12.10

Ubuntu
End-to-End
Guide

Table of Contents

Preface....4
Chapter 1 – introduction to Operating Systems....5
Chapter 2 - Introduction to Linux....6
  Linux Kernel....6
  GNU Project 6
  Why choose GNU/Linux Desktop Distribution....7
Chapter 3 – What is Ubuntu?....9
  Distribution....9
  General....9
  What does Ubuntu includes?....9
Chapter 4 – How to install Ubuntu?....11
Chapter 5 - Using Ubuntu GUI....15
  What is Desktop Environment?....15
  Unity Desktop Environment....16
  How To Run Applications?....21
  Built-in Applications....21
  Installing New Applications....22
  Recommended Applications to install at the beginning....22
  Configuring The Network....23
  Configuring Your Printer....24
  Burning Tools....24
  Firewall....25
  Useful Shortcut Keys....25
Chapter 6 - Advanced Ubuntu GUI use....26
  Background....26
  X Window System....26
  Window Manager....26
  Display Manager....27
  Ubuntu's GUI structure....30
  Cool Stuff....30
  Other Desktop Environments....33
  D-Bus....36
  Adding more applications to Aptitude....37
Chapter 7 - Command line theories....41
  What is a Command Line?....41
  CLI Facts....41
  The Virtual File System (VFS)....41
  Shell....43
  Redirection / Descriptors....43
  Pipes....44
  Other....44
Chapter 8 - Using Command line usefully (Bash)....45
  Introduction....45
  GNU Basic Commands....45
  Program Return Codes....48
  Bash Commands....48
  Text viewers and editors....48
  Ubuntu Specific Tools....48
  Deep Diving....49
  CLI Shortcut Keys....50
Chapter 8 - CLI Power User....51
  Introduction....51
  Wildcards....51
  Regular Expressions....51
  Job Control....52

Basic Permissions...53
Scripting Bash shell...55
Chapter 9 - Basic Administering Linux...56
Processes...56
Deamons...57
Chapter 10 - Advanced Administering Linux...58
General...58
Hard Drive and Partitioning...58
Linux FileSystems (ext)...58
Boot Process...59
Startup Managers...61
Shells...63
Networking...63
Devices and Hardware...67
Sound...67
Printer...67
Performance monitoring...67
logging...68
Task Schedule...69
Scripting within the textual shell...69
Remote Connections to and from other computers...70
Graphics...70
Compile Programs...71
Chapter 11 – Troubleshooting...72
Chapter 12 - Backup Your Data...73
Rsync...73
One Way Backup...73
Rsync Local...73
Rsync remote...73
Rsync Deamon...73
Chapter 13 – Advanced Working with X Server...74
Running Graphical Applications Remotely...74
More Materials...75
Glossary...76

Preface

When i've started to study linux, I understand that there are a lot of terms to learn and there isn't one place to study from, so i've decided to write this guide.

This book will explain

- what Linux is and the story behind it.
- what is Ubuntu and how it's connected to Linux.
- how to work with the Graphical Desktop (GUI) and also with the Textual Environment (CLI).
- Essential Administering GNU/Linux Operating System

This book DO NOT intend to teach how to work with each GNU/Linux Distribution. They are somewhat different.
This book also DO NOT intend to teach how to work with standard GUI concepts such as managing windows (maximize, minimize, close....) or changing desktop background because such concepts are intuitive, especially if you've worked before with Microsoft Windows Operating Systems or Mac Os X. this manual is as focused as possible at the differences between Ubuntu and other Operating Systems.

For example, when you want to install an application/package in Ubuntu, the installation based on dpkg tool. In Red Hat in contrast, based on rpm tool.

Of course some of the things you will study here refers to shared components between different distributions, so you might be able to make use of this material with other distribution, but it's likely that you will get into troubles and won't know how to solve it, or there will be things that you won't know how to do there.

The theories refers to most of GNU/Linux distributions.

There are several rules I wrote and recommend to stick with:

- Rule #1 – RTFM, RTFM, RTFM, RTFM, RTFM, RTFM. Got it? RTFM!
  What is it? Search in Google to understand.
- Rule #2 – Study step-by-step. This book is ordered logically. You start with the GUI as a basic user and then continue to work with the CLI which is more advanced.
- Rule #3 – if you really want to remember how to use Ubuntu and you want it to be your main Operating System, install Ubuntu as a bootable OS or in multiboot configuration. Installing Ubuntu as a virtual machine means that each time you get into trouble, you'll go back to windows. Believe me – been there, done that. DON'T BE AFRAID :)
- Rule #4 – You have some question or you got into trouble and can't find how to solve it?
  Ask Ubuntu! It's a very good forum website build by questions and answers and each answer gets scores. Askubuntu.com. Open an account before you start, there are grades.

Good luck!

Chapter 1 – introduction to Operating Systems

So... what is an Operating System?

Operating System is a set of applications that allows you, the user, to use the computer easily.
It allows you to work with the computer without understanding how to work with your hardware directly.

Instead of working with 01010101010, you'll use it with a common language such as English or with a graphical interface using your mouse and keyboard.

Operating System consist of mostly the following components:

- **Kernel** – the first component/layer that run underline all other components. This layer responsible to translate all component's requests to the hardware (connects between the hardware and the softwares) language through drivers.
  Each hardware has it's own driver by type of the hardware, vendor, model and more.
  Mainly manage multitasking processes and memory management.
- **Drivers –** a low level program which connects the kernel to specific hardware. Each hardware has its own drivers; each hardware has its own “language” and the driver is the translator between the kernel and the hardware.
- **File System** – manages files and directories in a specific partition. Each partition holds maximum one File System. The access to the file system performs through the kernel. The kernel has a general structure to access the File System, in fact it's a module, so in that way you can switch file system and still access it the same way.
- **Shells –** User interfaces
  - **CLI** – Command Line Interface. The textual environment (shell) which allows you to run commands.
    In windows for some of you it's known as the “cmd”.
  - **GUI** – Graphical User Interface. Today's standard working environment. In windows it's the environment that includes the taskbar (with the start button), desktop and windows.
- **Basic Libraries** - C Library, GUI libraries. All of this in order to enable writing applications and use the OS infrastructure.
- **Basic applications** – those are built in applications inside your OS. text editors for example are built in default.

[add diagram]

we'll discuss all of them at this book, briefly or deeply.

Along the common operating systems you can find windows, MacOsX, Android (which based on linux kernel).

Chapter 2 - Introduction to Linux

Linux Kernel

Linux has started at 1989. in fact, if we look at the whole picture, it started at 1984. confused? I'll explain:

Linux - Unix-like kernel. ONLY THE KERNEL. it's not an operating system! It has developed by *Linus Torvalds* and the name of the kernel stands for **LIN**us **U**ni**X**. Linus started the development and completed at 1991. the kernel is still develop until those days.

GNU Project

Project for free Operating System. Develops the GNU system. started at 1984 by *Richard Stallman*.

- originally developed the GNU Hurd Kernel. in 1992 GNU Project chose Linux as the kernel of their operating system.
- If you're more interested in details, try reading http://en.wikipedia.org/wiki/Linux
- Gnu System - Operating system based on Hurd Kernel.
- GNU/Linux System - Operating System based on Linux kernel. most of free Linux based OS are variants of GNU/Linux.
- Operating system based on Linux. philosophy of free software. it's a project.
- Ubuntu derives from GNU project.
- GPL - General Public License. Allows
  - copy the software
  - get the source code
  - alter the source, recompile it and distribute it
  - charge money for it
  - do not allow to change the license
  - your name stays by the feature you've created
- FSF - Free Software Foundation - corporation founded by Richard Stallman. funding free software, mainly GNU System. funded the beginning of Debian GNU/Linux
- Linux can run on variety architectures such as x86, Alpha and more.
- GNU/Linux is Multitasking nad Multi-User environment:
- Mutitasking – the ability to run concurrent applications.
- Multi-User – concurrent users can works simultaneously on the same machine. (Will be discussed later)
- Everything in GNU/Linux can be done by running commands. It means that if you have some action that can be done at GUI can be done from Command Line Interface.
  Of course people can build an application which can function only from GUI, but the OS functions can be done both from GUI or CLI.
- GNU/Linux architecture based on layers. For each layer there are modules that can implement the layer's role. Even the GUI is a module, consist of several modules (Will be discussed later)
- Tux – the mascot of GNU/Linux

Why choose GNU/Linux Desktop Distribution

Advantages

Linux	Other Competitors
Look-and-feel	
Dynamic – you can change your whole GUI beahviour easyly	One GUI Engine. You can change your GUI Components such as icons, colors and so on, but you cannot change to another GUI.
Free!	Not Free, such as Windows and Mac OSx
Application installation is awesome! Just choose the application you need, and choose to install it. The download and installation process are automatic.	In windows for example, you need to download the installation (sometimes it's annoying), and then to start the installation (usually installshield), press next next next next.... you know what :)
Lightweight. It does not consume huge amount of memory. As for the CPU, it depends what kind of Desktop Environment you chose	CPU and Memory intensive. You can change the details level but the differences are not great.
Stable Kernel – the core of this operating system is very robust.	Blue screen? Although windows today is more robust than before
For most of the common application there are alternatives in Linux, or the same application in Linux version.	You have a variety of applications, but for most of them you need to pay.
Security – very securely OS. Try to search for anti-virus for Linux. Most of pages you will find says that Linux doesn't need anti-virus	A lot of backdoors.
Linux Does not need defragmentation process. Its Filesystems are smart enough to keep files contagious	Windows needs defragment
Because the source code files are available, it's easier to adapt programs to both 32bit and 64bit architecture	
Workspaces – linux has several workspaces, so you can put your windows grouped by type. Easier to manage.	No workspaces
Control on the OS – in Linux you have better control on the OS. It means you can do almost anything. You can even build your own GUI	You cannot change basic components in the Operating System

Disadvantages

Linux	
Harder to maintenance – if you are not a power user you probably get stuck	Easy to manage
There are applications that haven't been written to run in Linux	
The best games are not available for linux – if you are a gamer, this platform is not ideal for you	
Office Alternatives – in my opinion, all Microsoft	

Office alternatives aren't good enough. Microsoft Office still does better job	
File Sharing are harder and not always out-of-the-box.	

Chapter 3 – What is Ubuntu?

Ubuntu is an operating system based on GNU/Linux.

In fact, is one of the distributions, or distros.

Distribution

[add diagram?]

Linux operating system which consists of specific components.

- **Desktop Environment** – the component which determine how your Graphical User Interface will look. (will described later)
- **Applications** – each distribution comes with its own applications for the user.
- **Package manager –** a program which pack an application into one file (package) with smart installation instructions. A package manager has many more advantages such as link dependencies between packages, so If you're going to install a package which rely on other package, it will be installed also.
- **Software / Installation manager** – advanced package manager. In fact it uses package manager but also responsible to downloading the packages you need and install them. All you have to do is to tell it which software you want, choose install and that's it!
- **Graphical services** – most of distributions (if not all) based on X Server (will be described later) which has two main versions: X.Org and XFree86.

of course some of this components are not unique for a specific distribution. There are some applications that in use in several distributions, and the same about desktop environment – there are some distributions that can look the same.
What is always common to all of them – they use the same kernel, Linux.

There are two main types of distributions:

1. **Desktop** – operating system for the end user. Includes applications such as multimedia applications.
2. **Server** – operating system for servers. Some of them include Graphical User Interface, Some of them not.
   There are some packages which come with them to serve other computer: file share server, vpn server, web server and so on.

General

Ubuntu is the desktop version to Debian distribution. It relies on most of the standard applications, components and architecture that Debian has.

What does Ubuntu includes?

- **Software manager** – Ubuntu software center
- **Desktop Environment** – Unity
- **Package Manager** – Debian Package (or Dpkg)

- **X Server** – X.Org
- **Applications** – LibreOffice, Empathy, Mozilla Thunderbird and more.

Chapter 4 – How to install Ubuntu?

First, what you have to do is to download the ISO file http://www.ubuntu.com/download/desktop.

If you have an 64bit CPU, I recommend to download 64bit version. If not (or if you don't know), download the 32bit version.

