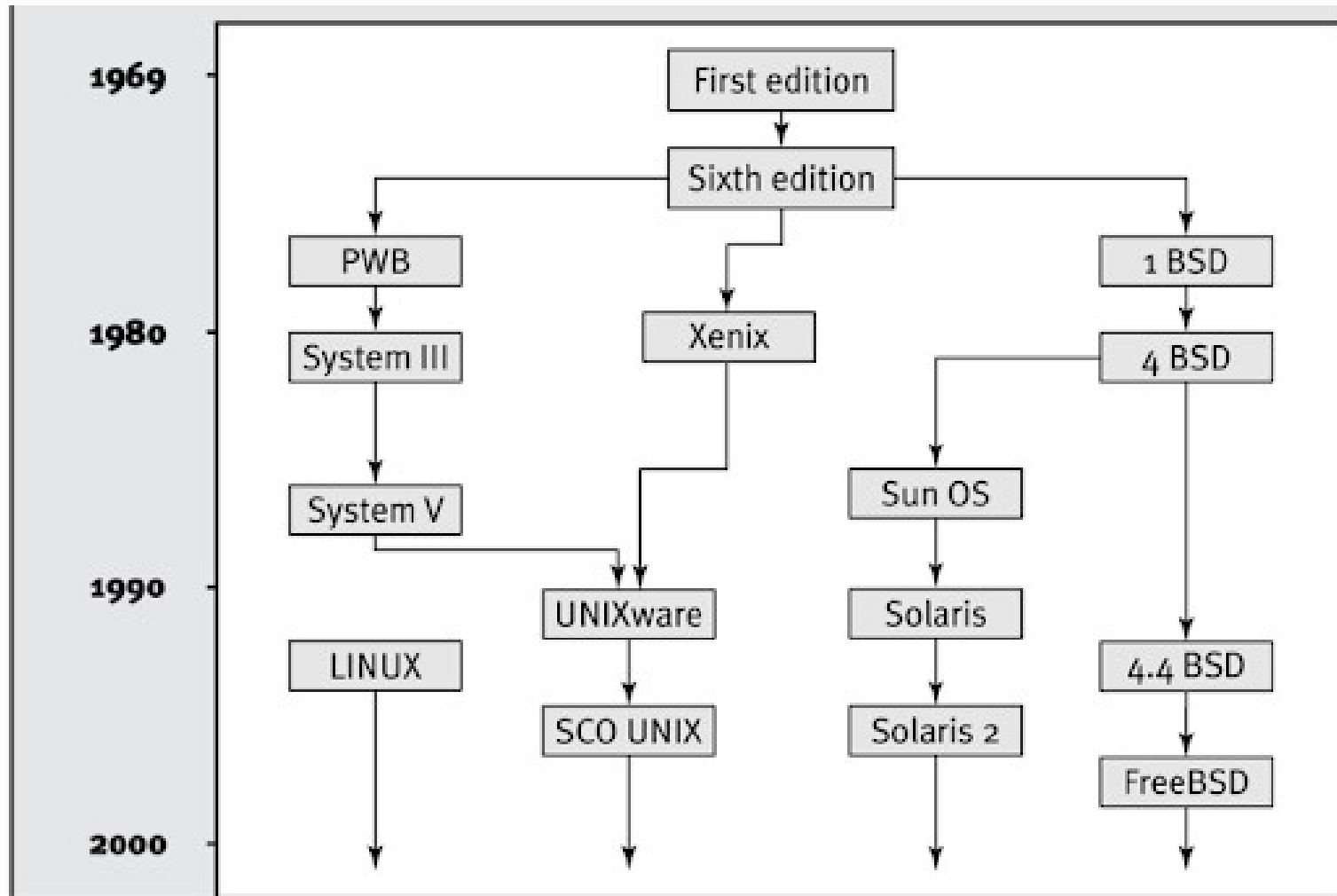
- History Of Unix and Linux , Overview Of Linux - Linux Goals, Interfaces to Linux, The Shell, Linux Utility Programs,
- Kernel structure, Processes in Linux – Process management system calls in Linux,
- Implementation of process and threads in Linux, Process scheduling Linux, Booting

Unix

- The Unix operating system found its beginnings in **MULTICS**, which stands for Multiplexed Operating and Computing System.
- The MULTICS project began in the mid 1960s as a joint effort by General Electric, Massachusetts Institute of Technology and Bell Laboratories. In 1969 Bell Laboratories pulled out of the project.
- One of Bell Laboratories people involved in the project was Ken Thompson. He liked the potential MULTICS had, but felt it was too complex and that the same thing could be done in simpler way.
- In 1969 he wrote the first version of Unix, called UNICS. UNICS stood for **Uniplexed Operating and Computing System**. Although the operating system has changed, the name stuck and was eventually shortened to Unix.

- Ken Thompson teamed up with Dennis Ritchie, who wrote the first **C compiler**.
- In 1973 they rewrote the **Unix kernel** in C. The following year a version of Unix known as the Fifth Edition was first licensed to universities.
- The Seventh Edition, released in 1978, served as a dividing point for two divergent lines of Unix development.
- These two branches are known as SVR4 (System V) and BSD.

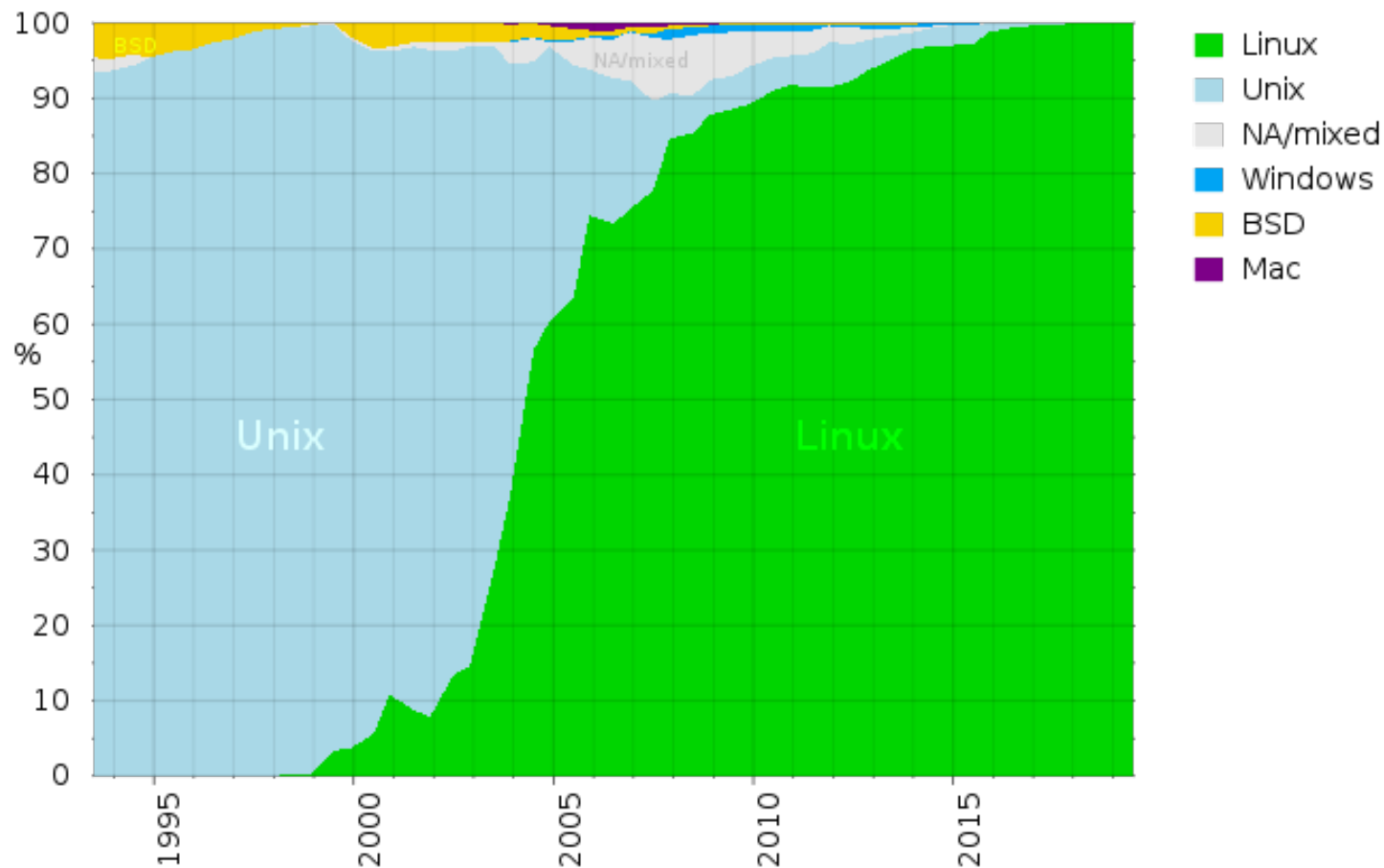
Unix



Linux Operating System

- Linux, computer operating system created in the early 1990s by Finnish software engineer **Linus Torvalds**.
- While still a student at the University of Helsinki, Torvalds started developing Linux to create a system similar to **MINIX**, a UNIX operating system.
- In 1991 he released version 0.02; Version 1.0 of the Linux kernel, the core of the operating system, was released in 1994.

Unix vs. Linux



Linux Design Goals

- Speed,
- Efficiency, and
- Standardization

Linux Operating System

- From smartphones to cars, supercomputers and home appliances, home desktops to enterprise servers, the Linux operating system is everywhere.
- Linux has been around since the mid-1990s and has since reached a user-base that spans the globe. Linux is actually everywhere: It's in your phones, your thermostats, in your cars, refrigerators, Roku devices, and televisions.
- It also runs most of the Internet, all of the world's top 500 supercomputers, and the world's stock exchanges.
- But besides being the platform of choice to run desktops, servers, and embedded systems across the globe, Linux is one of the most reliable, secure and worry-free operating systems available.

Linux Operating System

- Just like Windows, iOS, and Mac OS, Linux is an operating system.
- In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system.
- An operating system is software that manages all of the hardware resources associated with your desktop or laptop.
- To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn't function.

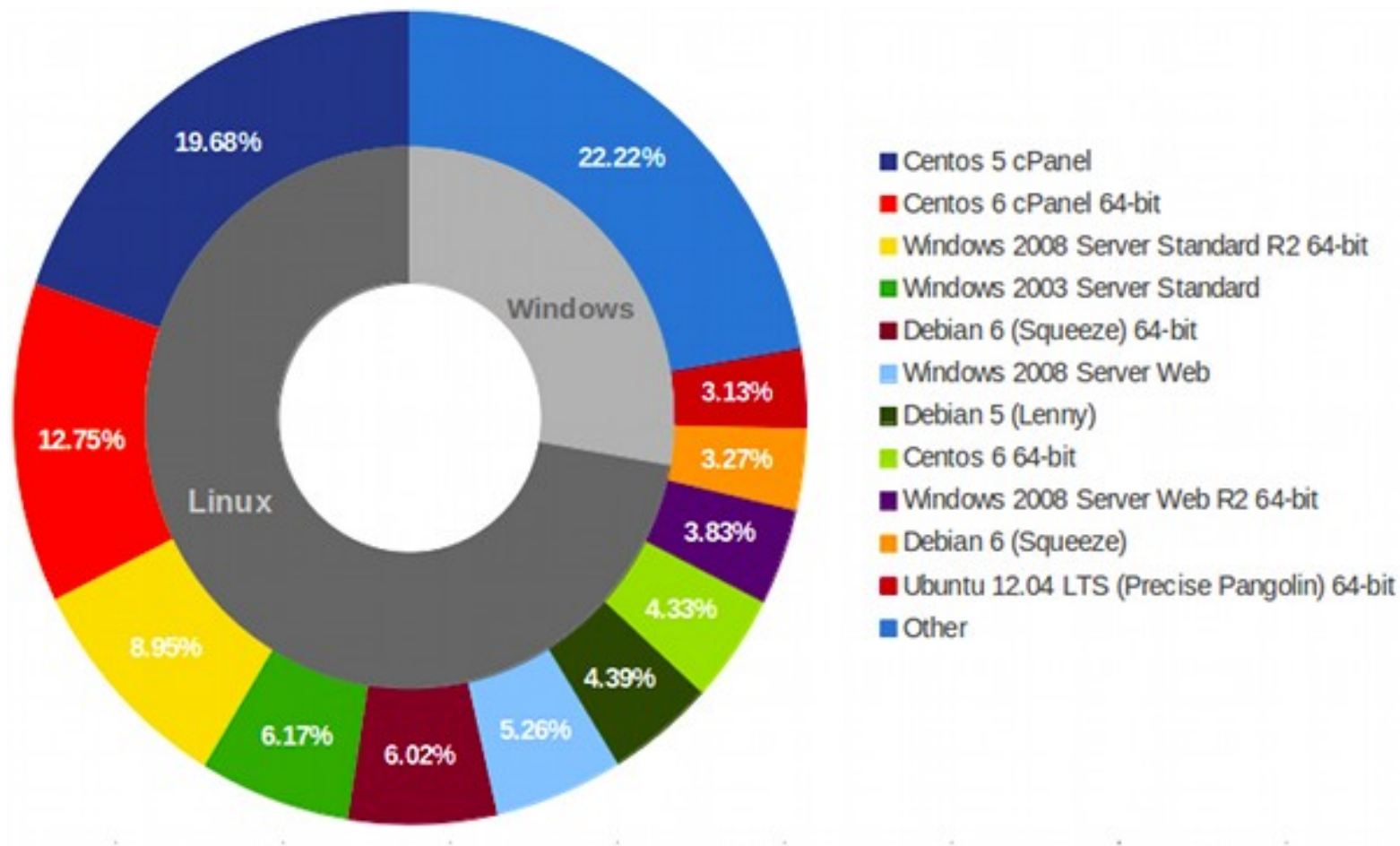
Top Supercomputers

Top 10 positions of the 57th TOP500 in June 2021^[32]^[needs update]

Rank (previous)	Rmax Rpeak (PFLOPS)	Name	Model	CPU cores	Accelerator (e.g. GPU) cores	Interconnect	Manufacturer	Site country	Year	Operating system
1	442.010 537.212	Fugaku	Supercomputer Fugaku	158,976 × 48 A64FX @2.2 GHz	0	Tofu interconnect D	Fujitsu	RIKEN Center for Computational Science 🇯🇵 Japan	2020	Linux (RHEL)
2▼ (1)	148.600 200.795	Summit	IBM Power System AC922	9,216 × 22 POWER9 @3.07 GHz	27,648 × 80 Tesla V100	InfiniBand EDR	IBM	Oak Ridge National Laboratory 🇺🇸 United States	2018	Linux (RHEL)
3▼ (2)	94.640 125.712	Sierra	IBM Power System S922LC	8,640 × 22 POWER9 @3.1 GHz	17,280 × 80 Tesla V100	InfiniBand EDR	IBM	Lawrence Livermore National Laboratory 🇺🇸 United States	2018	Linux (RHEL)
4▼ (3)	93.015 125.436	Sunway TaihuLight	Sunway MPP	40,960 × 260 SW26010 @1.45 GHz	0	Sunway ^[33]	NRPC	National Supercomputing Center in Wuxi 🇨🇳 China ^[33]	2016	Linux (Raise)
5▲ (new)	64.590 89.795	Perlmutter	HP	7 × 64 Epyc 7763 @2.45 GHz	? × 108 Ampere A100	Slingshot-10	Nvidia	NERSC 🇺🇸 United States	2021	Linux (Cray Linux Environment)
6▼ (5)	63.460 79.215	Selene	Nvidia	1,120 × 64 Epyc 7742 @2.25 GHz	4,480 × 108 Ampere A100	Mellanox HDR Infiniband	Nvidia	Nvidia 🇺🇸 United States	2020	Linux (Ubuntu)
7▼ (6)	61.445 100.679	Tianhe-2A	TH-IVB-FEP	35,584 × 12 Xeon E5-2692 v2 @2.2 GHz	35,584 × 128 Matrix-2000 ^[34]	TH Express-2	NUDT	National Supercomputer Center in Guangzhou 🇨🇳 China	2013	Linux (Kylin)
8▼ (7)	44.120 70.980	JUWELS (booster module) ^{[35][36]}	BullSequana XH2000	1,872 × 24 AMD Epyc 7402 @2.8 GHz	3,744 × 108 Ampere A100	Mellanox HDR Infiniband	Atos	Forschungszentrum Jülich 🇩🇪 Germany	2020	Linux (CentOS)
9▼ (8)	35.450 51.721	HPC5	Dell	3,640 × 24 Xeon Gold 6252 @2.1 GHz	7,280 × 80 Tesla V100	Mellanox HDR Infiniband	Dell EMC	Eni 🇮🇹 Italy	2020	Linux (CentOS)
10▼ (9)	23.516 38.746	Frontera	Dell C6420	16,016 × 28 Xeon Platinum 8280 @2.7 GHz (subsystems with e.g. POWER9 CPUs and Nvidia GPUs were added after official benchmarking ^[10])	0	InfiniBand HDR	Dell EMC	Texas Advanced Computing Center 🇺🇸 United States	2019	Linux (CentOS)