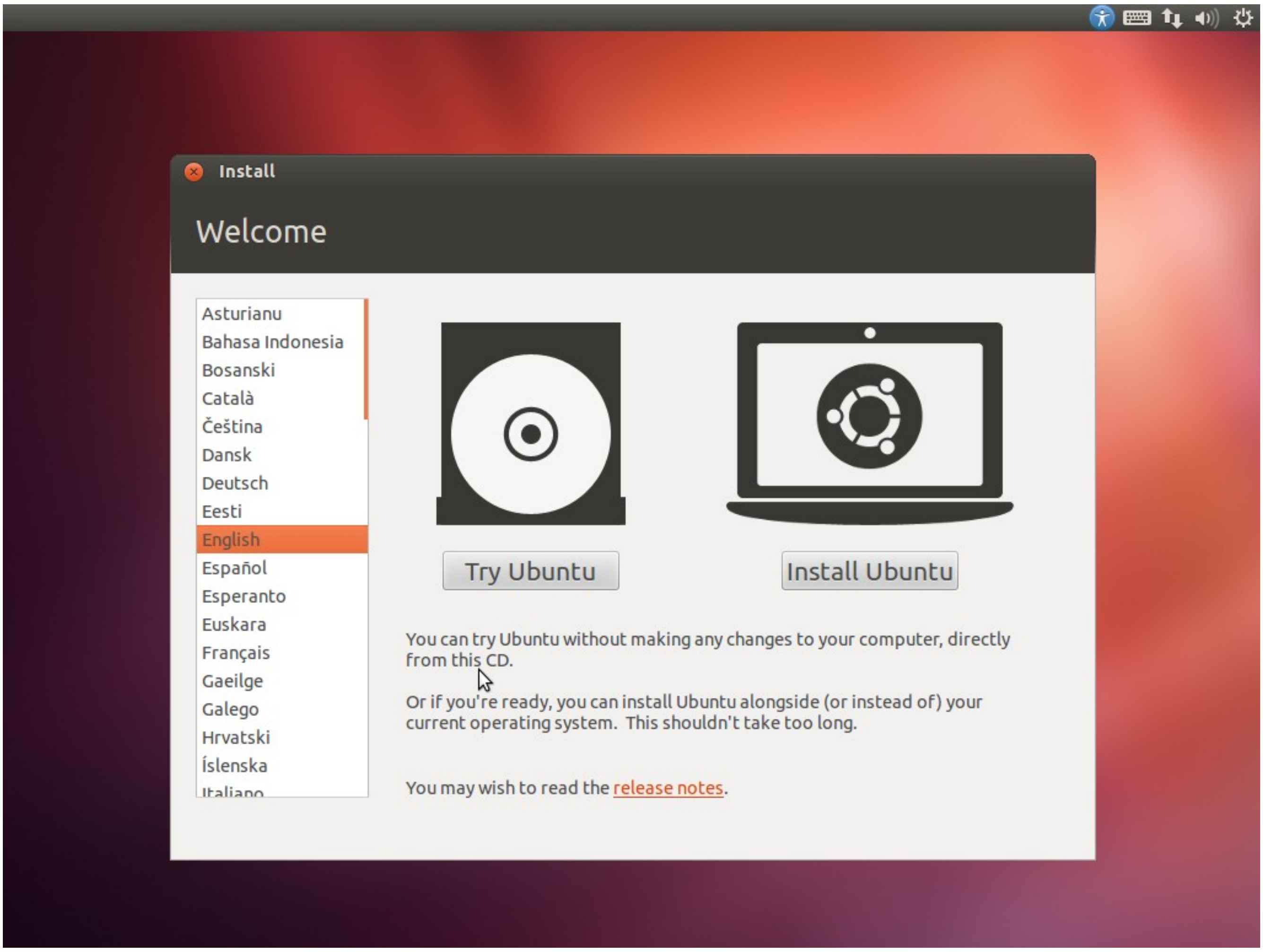
After you've downloaded the file, you have two ways to install Ubuntu:

1. burn the ISO file to CD.
2. Create a bootable USB stick (I recommend this way, it's easier in my opinion).

Those two options are explained in http://www.ubuntu.com/download/help/try-ubuntu-before-you-install.

Some important tips

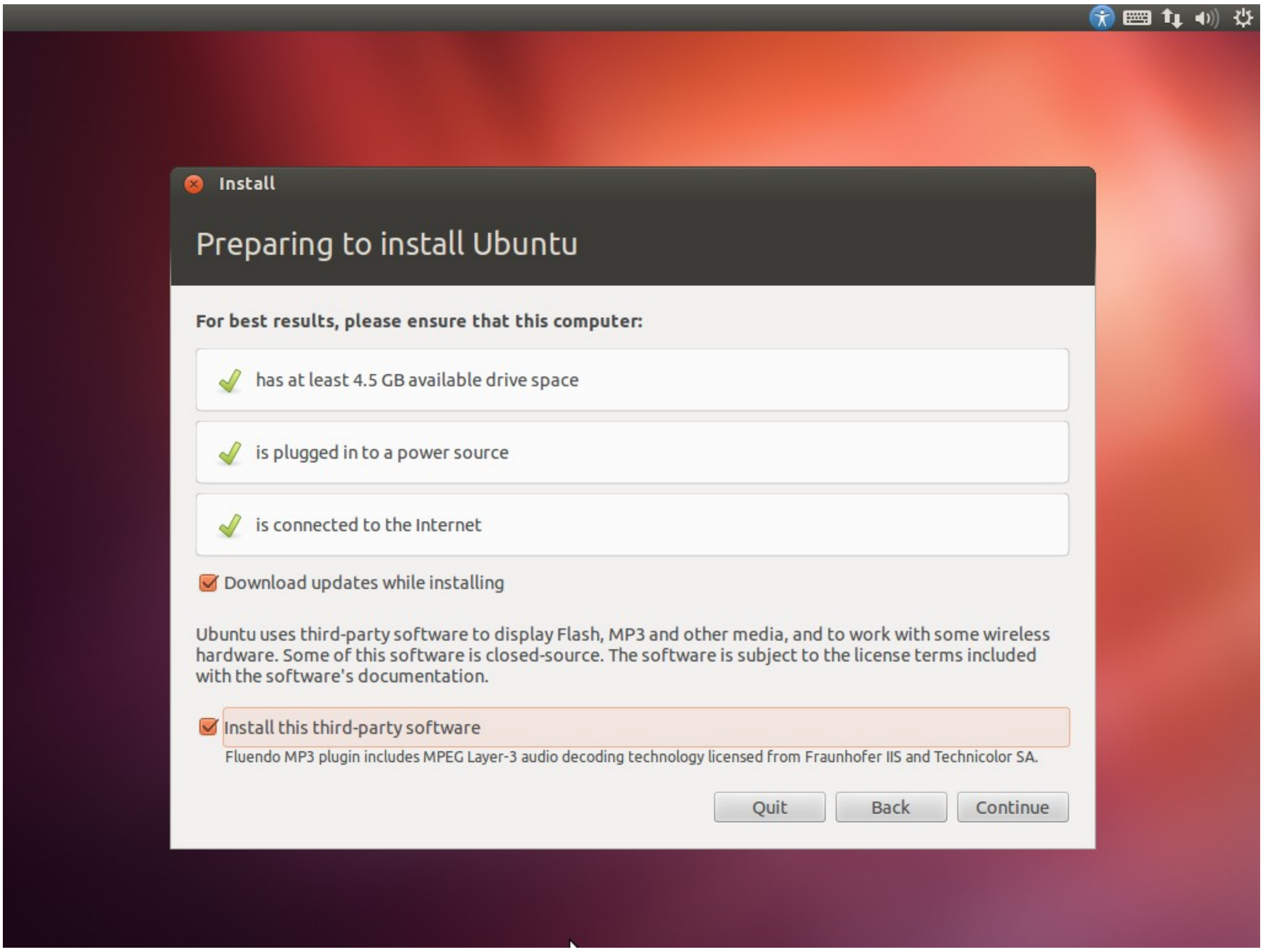
1. Divide your hard drive to 2 partitions at least: root and home.
   The root partition will hold most of your data such as system configurations, the operating system itself, applications.
   The home partition will hold your personal data (similar to user data / documents and settings in windows) plus your custom configurations to your GUI.
2. If you're installing Ubuntu aside to Windows
   1. resize your hard drive before installing Ubuntu.
      You can use Partition master, a free tool by Easeus – http://www.partition-tool.com/personal.htm.
   2. Install the boot loader on root's partition and not on your MBR (shown as the hard drive and not a partition). This will keep your default windows boot loader, so in case of a problem it will be easier to solve it. If you're an expert, install it on your MBR. It will install GRUB there (will be described later at chapter 10)
3. choose to download updates.



You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.



Preparing to install Ubuntu

For best results, please ensure that this computer:

✓ has at least 4.5 GB available drive space

✓ is plugged in to a power source

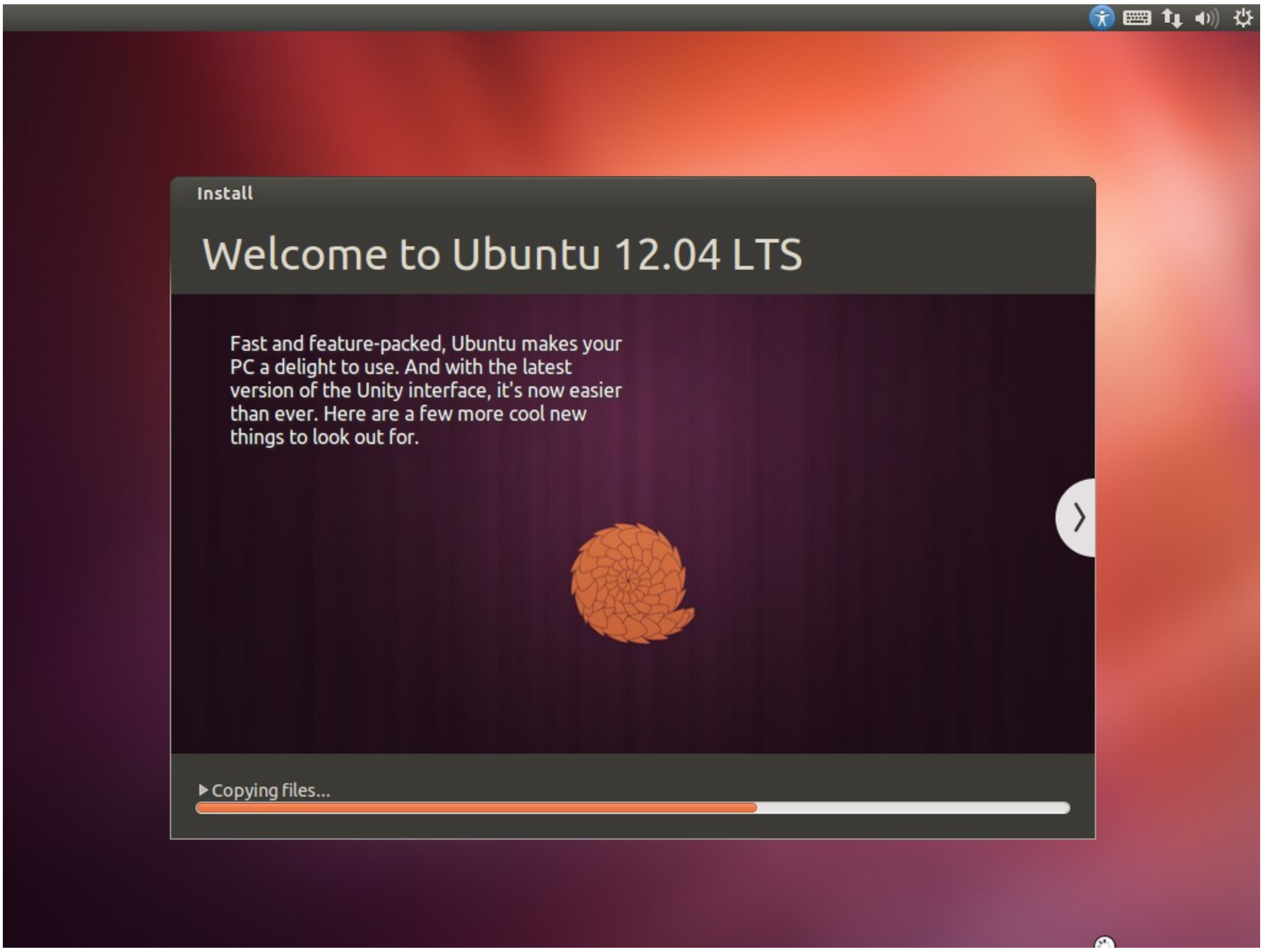
✓ is connected to the Internet

☑ Download updates while installing

Ubuntu uses third-party software to display Flash, MP3 and other media, and to work with some wireless hardware. Some of this software is closed-source. The software is subject to the license terms included with the software's documentation.

☑ Install this third-party software

Fluendo MP3 plugin includes MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Technicolor SA.



Welcome to Ubuntu 12.04 LTS

Fast and feature-packed, Ubuntu makes your PC a delight to use. And with the latest version of the Unity interface, it's now easier than ever. Here are a few more cool new things to look out for.

▸ Copying files...

Chapter 5 - Using Ubuntu GUI

What is Desktop Environment?

A Desktop Environment is a set of objects which make your GUI the way it is.

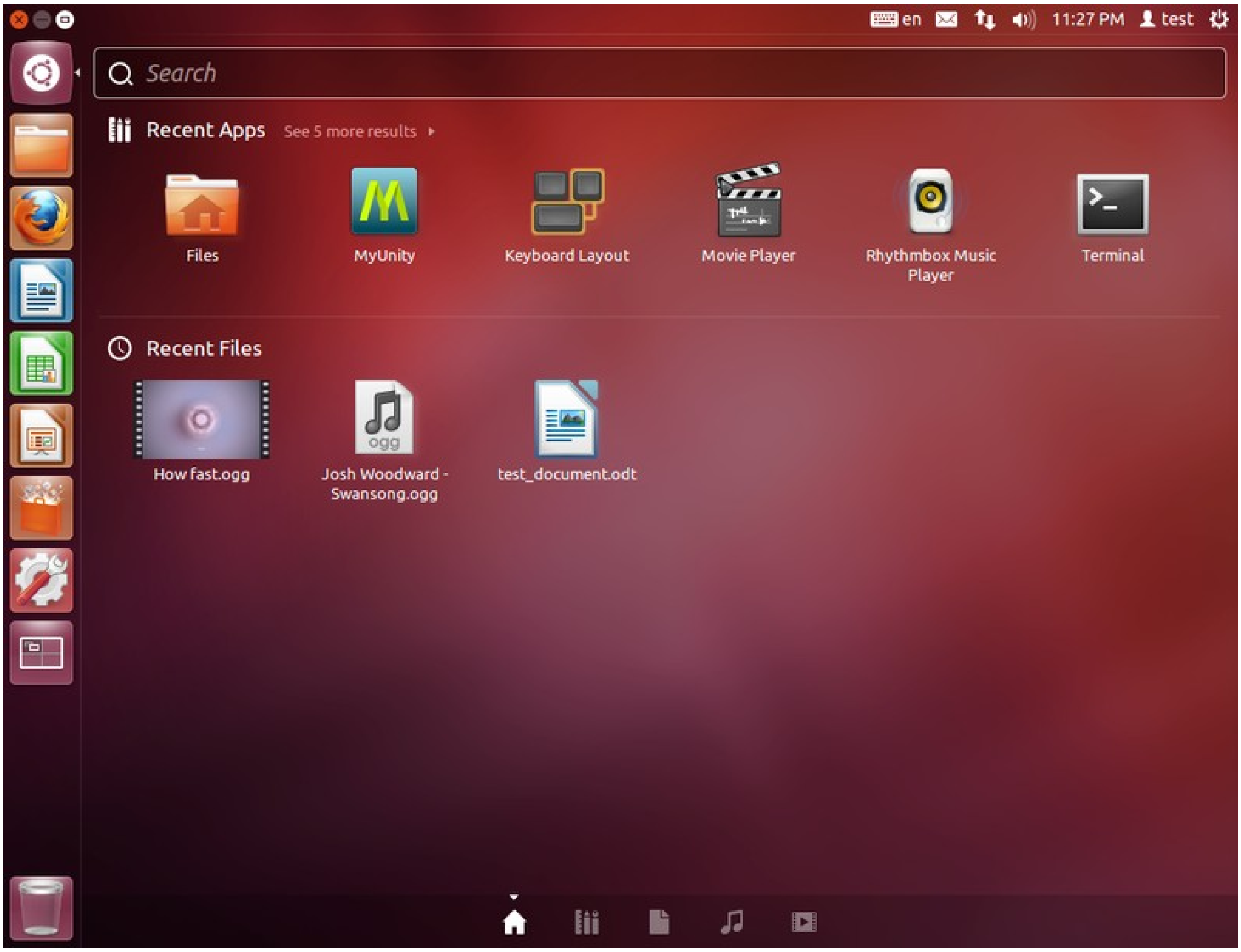
Each Desktop Environment has its own look-and-feel. It means that it might has its own icons, own Panels/Bars, special applications and so on.