<https://en.wikipedia.org/wiki/TOP500>

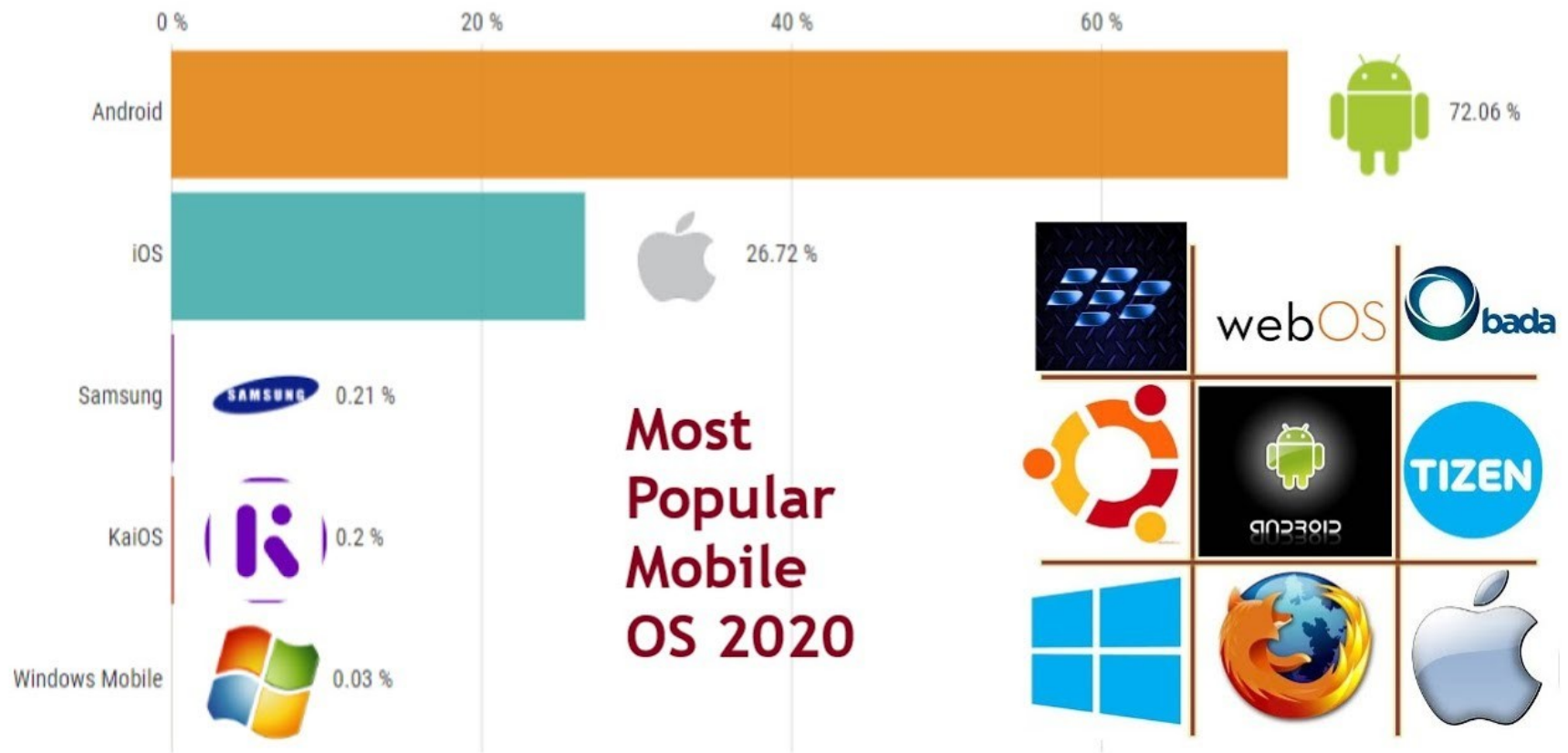
Top Server Operating Systems



Top Mobile Operating Systems

Most Popular Mobile Operating System 2000-2020

YouTube/WorldTopData | www.worldtopdata.in



Most Popular Mobile OS 2020



Linux Components

- **Bootloader** – The software that manages the boot process of your computer. For most users, this will simply be a splash screen that pops up and eventually goes away to boot into the operating system.
- **Kernel** – This is the one piece of the whole that is actually called Linux. The kernel is the core of the system and manages the CPU, memory, and peripheral devices. The kernel is the lowest level of the OS.

Linux Components

- Init system – This is a sub-system that bootstraps the user space and is charged with controlling daemons. One of the most widely used init systems is systemd, which also happens to be one of the most controversial. It is the init system that manages the boot process, once the initial booting is handed over from the bootloader (i.e., GRUB or GRand Unified Bootloader).

Linux Components

- Daemons – These are background services (printing, sound, scheduling, etc.) that either start up during boot or after you log into the desktop.
- Graphical server – This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just X.

Linux Components

- Desktop environment – This is the piece that the users actually interact with. There are many desktop environments to choose from (GNOME, Cinnamon, Mate, Pantheon, Enlightenment, KDE, Xfce, etc.). Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, and games).

Linux Components

- Applications – Desktop environments do not offer the full array of apps. Just like Windows and macOS, Linux offers thousands upon thousands of high-quality software titles that can be easily found and installed. Most modern Linux distributions (more on this below) include App Store-like tools that centralize and simplify application installation.

FOSS Linux Features

- Linux is also distributed under an open source license. Open source follows these key tenants:
 - The freedom to **run** the program, for any purpose.
 - The freedom to **study** how the program works, and change it to make it do what you wish.
 - The freedom to **redistribute** copies so you can help your neighbor.
 - The freedom to **distribute** copies of your modified versions to others.

Linux Features

- Portable
- Open Source
- Multi-User
- Multiprogramming
- Hierarchical File System
- Shell
- Security

Distributions

- Linux has a number of different versions to suit any type of user. From new users to hard-core users, you'll find a "flavor" of Linux to match your needs.
- These versions are called distributions (or, in the short form, "distros").
- Nearly every distribution of Linux can be downloaded for free, burned onto disk (or USB thumb drive), and installed (on as many machines as you like).

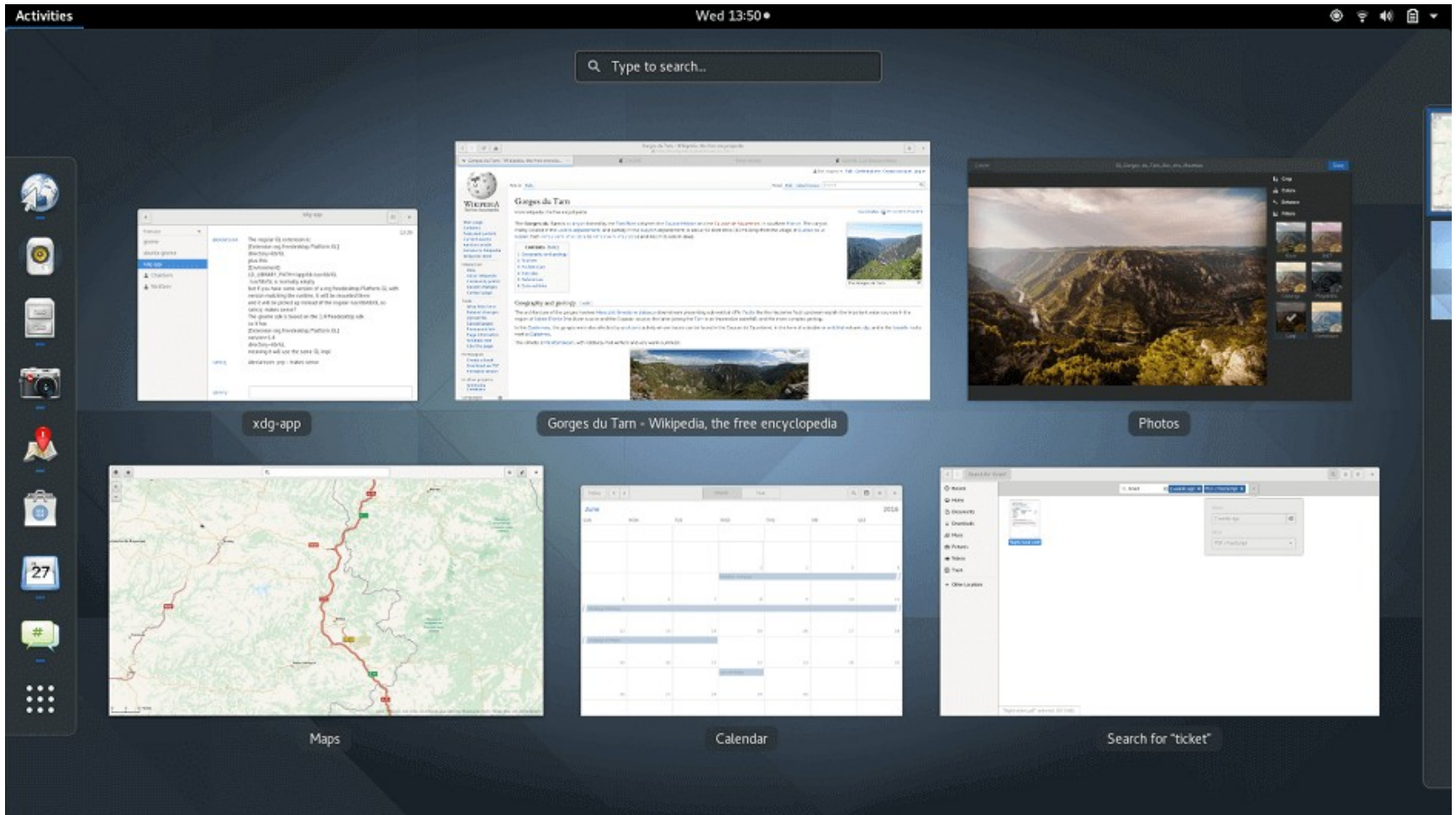
Distributions

- Popular Linux distributions include:
 - LINUX MINT
 - MANJARO
 - DEBIAN
 - UBUNTU
 - ANTERGOS
 - SOLUS
 - FEDORA
 - ELEMENTARY OS
 - OPENSUSE