- Manage Clipboards
- Comes with lots of applications as a suite. (Image apps, email and internet apps, extended taskbar, administrative tools and so on).
- Based on specific window manager. In fact, wrapping a window manager, uses it and extends it.
- Manage workspaces
- common objects – Workspace Switcher.
- Applets – small programs which has shown in the Top Panel.
- Launcher – shortcut to a file. Launch the program which opens it, or launch the executable itself.

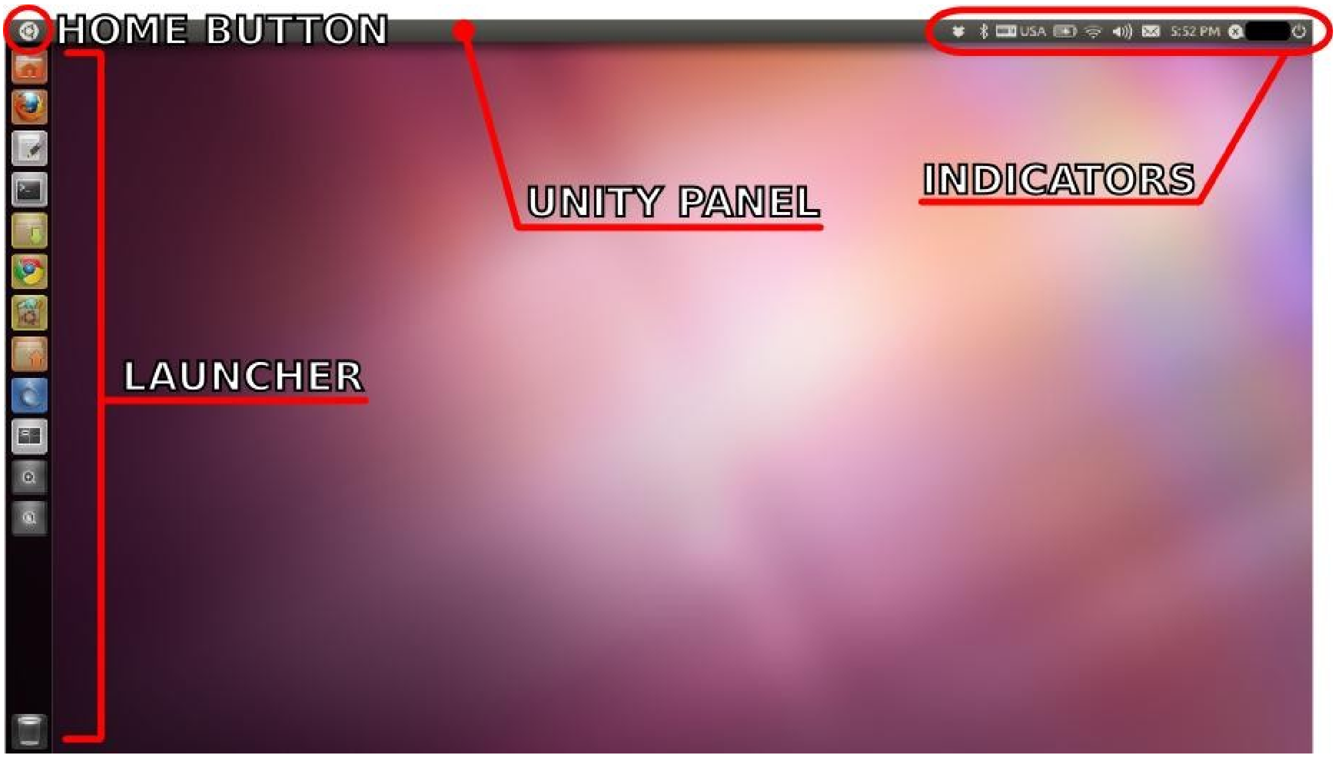
The default Desktop Environment in Ubuntu, starting at version 11.10, is Unity.

Unity Desktop Environment

Unity is based on Gnome Desktop Environment.

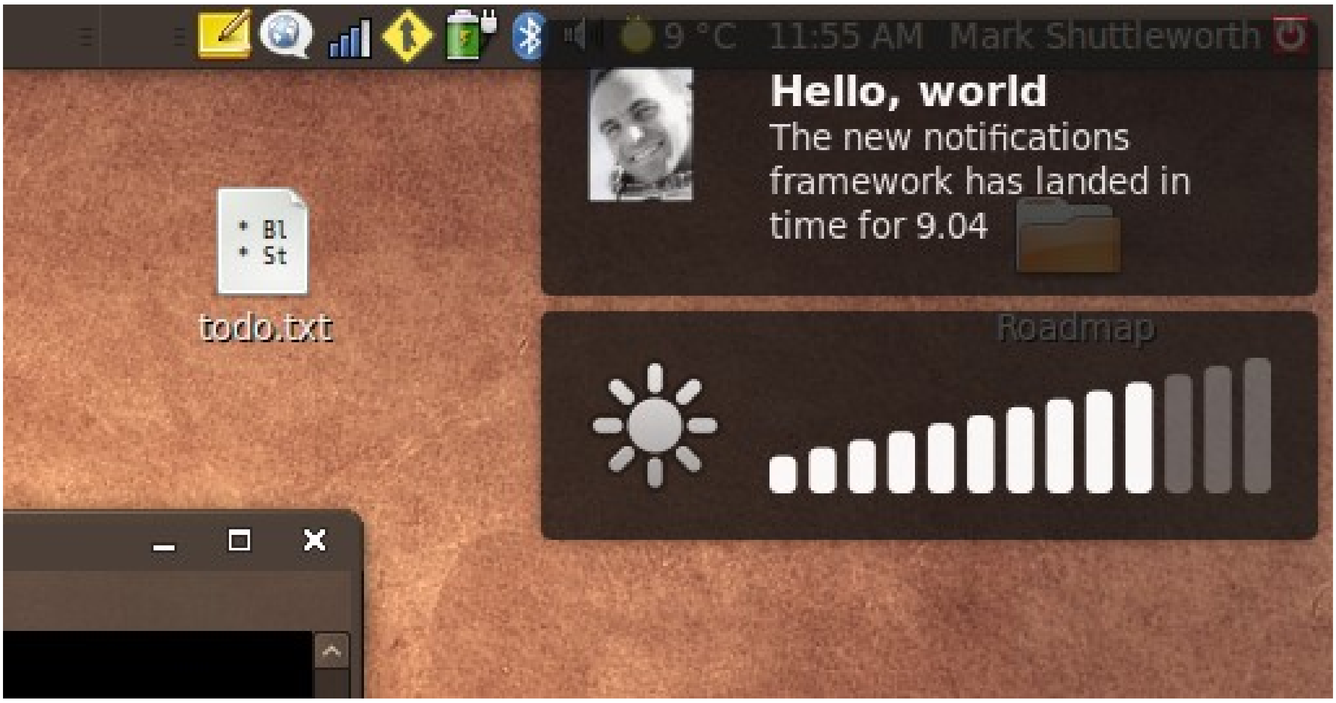


Unity Desktop Environment



The Gnome components:

Notify OSD



Hello, world

The new notifications framework has landed in time for 9.04

Notify OSD with notification bubble and Icon bubble

shows notifications with the following types:

1. confirmation bubbles – the first notification in the next figure. Notifications about changing volume, brightness, etc.
2. notification bubbles – the second notification in the next figure. Notifications about new message and body of the message.
3. Icon – when you're doing some action that is on or off such as disable touchpad.

Top Panel
[add picture]

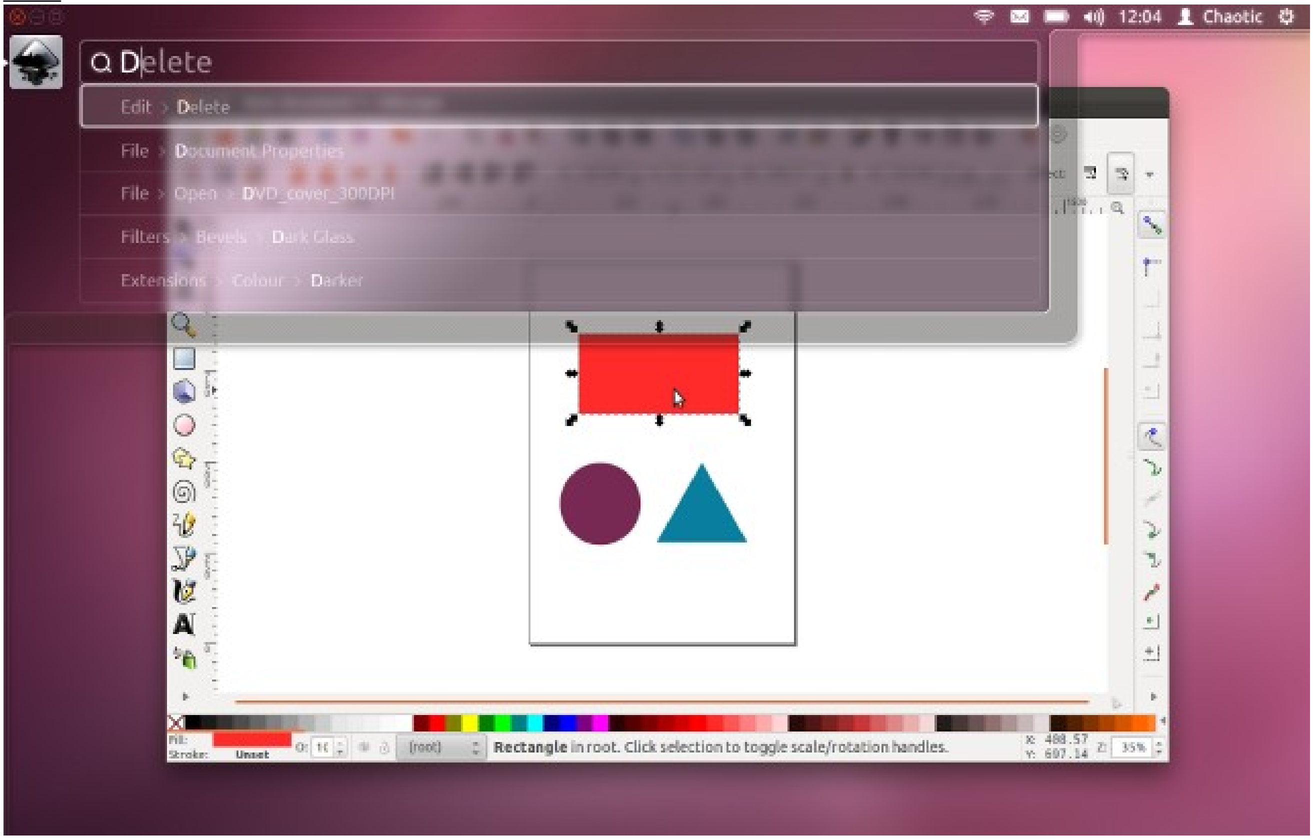
a panel which consists of:

- current window menu (instead of placing the menu at the window's top panel)
- indicators
- Off Button – allows you to do things such as log off, change system settings and more.
- Me Button – shows who is connected right now and allows you switch user
- Time indicator – shows the time
- Sound management
- Network management (Wired / Wireless)
- Bluetooth Indicator – manage bluetooth settings and connections
- Battery Indicator – shows the battery status and can show time left until the battery is empty.
- Indicator Applet – allows you to
    - connect to Instant Messages (IM) such as Facebook, skype, live messenger and more.
    - Mail
- Notification Area – area that appears when a new message arrives. Messages can be IM message, new mail, warnings, new track is been playing.
- indicator framework (libindicator)
    - application
    - sound
    - messages
    - session

Workspaces
[completion needed]

So what special about Unity?