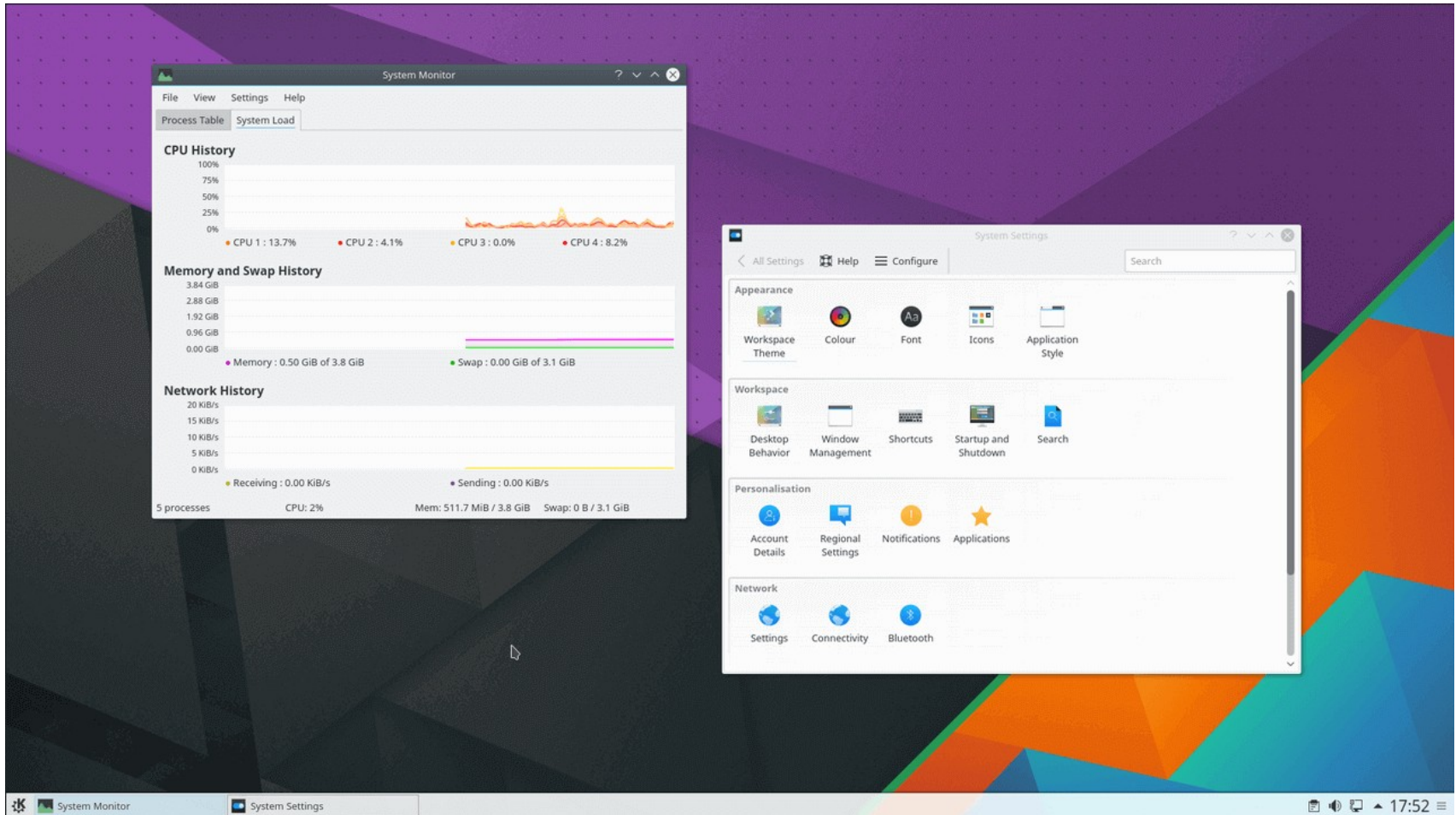
Distributions

- And don't think the server has been left behind. For this arena, you can turn to:
 - Red Hat Enterprise Linux
 - Ubuntu Server
 - Centos
 - SUSE Enterprise Linux

Linux Interface – GNOME



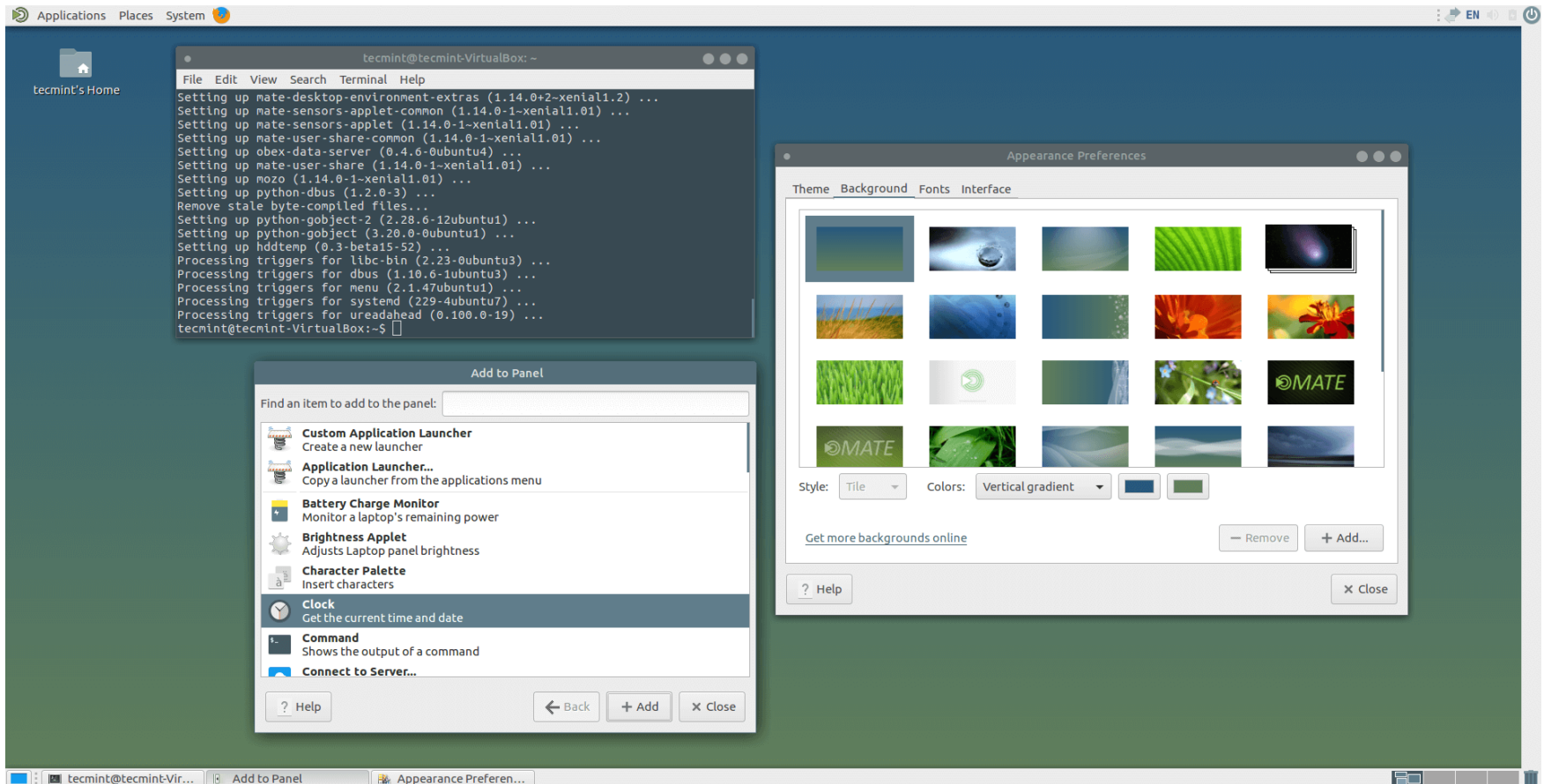
Linux Interface – KDE



Linux Interface – Cinnamon



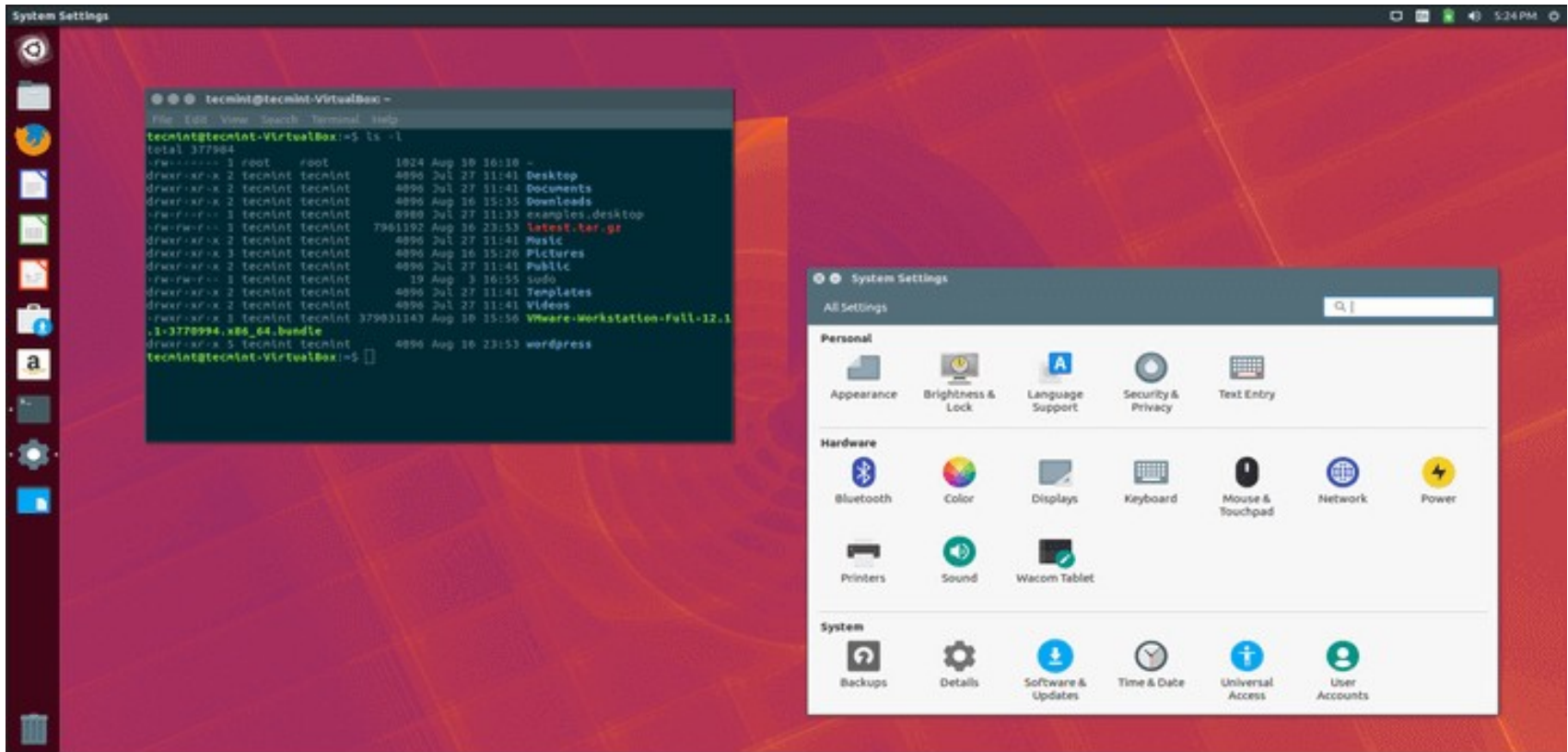
Linux Interface – MATE



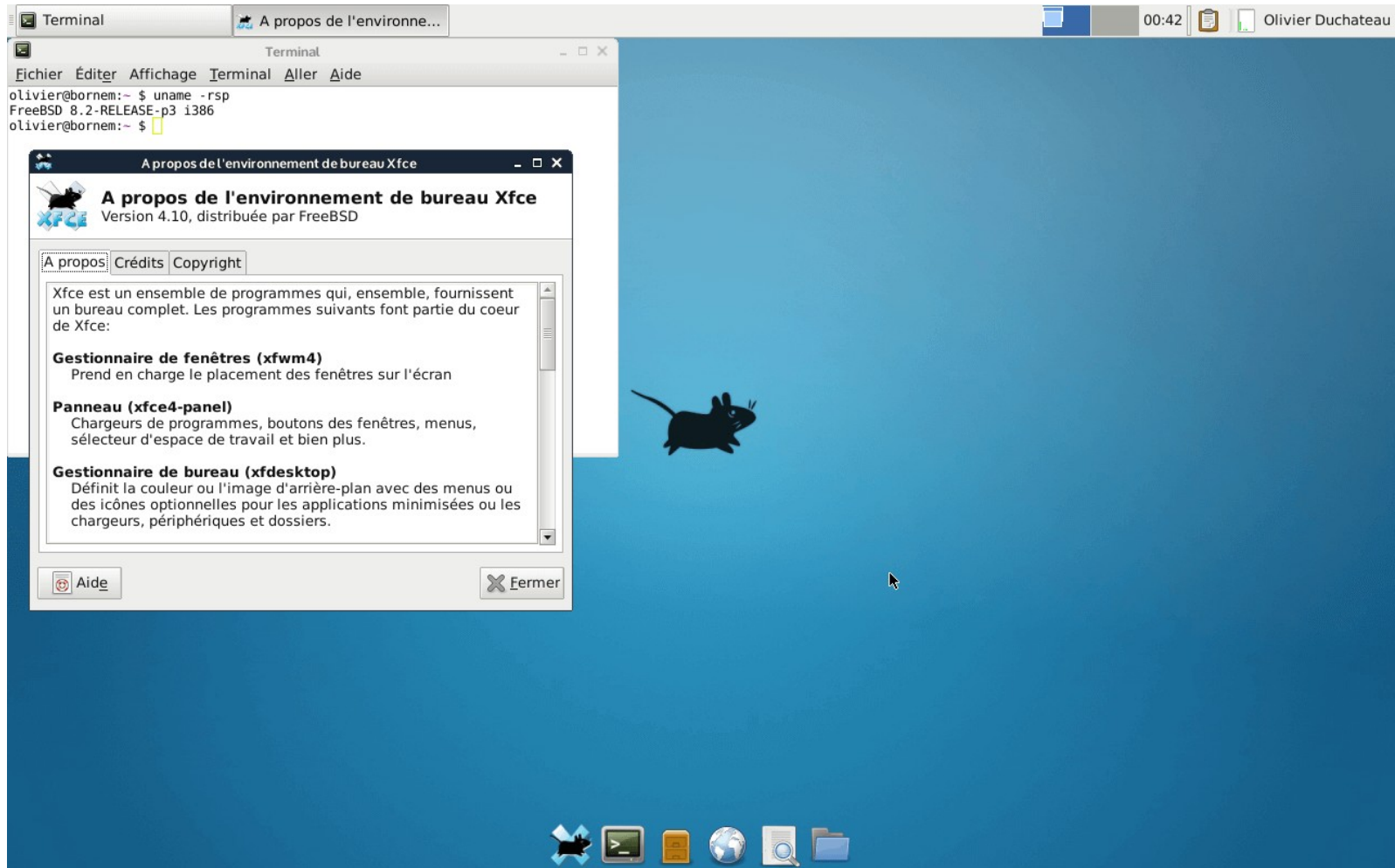
The screenshot displays the MATE Linux desktop environment. In the top-left corner, the desktop menu shows 'Applications', 'Places', and 'System'. The desktop background is a green landscape. A terminal window titled 'tecmint@tecmint-VirtualBox: ~' is open, showing the output of a series of system configuration commands. A 'Add to Panel' dialog is open in the center, listing various desktop applets such as 'Custom Application Launcher', 'Application Launcher...', 'Battery Charge Monitor', 'Brightness Applet', 'Character Palette', 'Clock', 'Command', and 'Connect to Server...'. The 'Clock' applet is currently selected. To the right, the 'Appearance Preferences' window is open, showing a grid of desktop themes and backgrounds. The 'MATE' theme is selected, and the 'Interface' tab is active. The window also shows options for 'Style' (set to 'Title') and 'Colors' (set to 'Vertical gradient').

```
tecmint@tecmint-VirtualBox: ~  
File Edit View Search Terminal Help  
Setting up mate-desktop-environment-extras (1.14.0+2-xenial1.2) ...  
Setting up mate-sensors-applet-common (1.14.0-1-xenial1.01) ...  
Setting up mate-sensors-applet (1.14.0-1-xenial1.01) ...  
Setting up mate-user-share-common (1.14.0-1-xenial1.01) ...  
Setting up obex-data-server (0.4.6-0ubuntu4) ...  
Setting up mate-user-share (1.14.0-1-xenial1.01) ...  
Setting up mozo (1.14.0-1-xenial1.01) ...  
Setting up python-dbus (1.2.0-3) ...  
Remove stale byte-compiled files...  
Setting up python-gobject-2 (2.28.6-12ubuntu1) ...  
Setting up python-gobject (3.20.0-0ubuntu1) ...  
Setting up hddtemp (0.3-beta15-52) ...  
Processing triggers for libc-bin (2.23-0ubuntu3) ...  
Processing triggers for dbus (1.10.6-1ubuntu3) ...  
Processing triggers for menu (2.1.47ubuntu1) ...  
Processing triggers for systemd (229-4ubuntu7) ...  
Processing triggers for ureadahead (0.100.0-19) ...  
tecmint@tecmint-VirtualBox:~$
```