HUD

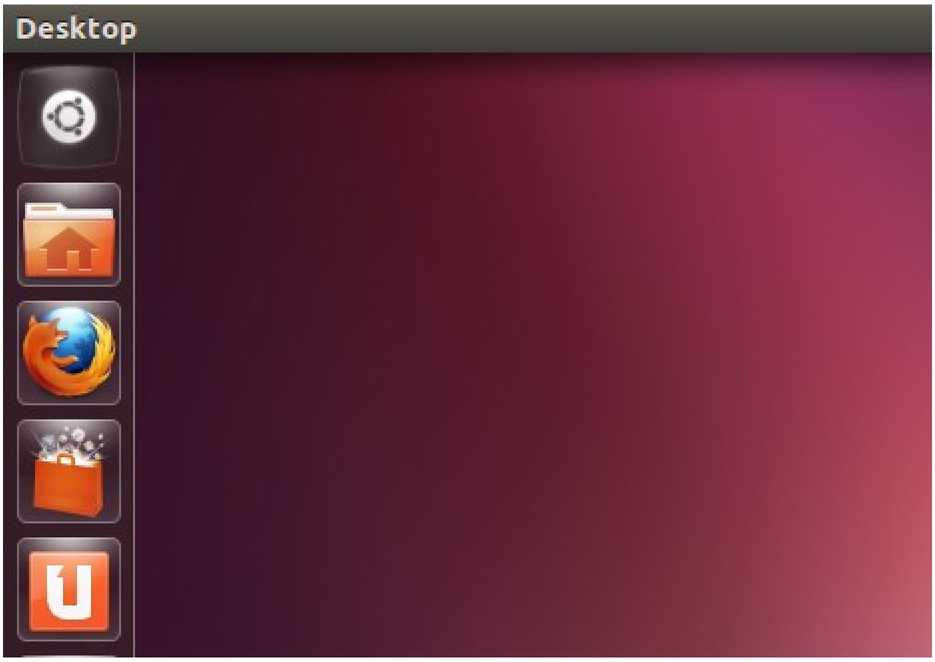

12:04
Chaotic
Delete
Edit > Delete
File > Document Properties
File > Open > DVD_cover_300DPI
Filters > Bevels > Dark Glass
Extensions > Colour > Darker
Fill:
Stroke:
Unset
(root)
35%

Rectangle in root. Click selection to toggle scale/rotation handles.

Heads-Up Display application search menu

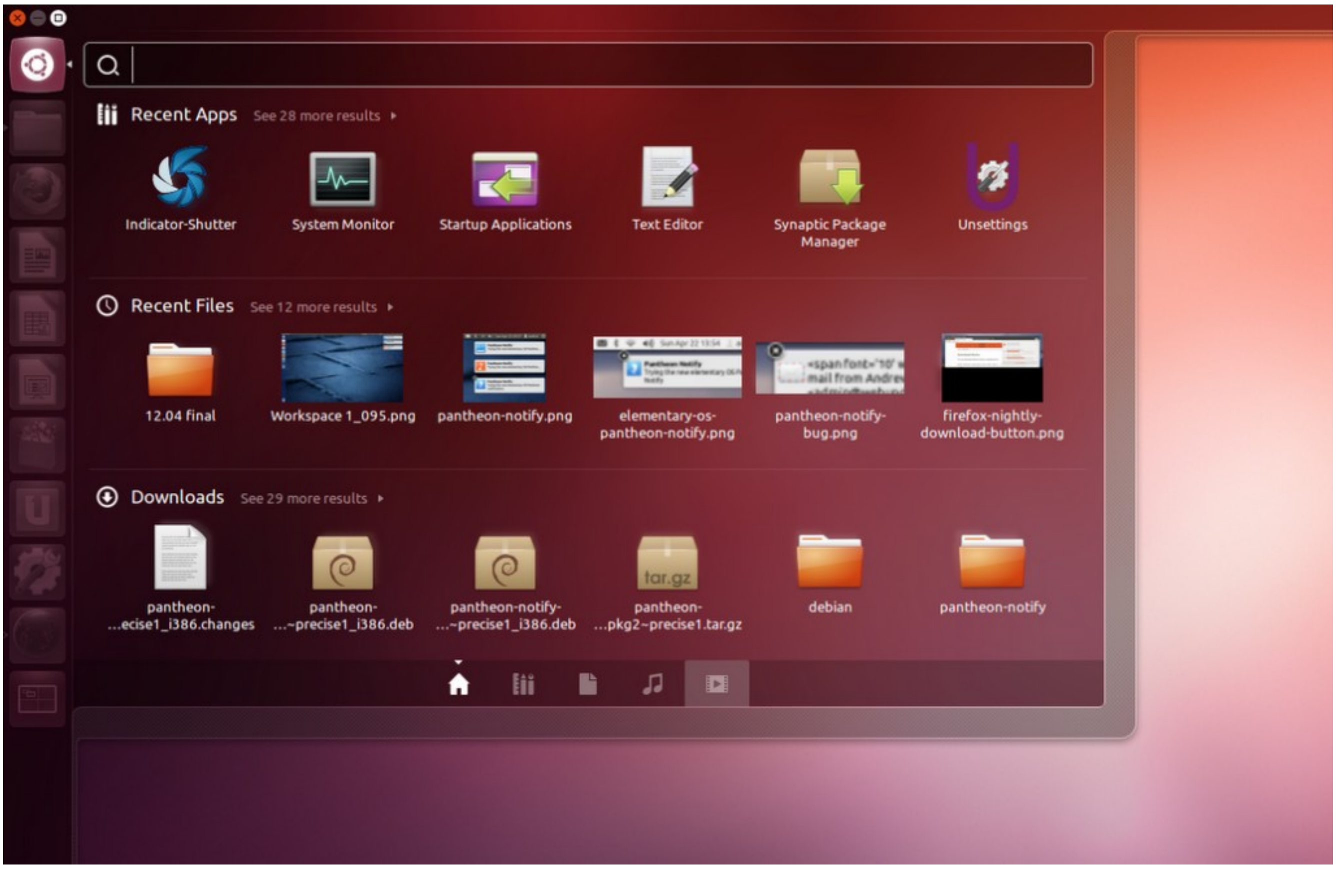
Heads-Up Display – ubuntu's tool for accessing and searching within the menu bar. Started at version 12.04.
when you press the alt key, a search bar will appear at the top of the screen, then you can search the choice you want by writing the choice's name.

Launcher



Unity has the Launcher panel, which allows you to search for programs faster. It's similar to windows taskbar but only to the middle area.
the panel at the left side of the desktop by default.

Dash



The dash is the new start menu. Instead of searching within a menu for your application, you just type your application's name and find all of the applications which match the criteria.

How To Run Applications?

Just push the Super button (Winkey) and write the name of the application you're searching.
Then the search will filter the applications by your phrase

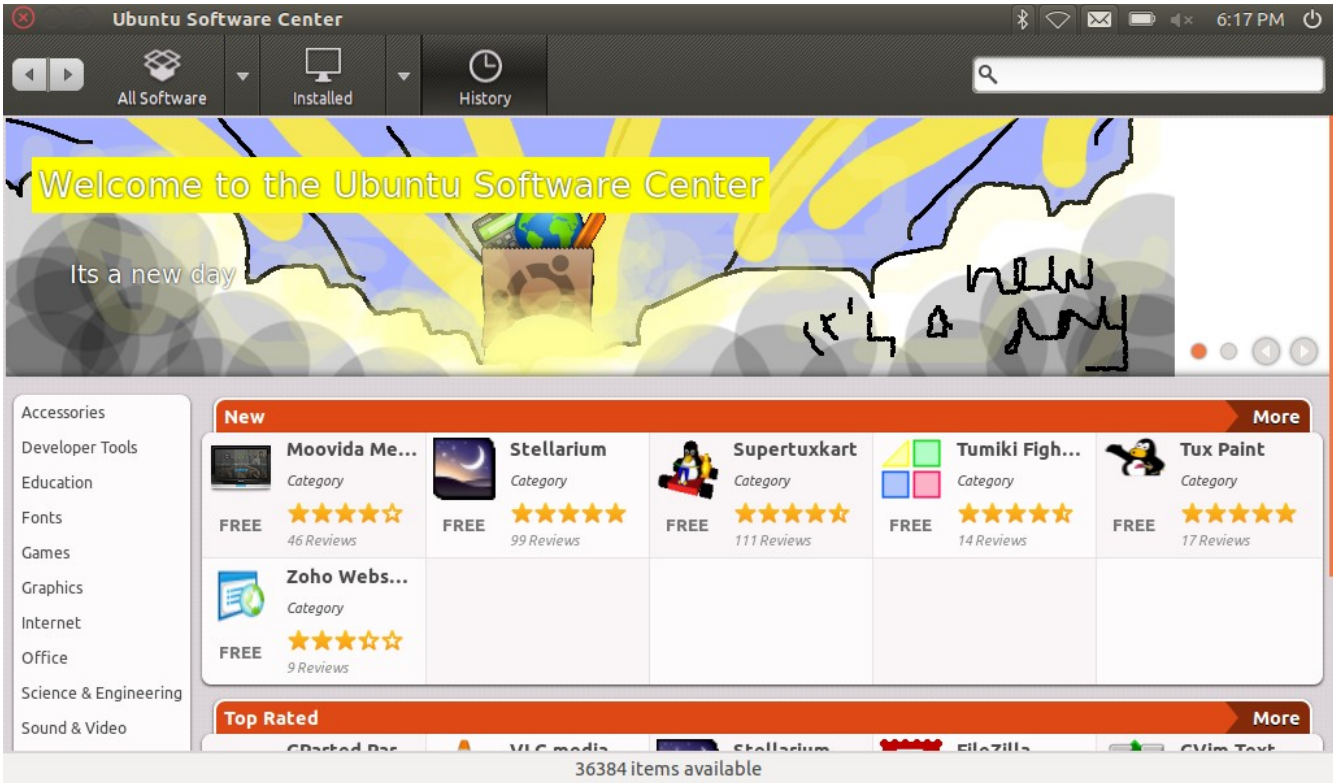
[add picture - example]

Built-in Applications

Application	Description	You probably used to work in windows with
Empathy	A very nice chatting/Instant Messaging application. It can connect to most, if not to all of the IM protocol such as Facebook chat, Skype, Live Messenger, Google Talk and so on.	Nothing to Windows Live Messenger
Mozilla Thunderbird	Mail application. It can connect to Pop3 and Imap mail servers such as Gmail.	Microsoft Outlook (only the mail component)
Nautilus	The default file explorer in Ubuntu. Easy to use and has advance features that Windows Explorer doesn't have such as Extra Pane (explore in two directories side by side)	Windows Explorer
Banshee	Music library player. Working with watchdog on a specific folder and updates each time you add music.	Windows Media Player / iTunes
Mozilla Firefox	Internet browser. You're probably familier with this application because it has windows version.	Internet Explorer, Firefox and Chrome
LibreOffice	A complete office tools for creating document, spreadsheets, presentations and so on.	Microsoft Office (2003/2007/2010)
Transmission	Torrent client	Utorrent
Update Manager	Application which checks for updates for the OS as well as for your applications	Windows Update
Shotwell		
Archive Manager	Extracting compressed files	
Gedit	Plain text editor but with features	Notepad

Application	Description	You probably used to work in windows with
	like Highlight mode (for highlighting reserved words for scripts for example)	

Installing New Applications



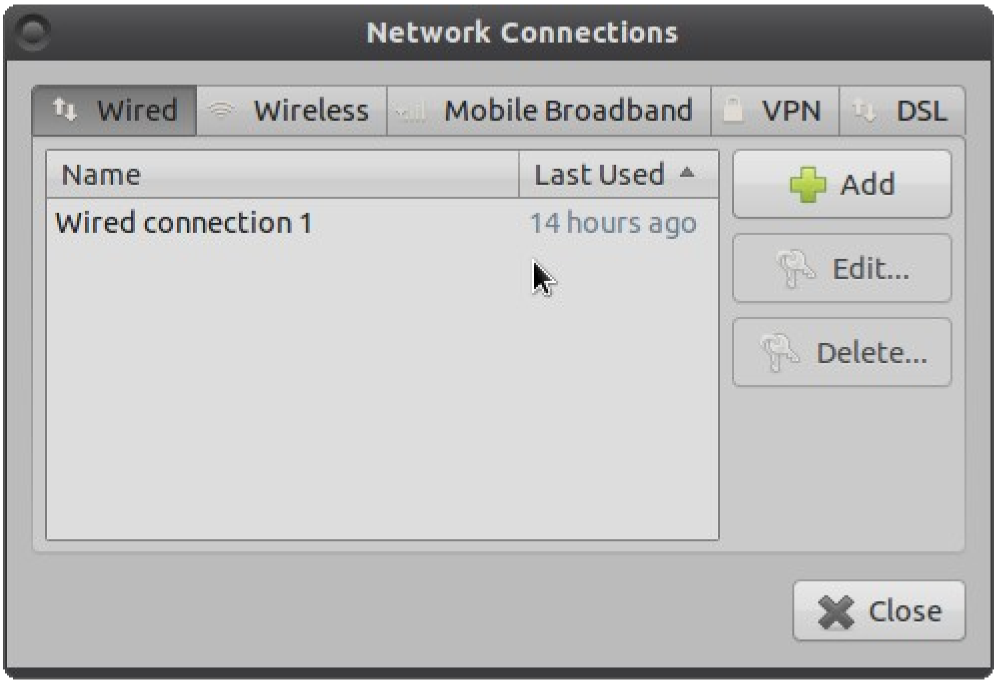
Ubuntu Software Center

To install new applications, all you have to do is to run Ubuntu Software Center, choose the application you want to install, and press the “Install” button. Pretty easy, ha?