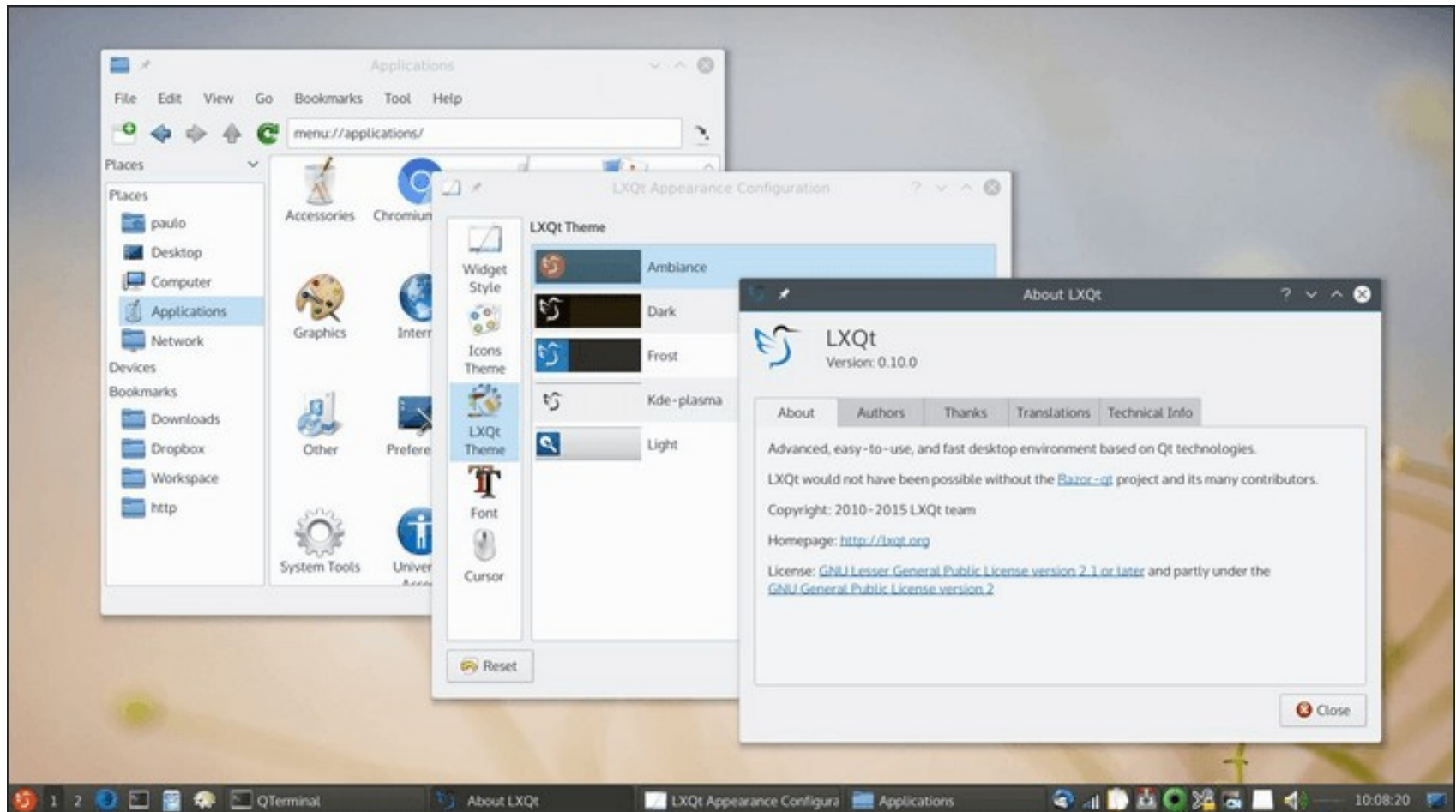
Linux Interface – Unity



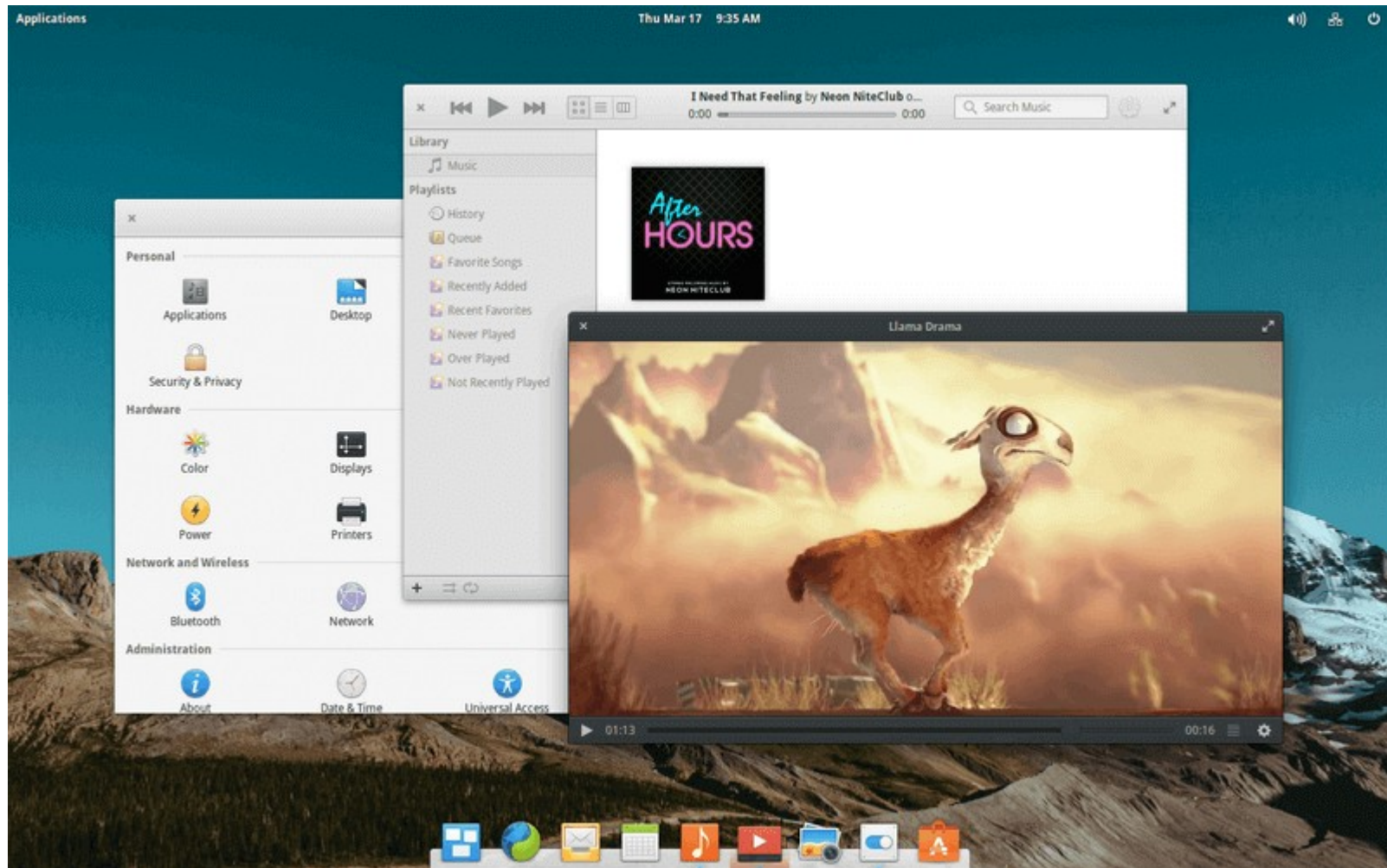
Linux Interface – XFCE



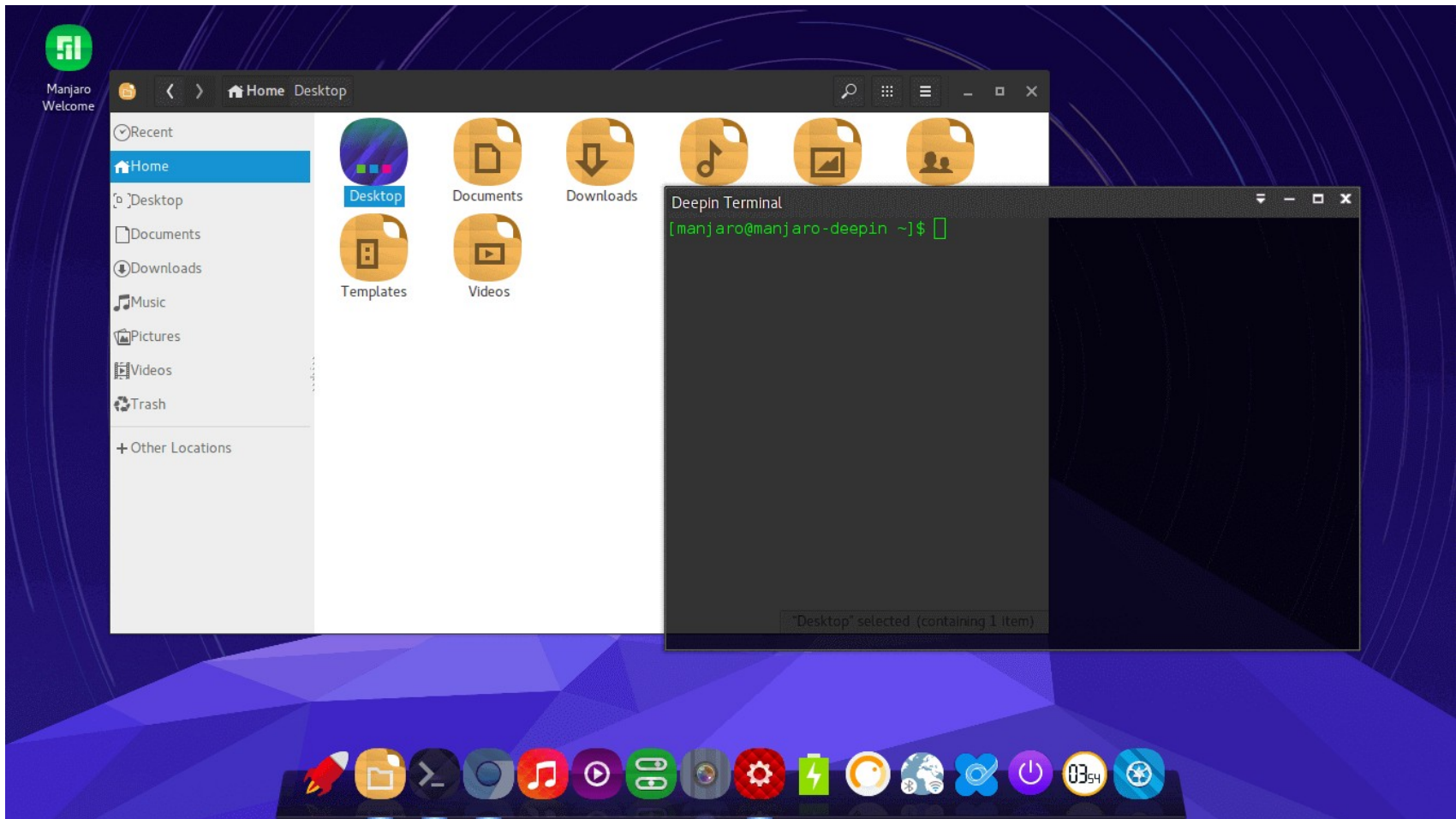
Linux Interface – LXQT



Linux Interface – Pantheon



Linux Interface – Deepin



Shell

- The shell is an interactive interface that allows users to execute other commands and utilities in Linux and other UNIX-based operating systems.
- When you login to the operating system, the standard shell is displayed and allows you to perform common operations such as copy files or restart the system.

Shell

- Linux shells are a lot more powerful than the Windows command line, because they function as a scripting language as well, with a complete set of tools.
- Multiple shells can be installed on a system and it is possible to quickly switch between them.
- Every shell comes with its own syntax and scripting features

Shell

01

Bash
Shell

02

Csh/Tcsh
Shell

03

Ksh Shell

04

Zsh Shell

05

Fish

Bash

- Bash, or the Bourne-Again Shell, is by far the most widely used choice and it comes installed as the default shell in the most popular Linux distributions.
- It was developed from the original UNIX Bourne shell (also known as sh) and was designed to be fully compatible with the old scripts, while adding multiple improved features.

Zsh

- Zsh or the Z-shell was designed from the onset to be interactive and incorporate some of the best features of older shells.
- It provides unique scripting features, it is highly customizable and is easy to use, with spelling correction, command completion or filename globbing.