Recommended Applications to install at the beginning

Application	Description	Apt Package Names	Require updating sources.list?	You probably used to work in windows with
Google Chrome		google-chrome-stable		
Skype		skype		
Cairo Dock		cairo-dock		
Rar		unrar unrar-free		

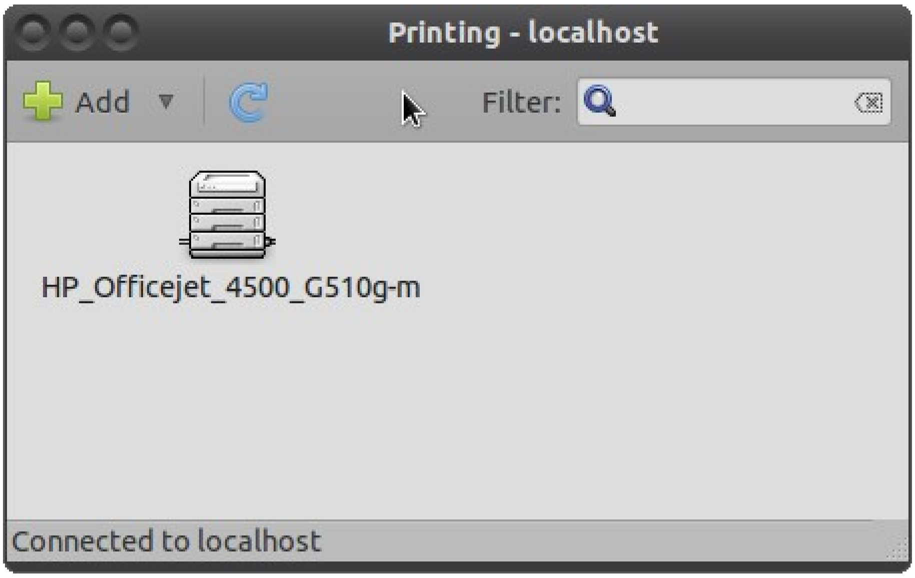
Configuring The Network



Name	Last Used
Wired connection 1	14 hours ago

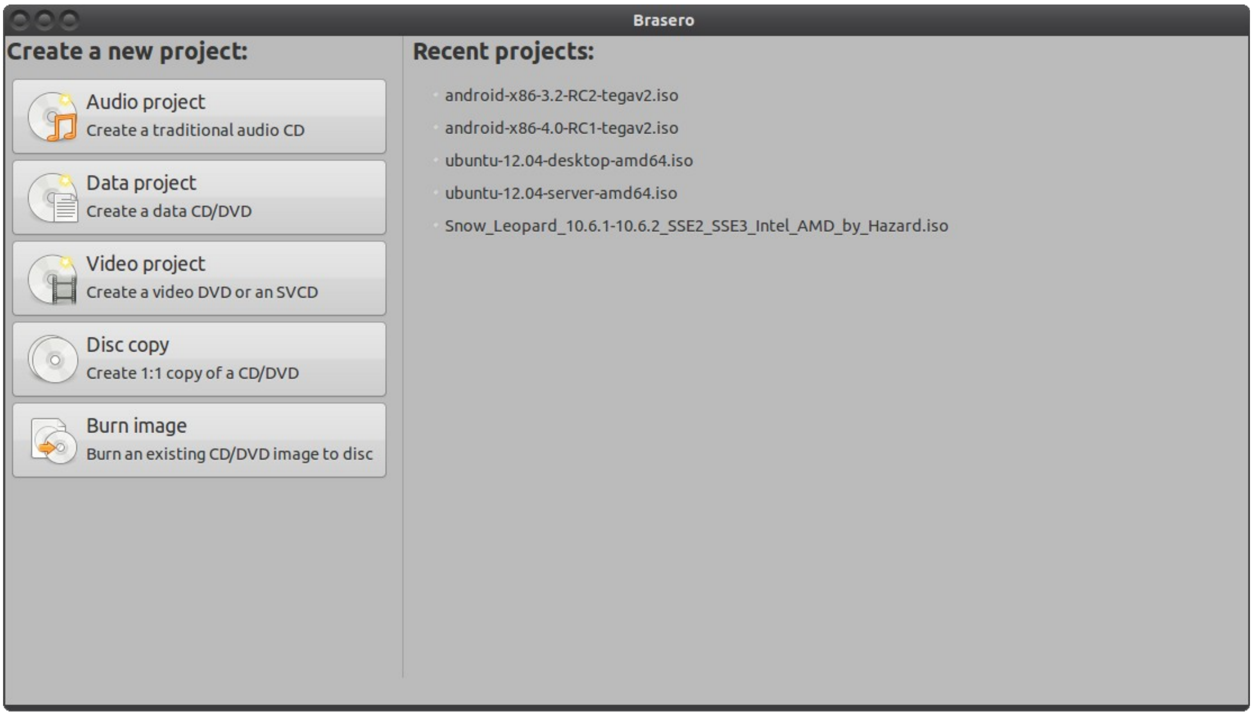
From this application you can configure your network settings, such as your IP (Static or Dynamic).

Configuring Your Printer



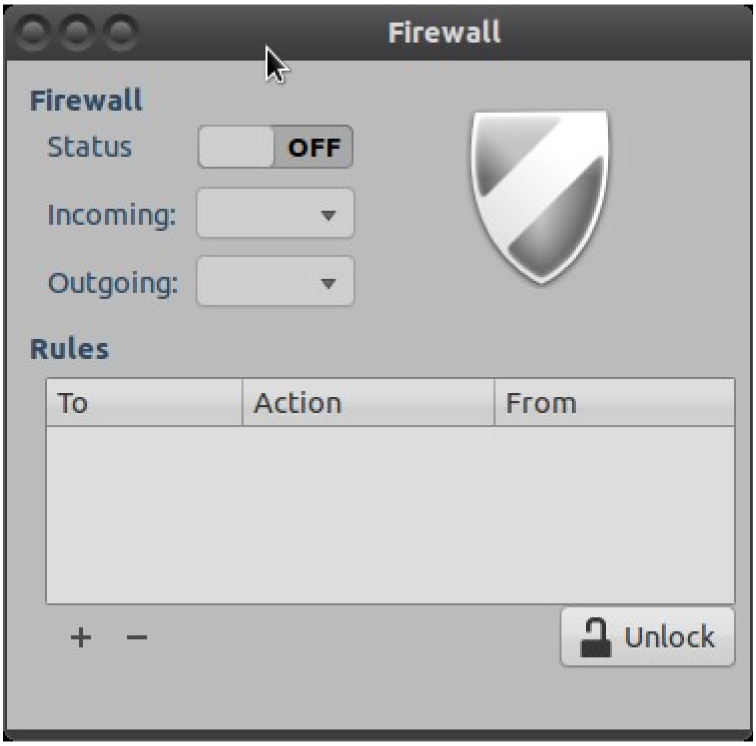
Connected to localhost

Burning Tools


Brasero
Create a new project:
Audio project
Create a traditional audio CD
Data project
Create a data CD/DVD
Video project
Create a video DVD or an SVCD
Disc copy
Create 1:1 copy of a CD/DVD
Burn image
Burn an existing CD/DVD image to disc
Recent projects:

- android-x86-3.2-RC2-tegav2.iso
- android-x86-4.0-RC1-tegav2.iso
- ubuntu-12.04-desktop-amd64.iso
- ubuntu-12.04-server-amd64.iso
- Snow_Leopard_10.6.1-10.6.2_SSE2_SSE3_Intel_AMD_by_Hazard.iso

Firewall



Status: OFF

Gufw – Graphical Uncomplicated Firewall

Installation

```
sudo apt-get install gufw
```

zeitgeist

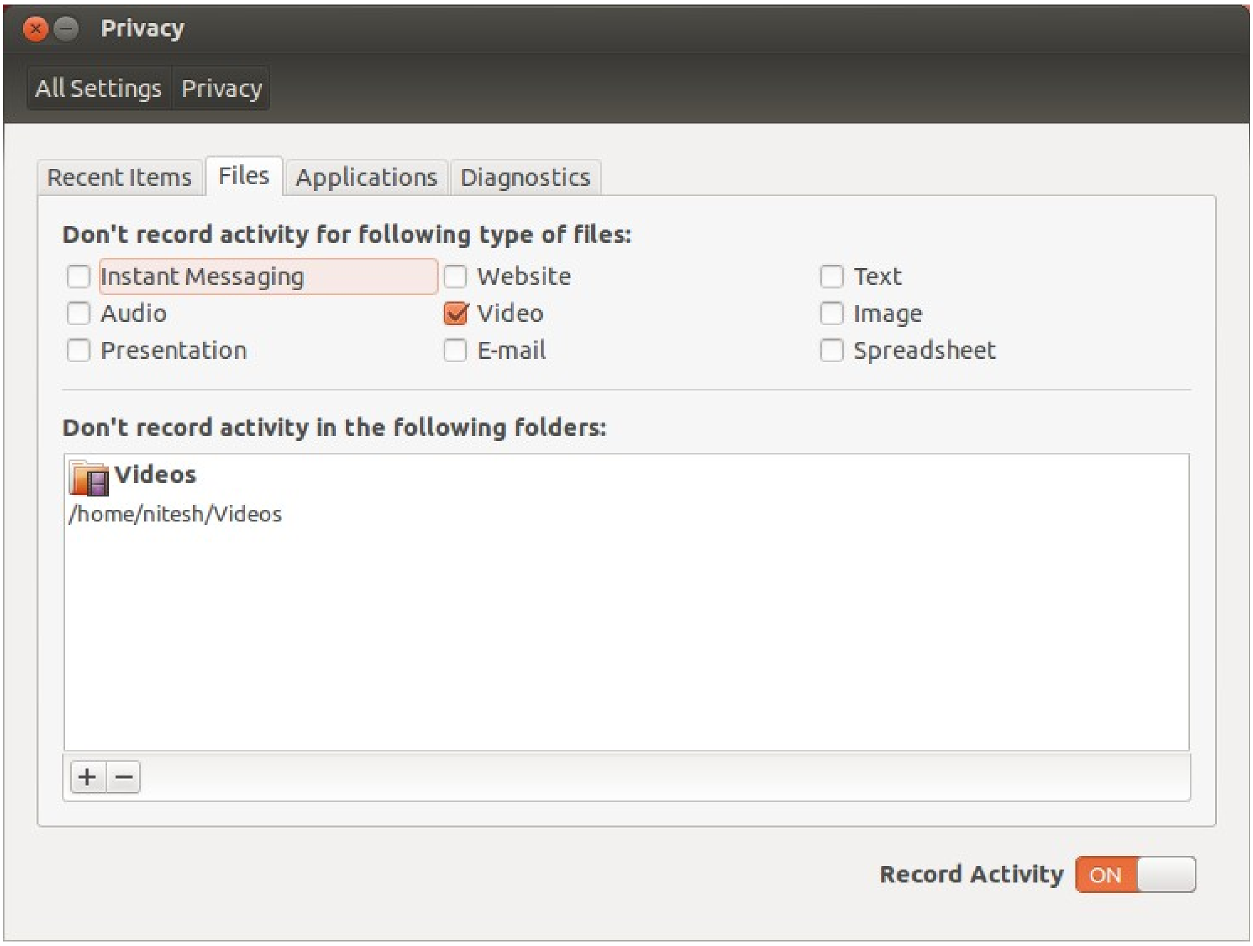
what is zeitgeist?

Zeitgeist Is a daemon which collects information (aka logging) about your activity – which application used recently, recent files opened and more (those things are shown at the unity dash).

Unity dash is lying down on that service.

Zeitgeist action in Ubuntu called “Record activity”

System settings → privacy



Don't record activity for following type of files:

- □ Instant Messaging
- □ Audio
- □ Presentation
- □ Website
- ☑ Video
- □ E-mail
- □ Text
- □ Image
- □ Spreadsheet

Don't record activity in the following folders:

Videos
/home/nitesh/Videos

Record Activity ON

you can turn record activity off to disable logging. (does it disable indexing?)

To disable record activity on specific mountpoint, just add the mointpoint to the exclusion list (“don't record activity in the following folders”)

it runs behind the scenes indexing service called FTS++

daemon - /usr/bin/zeitgeist-daemon

indexing service - /usr/lib/zeitgeist/zeitgeist-fts zeitgeist-datahub - ?

Useful Shortcut Keys

Shortcut	Description
Super (Winkey)	Reveal the launcher
Ctrl+Super+D	Show/Hide desktop
Ctrl+Alt+Shift+Arrow	move a window from one workspace to another
Ctrl+Alt+Arrow	change workspace
Ctrl+Page Up/Down	move between tabs
Ctrl+Q / Alt+F4	quit application/window
Alt+F2	Run a command
Ctrl+Alt+L	Locks the computer
Ctrl+Alt+Delete	Log out from the current session
Ctrl+Alt+T	run terminal (CLI)

Chapter 6 - Advanced Ubuntu GUI use

Background

In order to make great tweaks in your GUI, it's better to know how the GUI works in GNU/Linux.

[add diagram]

The following text will explain the main GUI components:

X Window System

XFree86/Xorg - X Window System. the base for the GUI in GNU/Linux (Xorg is in use with ubuntu)

we have already said that X Window System cannot do anything by himself. He needs a window manager - component which manage windows in X server.

[add diagram]

X Server

X Server is the side which is connected to the console – keyboard, mouse and screen.

X Client

X Client is the side which runs applications but gets the input from X Server and sends the output to X Server.

Window Manager

- Determine where each application will be located in X Server.
- Determine which application will be on focus and though will get the input from the keyboard.
- May include taskbar.
- May include basic menu.
- If you would like to demonstrate how window manager looks, you can install fvwm window manager for example:
  sudo apt-get install fvwm.

Examples of window managers

- **Compiz** – 3D window manager, also called compositing window manager. Uses OpenGL (which uses 3D hardware acceleration). Unity uses compiz.
  - Desktop Cube and Rotate Cube (the cube effect which turns your workspaces into cube) – stick to This Guide!. BEWARE, if you won't stick to this guide, compiz can break your desktop environment, believe me, been there done that.
  - If you can't fix things after you've changed the compiz settings, go to the terminal by Ctrl+Alt+T and write “unity –reset”. It will disable your compiz settings.