Korn Shell

- KornShell (also known as ksh) is a very old bash alternative that has been developed in the 1980s.
- It is very similar to bash but doubles as a complete and powerful programming language, so it has a number of passionate fans among sysadmins.
- It is not widely used, so it's a bit more difficult to find online documentation or help.

TC Shell

- Tcsh is a better version of the C shell (csh), which was developed in the UNIX era.
- It is favored by programmers because its syntax is very similar to the C programming language, so they can use its scripting features without having to learn bash.
- It is also the default shell in operating systems from the BSD family.

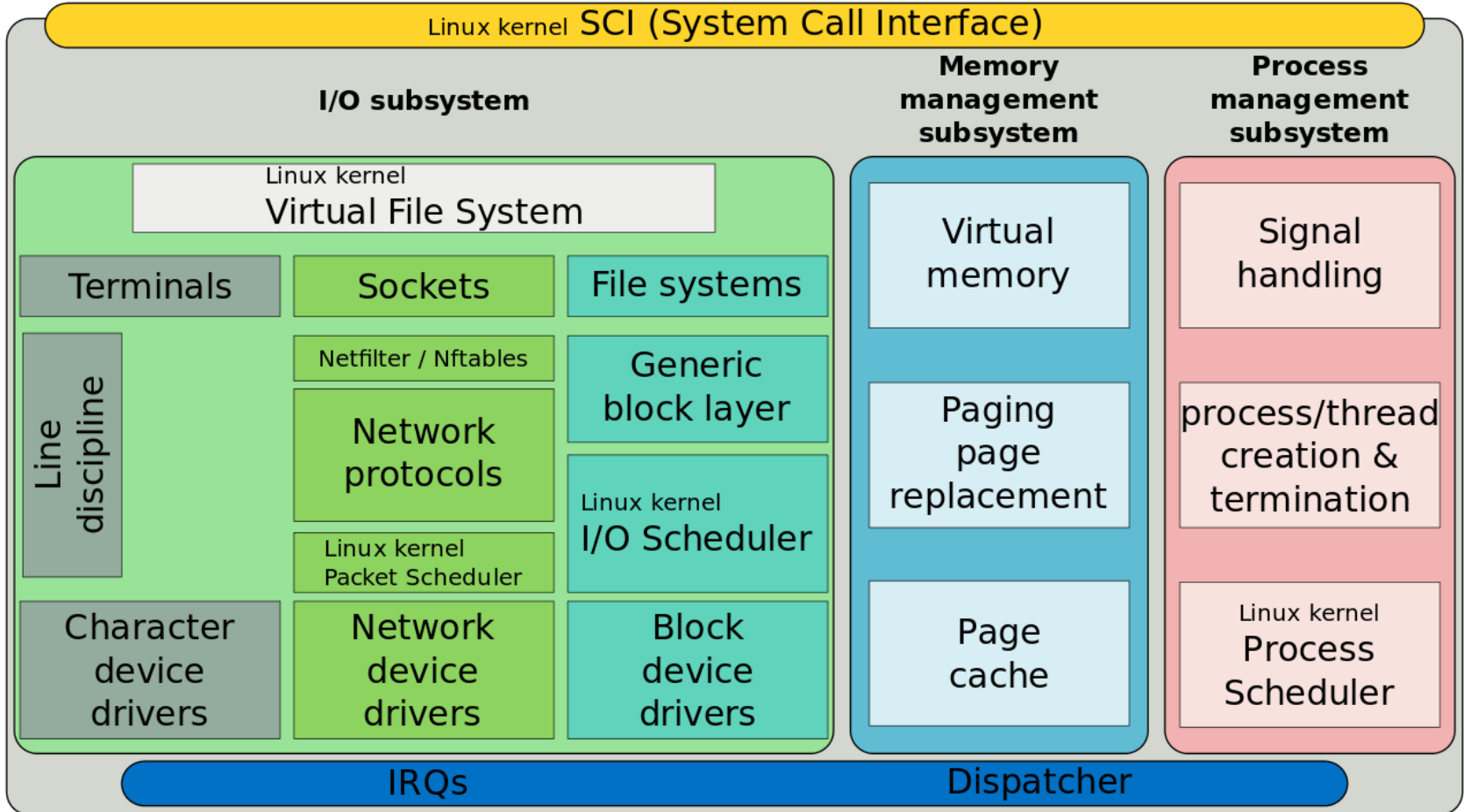
Fish

- Fish, or the friendly interactive shell, aims to be simpler to use and more user friendly than its competitors.
- It is a great choice for Linux beginners, because it uses colors to help the user.
- For example, commands with incorrect syntax are displayed in red, while correct ones are blue.

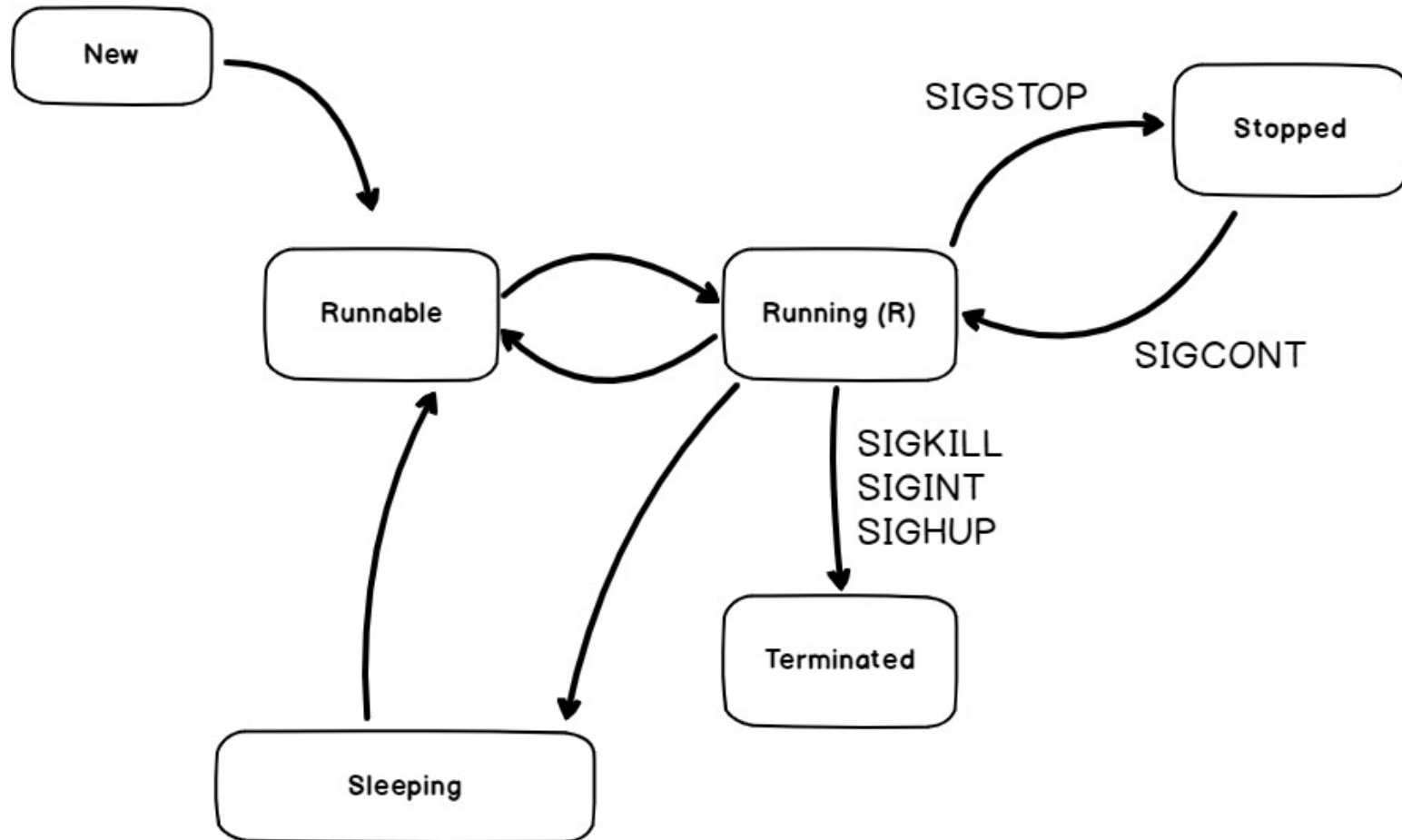
Linux Utility Programs

Program	Typical use
cat	Concatenate multiple files to standard output
chmod	Change file protection mode
cp	Copy one or more files
cut	Cut columns of text from a file
grep	Search a file for some pattern
head	Extract the first lines of a file
ls	List directory
make	Compile files to build a binary
mkdir	Make a directory
od	Octal dump a file
paste	Paste columns of text into a file
pr	Format a file for printing
ps	List running processes
rm	Remove one or more files
rmdir	Remove a directory
sort	Sort a file of lines alphabetically
tail	Extract the last lines of a file
tr	Translate between character sets

Linux Kernel Structure



Linux Process Management



Sleep can be interrupted or not

Process Management System Calls

System call	Description
pid = fork()	Create a child process identical to the parent
pid = waitpid(pid, &statloc, opts)	Wait for a child to terminate
s = execve(name, argv, envp)	Replace a process' core image
exit(status)	Terminate process execution and return status
s = sigaction(sig, &act, &oldact)	Define action to take on signals
s = sigreturn(&context)	Return from a signal
s = sigprocmask(how, &set, &old)	Examine or change the signal mask
s = sigpending(set)	Get the set of blocked signals
s = sigsuspend(sigmask)	Replace the signal mask and suspend the process
s = kill(pid, sig)	Send a signal to a process
residual = alarm(seconds)	Set the alarm clock
s = pause()	Suspend the caller until the next signal