- Metacity – 2D window manager. (gnome2 uses it) EXISTS UNTIL 12.10 (OR 12.04?)
- mutter – 3d window manager for gnome 3. based on metacity and clutter. Support openGL.
- Kwin – KDE window manager. Support openGL (version 4 and later)
- xfwm – XFCE window manager (2D)
- Openbox – LXDE window manager (2D)

to replace window manager, use –replace parameter. For example, to replace your current window manager with metacity window manager -

metacity –replace

other options:

xfwm4 –replace

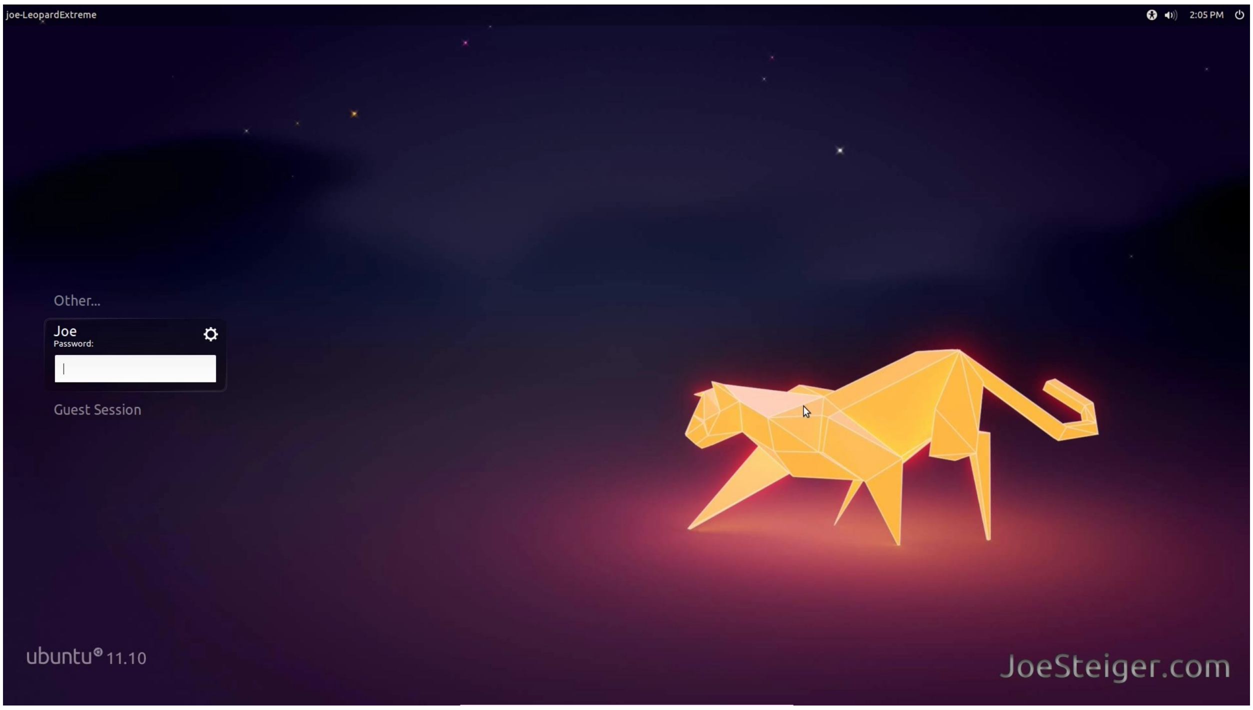
compiz –replace

and so on...

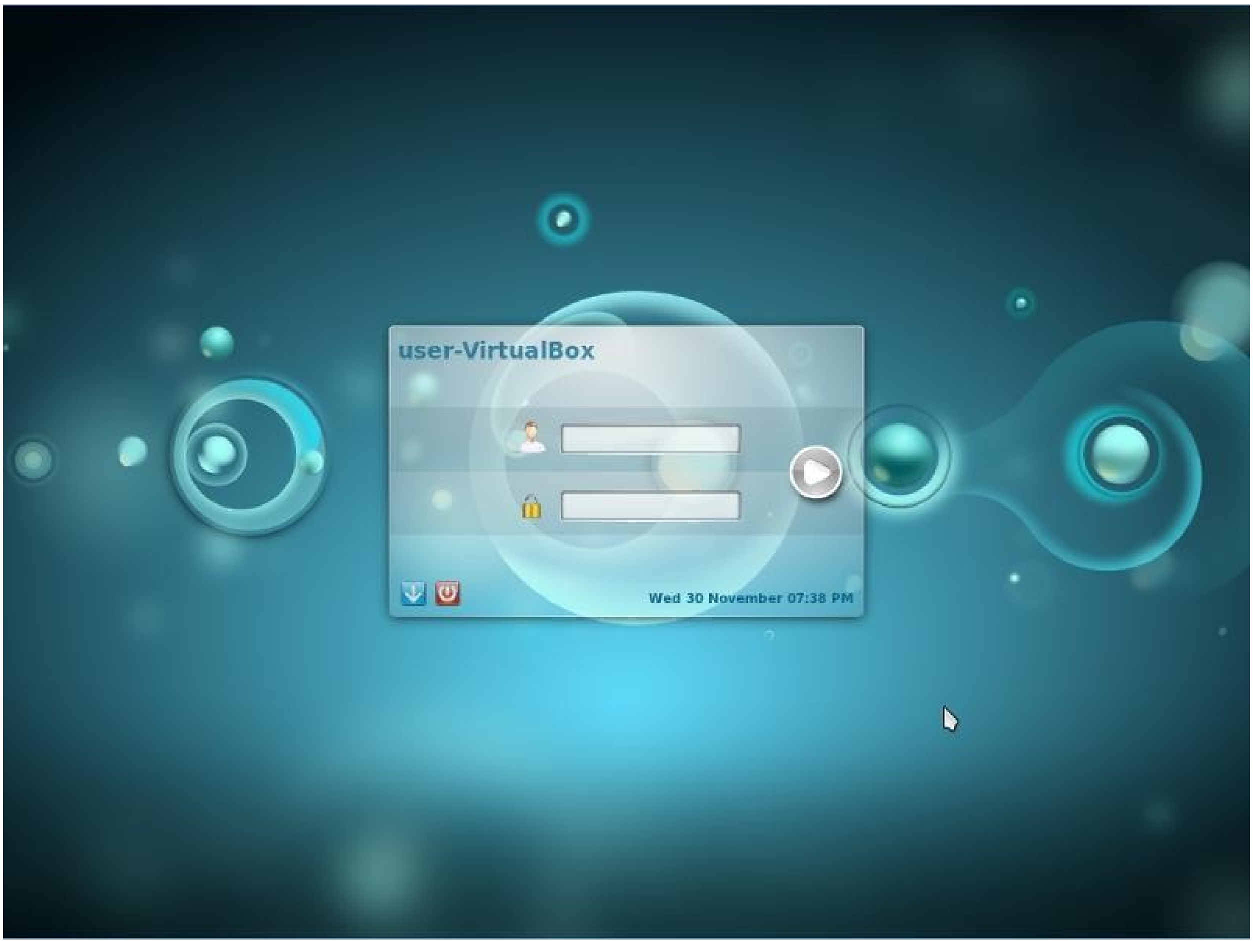
Display Manager

Display Manager is the startup login screen, runs the display environment as a decsandant.
Display Manager is also called Session manager.

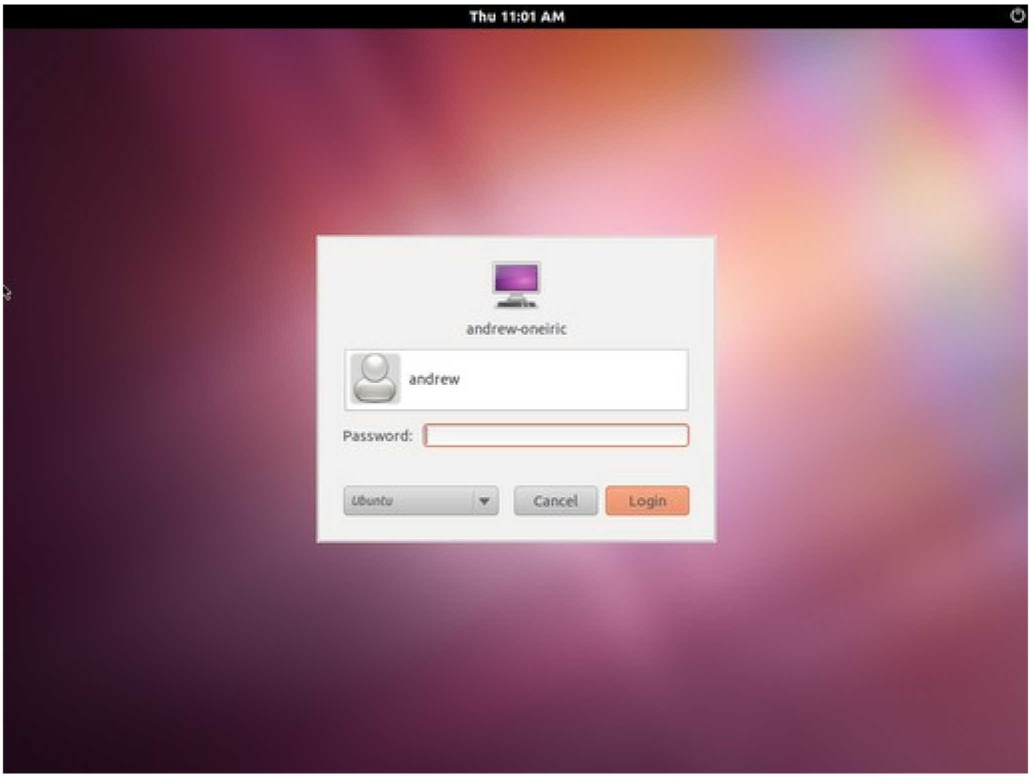
- lightdm - light display manager (default in ubuntu 11.10)



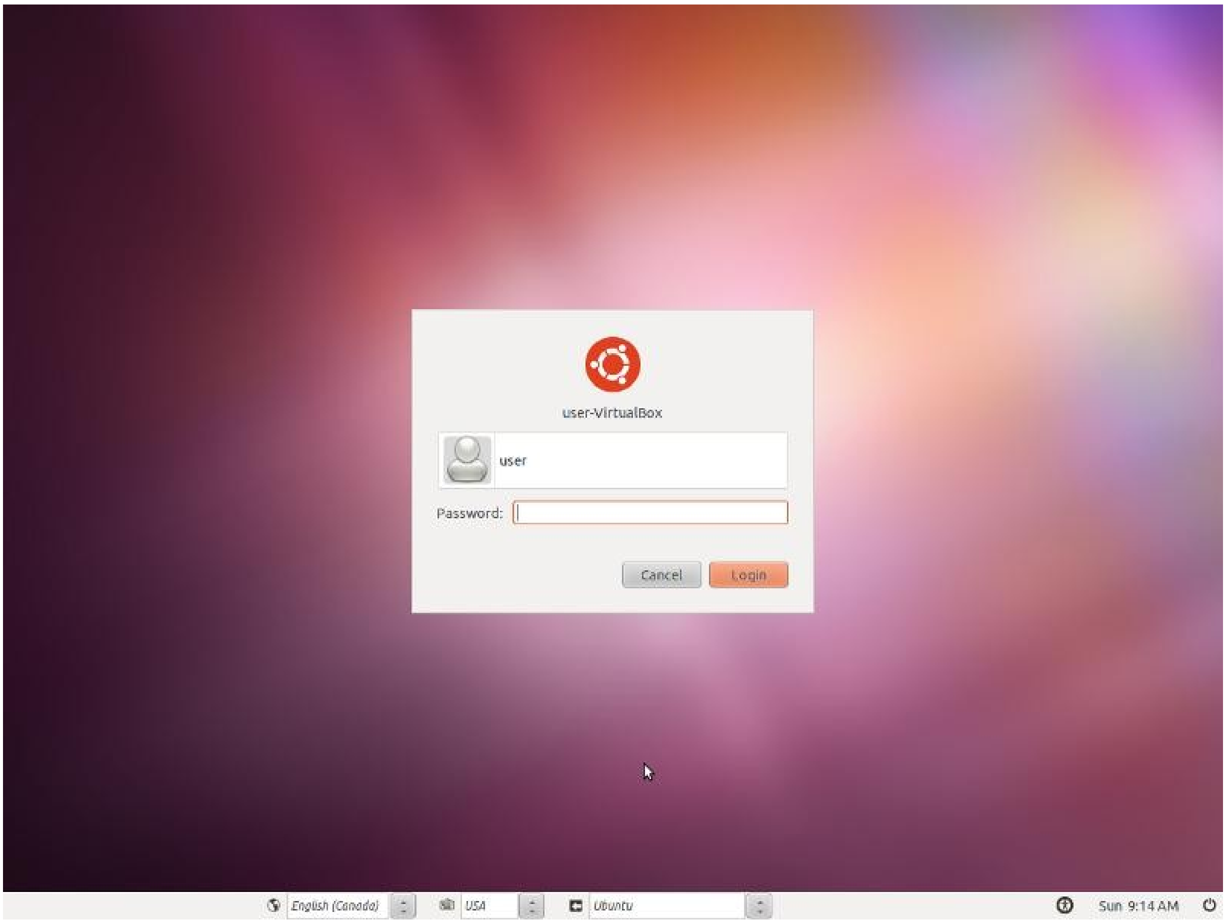
- KDM - KDE Display Manager



- GDM - Gnome Display Manager



- XDM - X window Display Manager (X11 default Display Manager)



Ubuntu's GUI structure

Unity

Compiz

Xorg

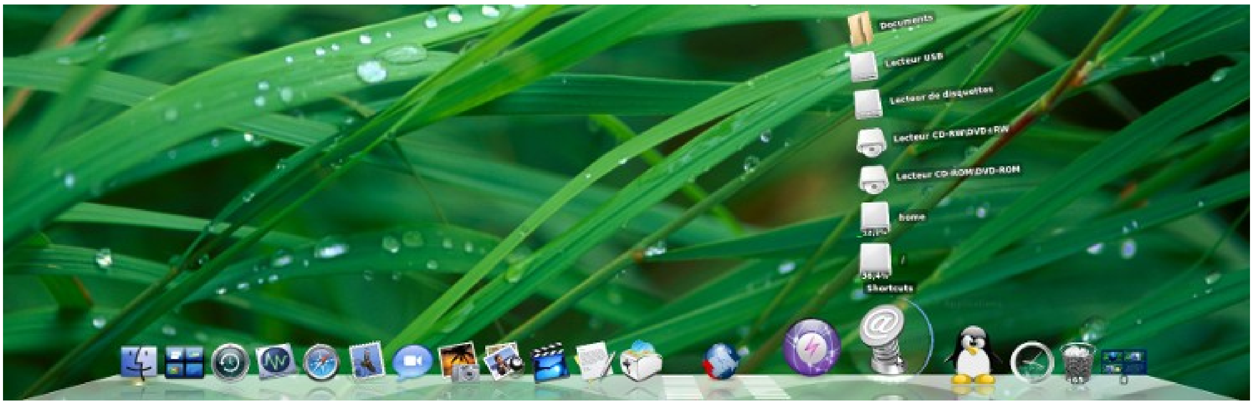
LightDM

Cool Stuff

There are a lot of extensions for each Desktop Environment. I'll show here the most common and useful extensions.

All the desktop environments

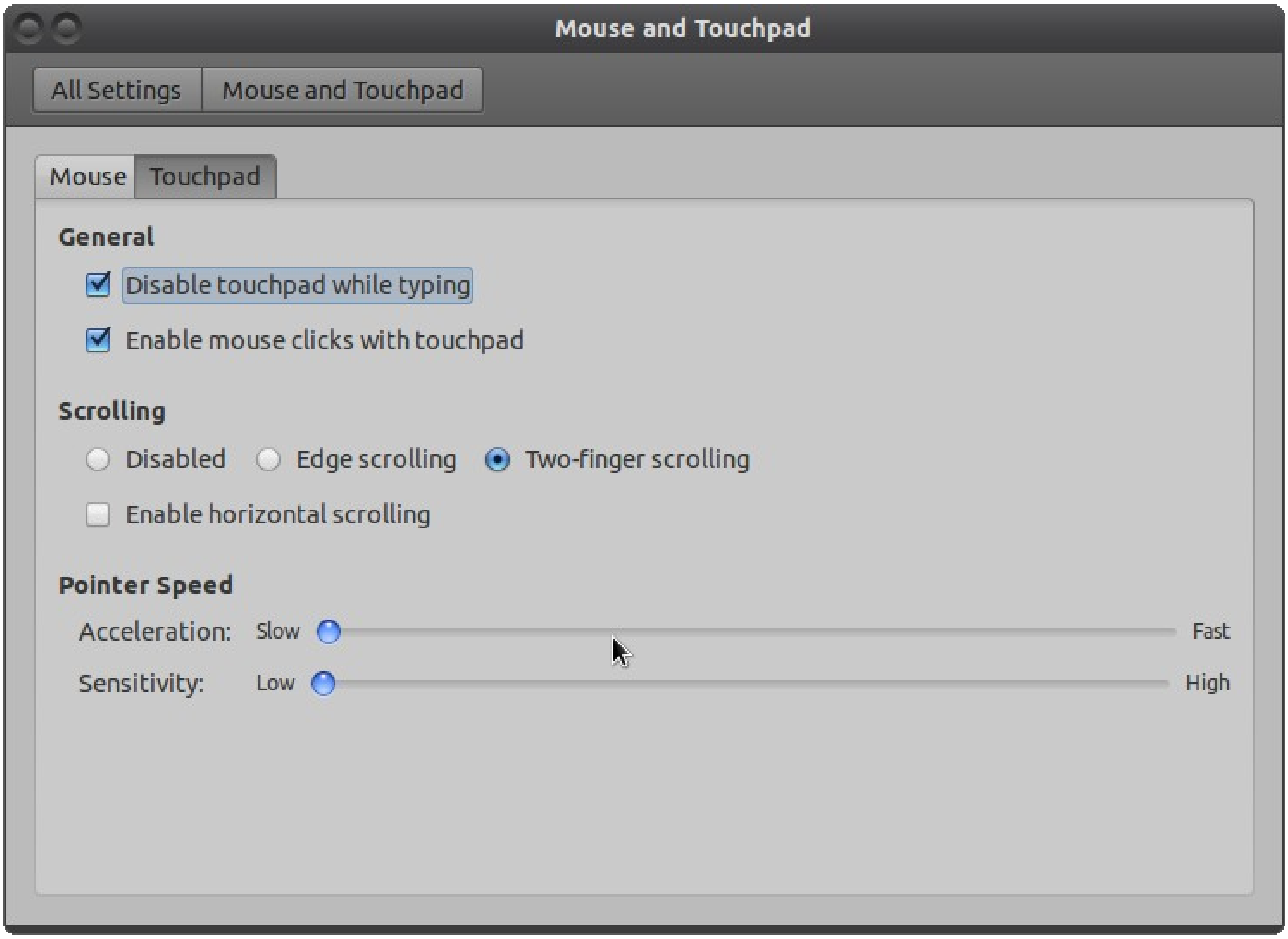
Cairo Dock – a nice dock bar which by default located in the bottom of screen.



Cairo Dock, another way to access applications

Note – when you install cairo dock, by default the workspaces launcher will be outside the dock. It disables the show desktop combination key (ctrl+alt+d). what you need to do is put this launcher inside the dock by Configure → advanced mode → Switcher... [completion needed]

Two Fingers touchpad scrolling:



Mouse and Touchpad

All Settings | Mouse and Touchpad

Mouse | Touchpad

General

☑ Disable touchpad while typing

☑ Enable mouse clicks with touchpad

Scrolling

○ Disabled ○ Edge scrolling ● Two-finger scrolling

□ Enable horizontal scrolling

Pointer Speed

Acceleration: Slow ○──────── Fast

Sensitivity: Low ○──────── High

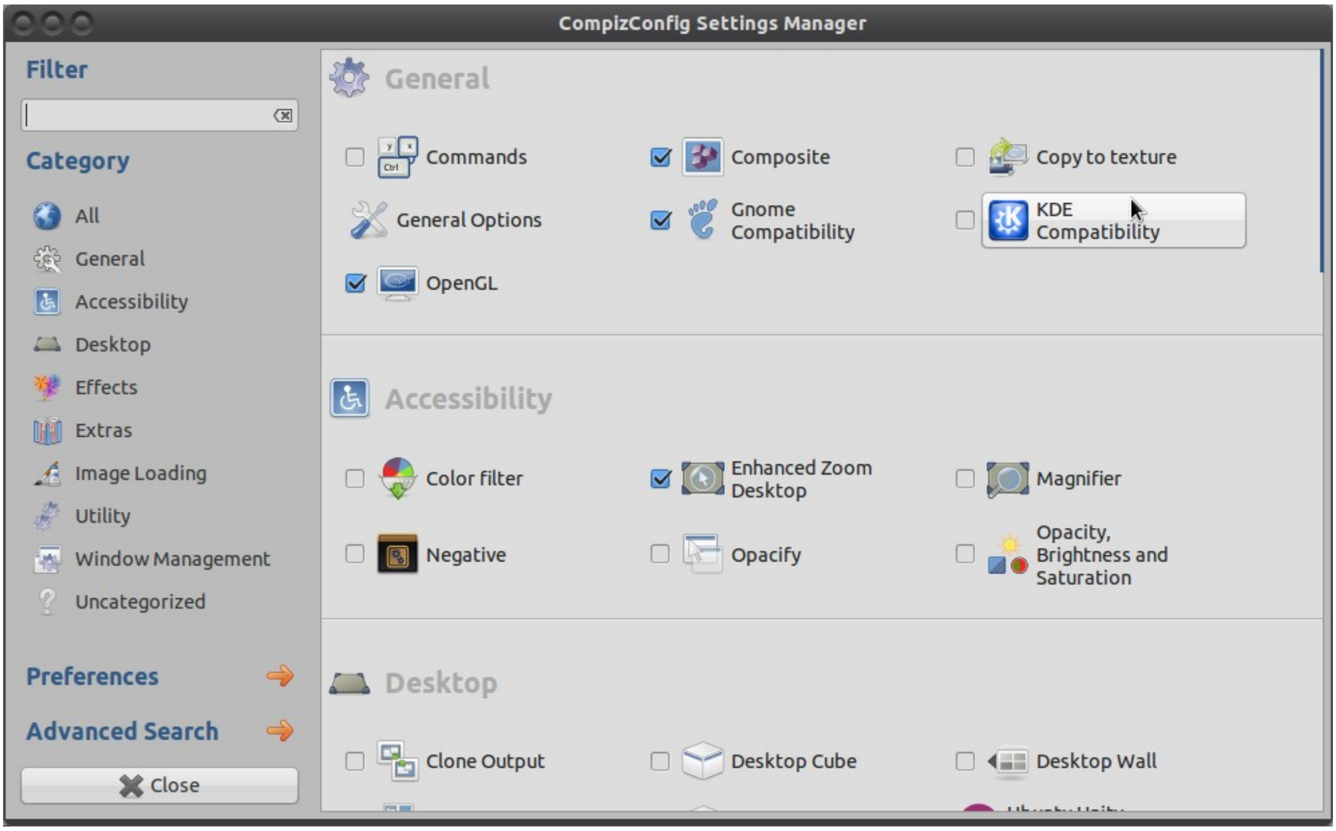
Mouse and Touchpad application

Unity

Compiz – 3D Window manager. also called composite window manager.

Installation

```
sudo apt-get install compiz compizconfig-settings-manager
```



General

□ Commands ☑ Composite □ Copy to texture
General Options ☑ Gnome Compatibility □ KDE Compatibility
☑ OpenGL

Accessibility

□ Color filter ☑ Enhanced Zoom Desktop □ Magnifier
□ Negative □ Opacify □ Opacity, Brightness and Saturation

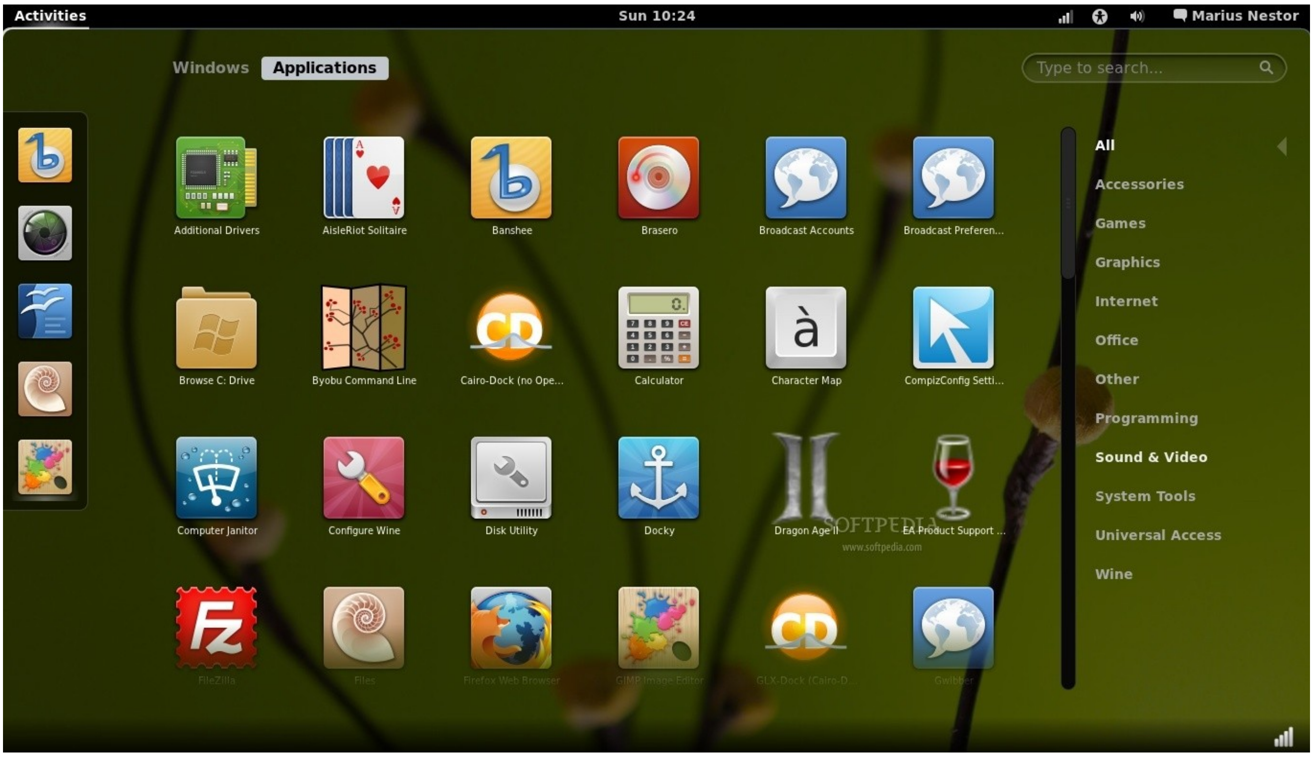
Desktop

□ Clone Output □ Desktop Cube □ Desktop Wall

CompizConfig Settings Manager – tool to tweak your desktop effects

Other Desktop Environments

Gnome



Gnome 3 Desktop Environment

Stands for “GNU Network Object Model Environment”.

Written in C.

Menu panel – top panel.

Application switcher – [completion needed]

Window Manager -

Installation

```
sudo apt-get install gnome-shell
```

Gnome 3 - Enable indicators at the top panel notification area – install from This link

after the installation, you will have a file in the following path:

```
~/.local/share/gnome-shell/extensions/EvilStatusIconForever@bone.twbbs.org.tw/extension.js
```

each indicator you want to move from the bottom panel to the top, just put its name at the notification array.