Process Management Example

Parent

```
main()  pid = 3456
{
  pid=fork();
  if (pid == 0)
    ChildProcess();
  else
    ParentProcess();
}

void ChildProcess()
{
  .....
}

void ParentProcess()
{
  .....
}
```

Child

```
main()  pid = 0
{
  pid=fork();
  if (pid == 0)
    ChildProcess();
  else
    ParentProcess();
}

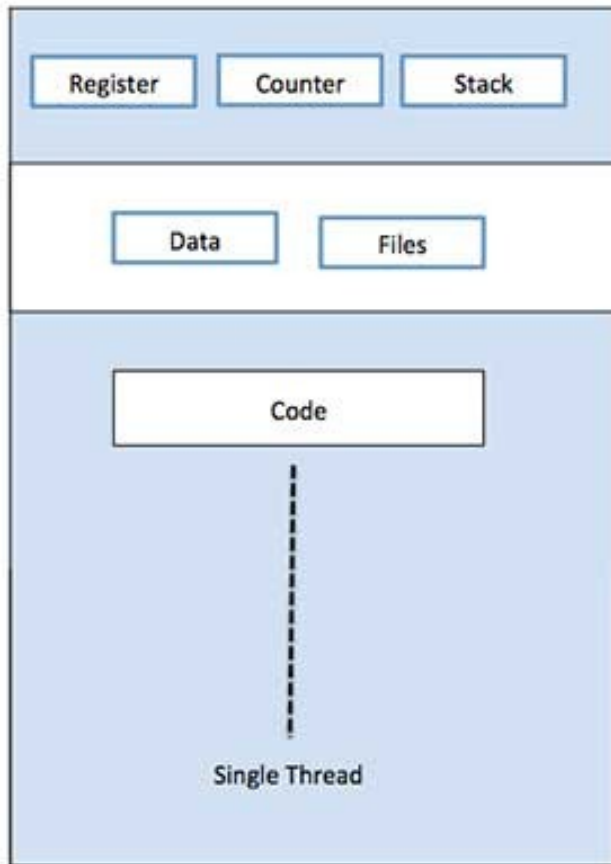
void ChildProcess()
{
  .....
}

void ParentProcess()
{
  .....
}
```

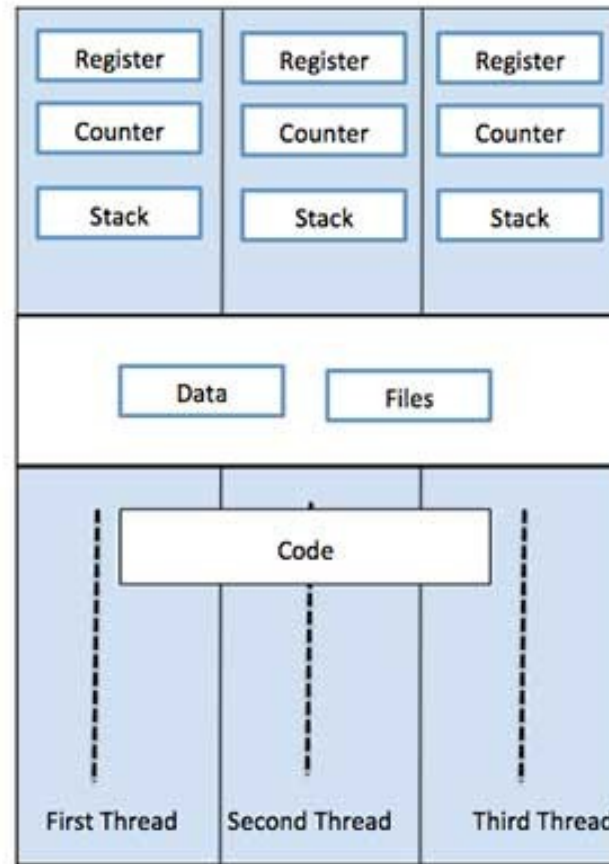
Process Management Commands

- ps
- kill
- top
- htop
- pstree

Process and Threads



Single Process P with single thread



Single Process P with three threads

POSIX Threads

- POSIX – Portable Operating System Interface for Unix.
- For UNIX systems, implementations of threads that adhere to the IEEE POSIX 1003.1c standard are Pthreads.
- Pthreads are C language programming types defined in the pthread.h header/include file.

Thread Library

- Pthread Library (60+ functions)
 - Thread management: create, exit, detach, join, ...
 - Thread cancellation
 - Mutex locks: init, destroy, lock, unlock, ...
 - Condition variables: init, destroy, wait, timed wait,
- Programs must include the file pthread.h
- Programs must be linked with the pthread library (-lpthread)

Multithreaded Program

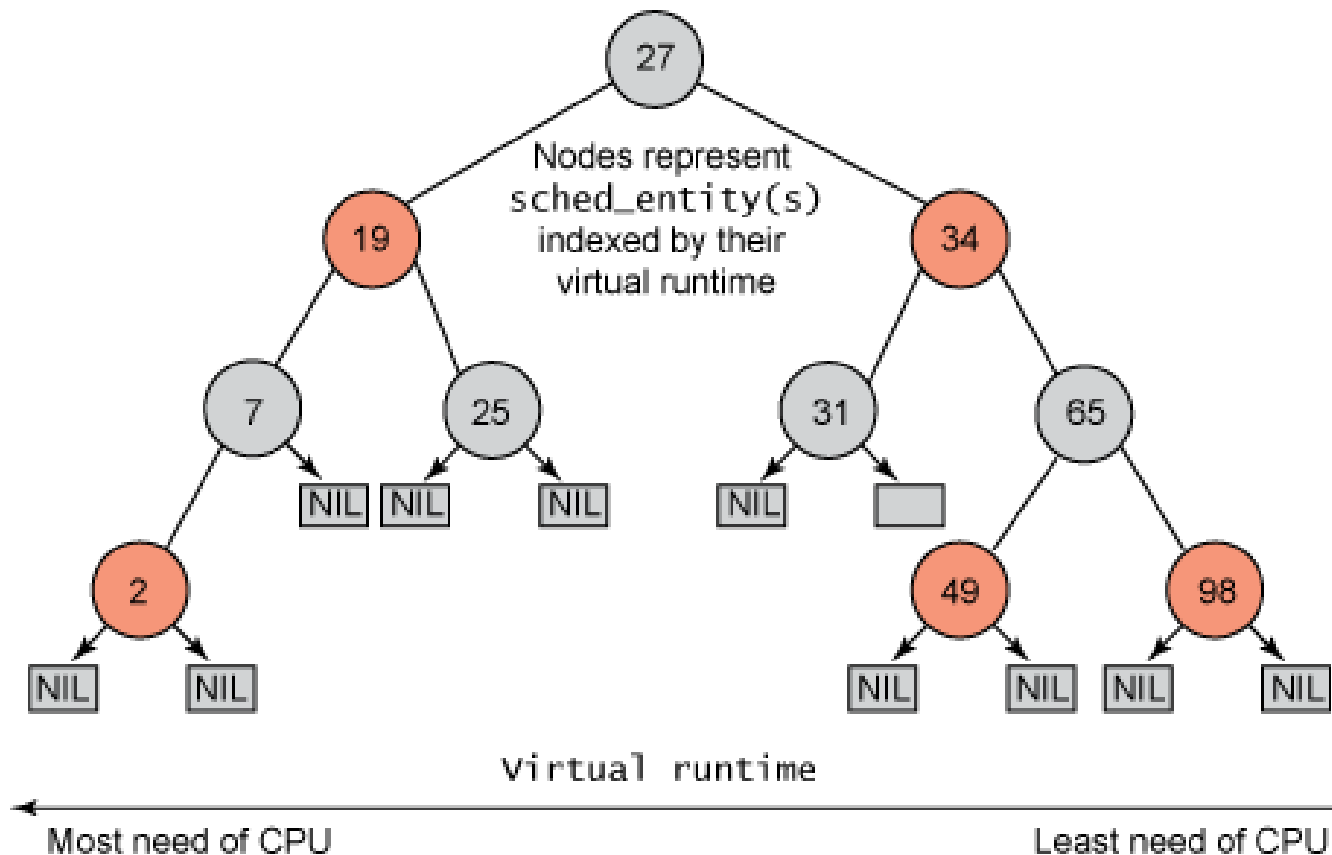
```
#include<unistd.h>
#include<stdio.h>
#include<pthread.h>
int first()
{
    int i;
    for(i=0;;i++)
    {
        printf("\nFirst: %d",i);
        sleep(1);
    }
}
```

```
int main()
{
    pthread_t th;
    int i;
    pthread_create(&th, 0, (void
        *)&first, NULL);
    for(i=0;;i++)
    {
        printf("\nMain: %d",i);
        sleep(1);
    }
    pthread_join(th, NULL);
    return 0;
}
```

Linux Process Scheduling

- The Completely Fair Scheduler (CFS) is a process scheduler that was merged into the 2.6.23 (October 2007) release of the Linux kernel and is the default scheduler of the tasks of the SCHED_NORMAL class (i.e., tasks that have no real-time execution constraints).
- It handles CPU resource allocation for executing processes, and aims to maximize overall CPU utilization while also maximizing interactive performance.

Linux Process Scheduling



Linux Booting Process



Thank you

This presentation is created using LibreOffice Impress 7.0.1.2, can be used freely as per GNU General Public License



@mitu_skillologies



/mITuSkillologies



@mitu_group



/company/mitu-
skillologies



c/MITUSkillologies

Web Resources

<https://mitu.co.in>
<http://tusharkute.com>

contact@mitu.co.in
tushar@tusharkute.com