GTK+

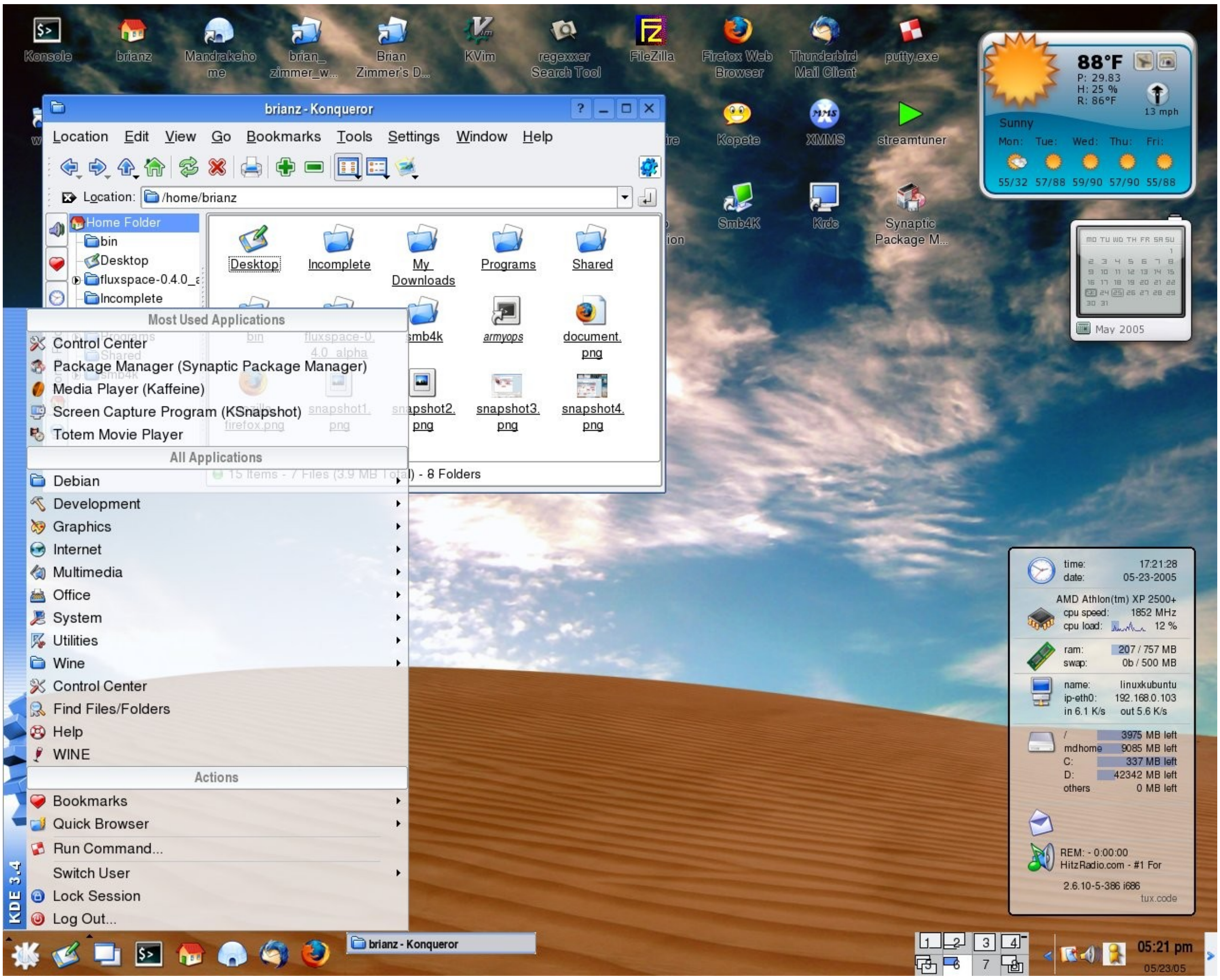
Toolkit / Library for developing graphical applications, usually resides on X Window System.

Some of the applications you are familier with was developed based on GTK+.

GTK+ is the default platform to develop graphical applications for GNOME, although all basic GTK+ based applications can run on other desktop environments.

One exception is – if the application written in GTK+ uses GNOME libraries, you have to add them to your system.

KDE



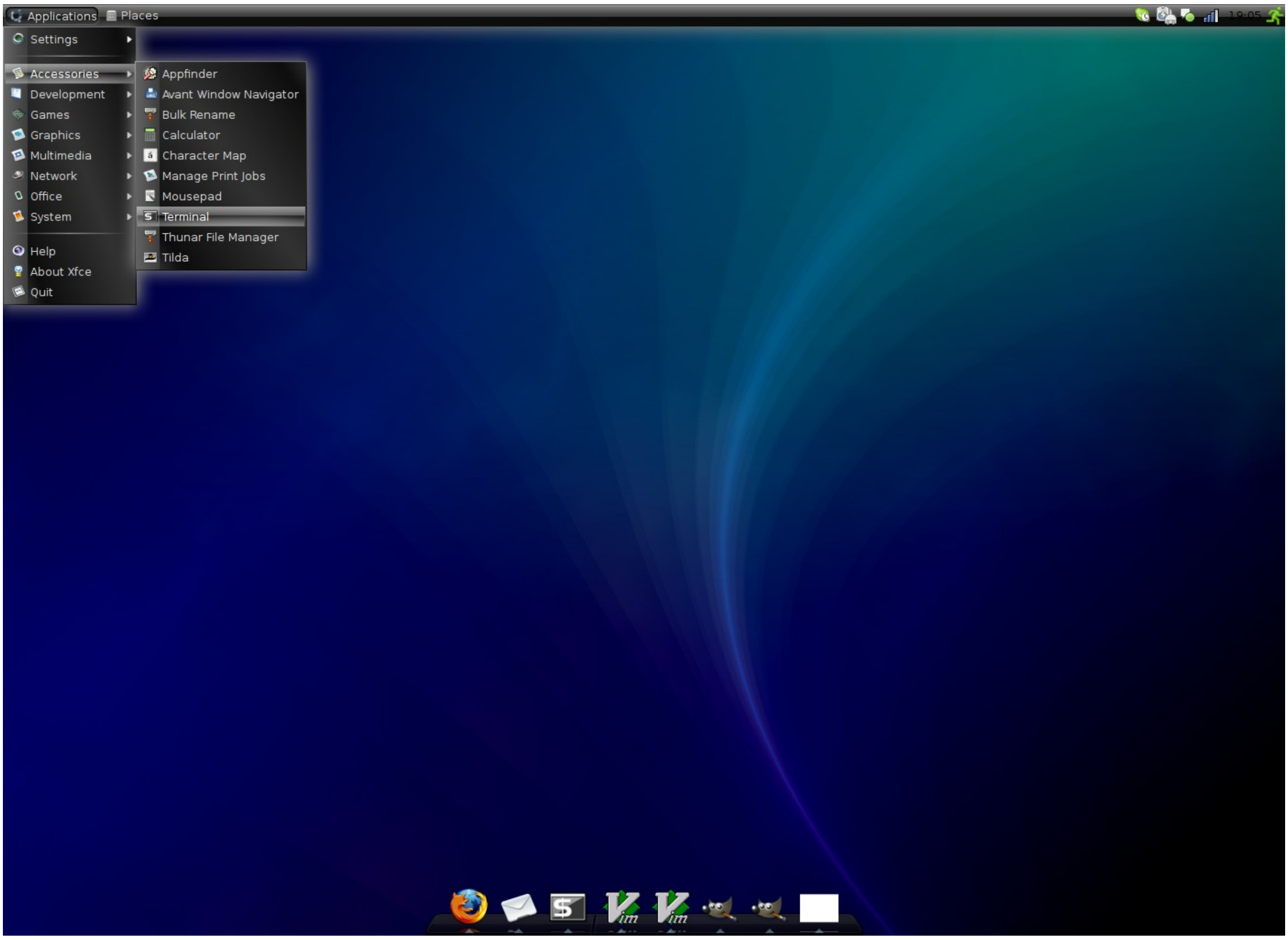
KDE Desktop Environment (Kubuntu is an Ubuntu version which has KDE instead Unity)

Written in C++

kde/qt

[completion needed]

XFCE



XFCE Desktop Environment (Xubuntu is an Ubuntu version which has XFCE instead Unity)

Stands for “”.

Written in ?.

Menu panel – top panel.

Application switcher – [completion needed]

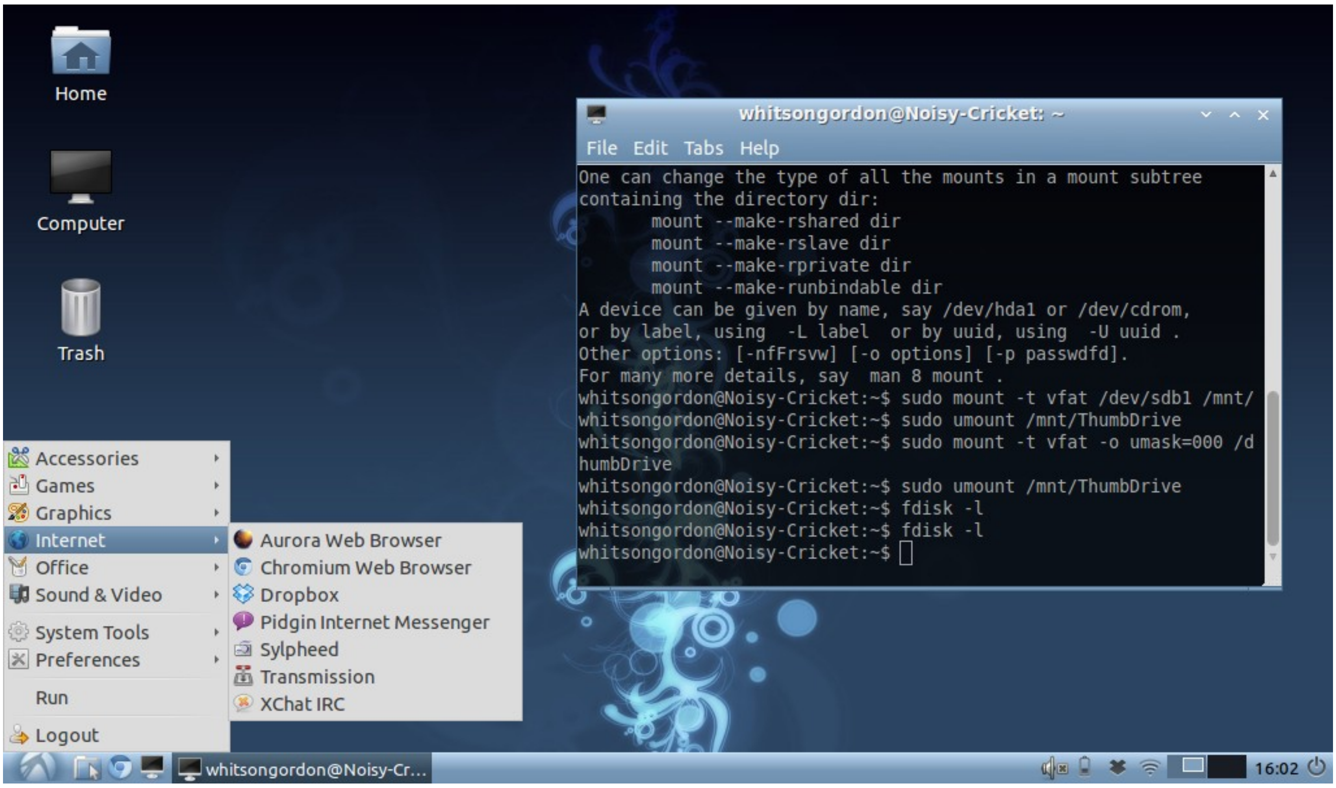
Window Manager - xfwm

Lightweight desktop environment

Installation

```
sudo apt-get install xfce4
```

LXDE



```
One can change the type of all the mounts in a mount subtree
containing the directory dir:
       mount --make-rshared dir
       mount --make-rslave dir
       mount --make-rprivate dir
       mount --make-runbindable dir
A device can be given by name, say /dev/hda1 or /dev/cdrom,
or by label, using  -L label  or by uuid, using  -U uuid .
Other options: [-nfFrsvw] [-o options] [-p passwdfd].
For many more details, say  man 8 mount .
whitsongordon@Noisy-Cricket:~$ sudo mount -t vfat /dev/sdb1 /mnt/
whitsongordon@Noisy-Cricket:~$ sudo umount /mnt/ThumbDrive
whitsongordon@Noisy-Cricket:~$ sudo mount -t vfat -o umask=000 /d
humbDrive
whitsongordon@Noisy-Cricket:~$ sudo umount /mnt/ThumbDrive
whitsongordon@Noisy-Cricket:~$ fdisk -l
whitsongordon@Noisy-Cricket:~$ fdisk -l
whitsongordon@Noisy-Cricket:~$ 
```

LXDE Desktop Environment (Lubuntu is an Ubuntu version which has LXDE instead Unity)

Stands for “”.

Written in ?.

Menu panel – ?.

Application switcher – [completion needed]

Window Manager -

The most Lightweight desktop environment

Installation

```
sudo apt-get install lxde
```

D-Bus

Desktop Bus that function as message bus system. An inter-process communication between applications which allows to exchange data and call each other functions.

It is a service.

It used by both KDE and GNOME Desktop Environments.

Comparison between desktop environments

http://www.renewablepcs.com/about-linux/kde-gnome-or-xfce

Adding more applications to Aptitude

What is a Package?

A Package is a set of files which consists an application. The package may include:

1. Documentation files
2. Configuration files
3. Executable files

What is Dpkg?

Dpkg stands for Debian PacKaGe. It's an Package Manager.

When you get a package, you need an application that knows how to install this package. This is the Package Manager.

What is Aptitude?

Aptitude is a Package Tool which:

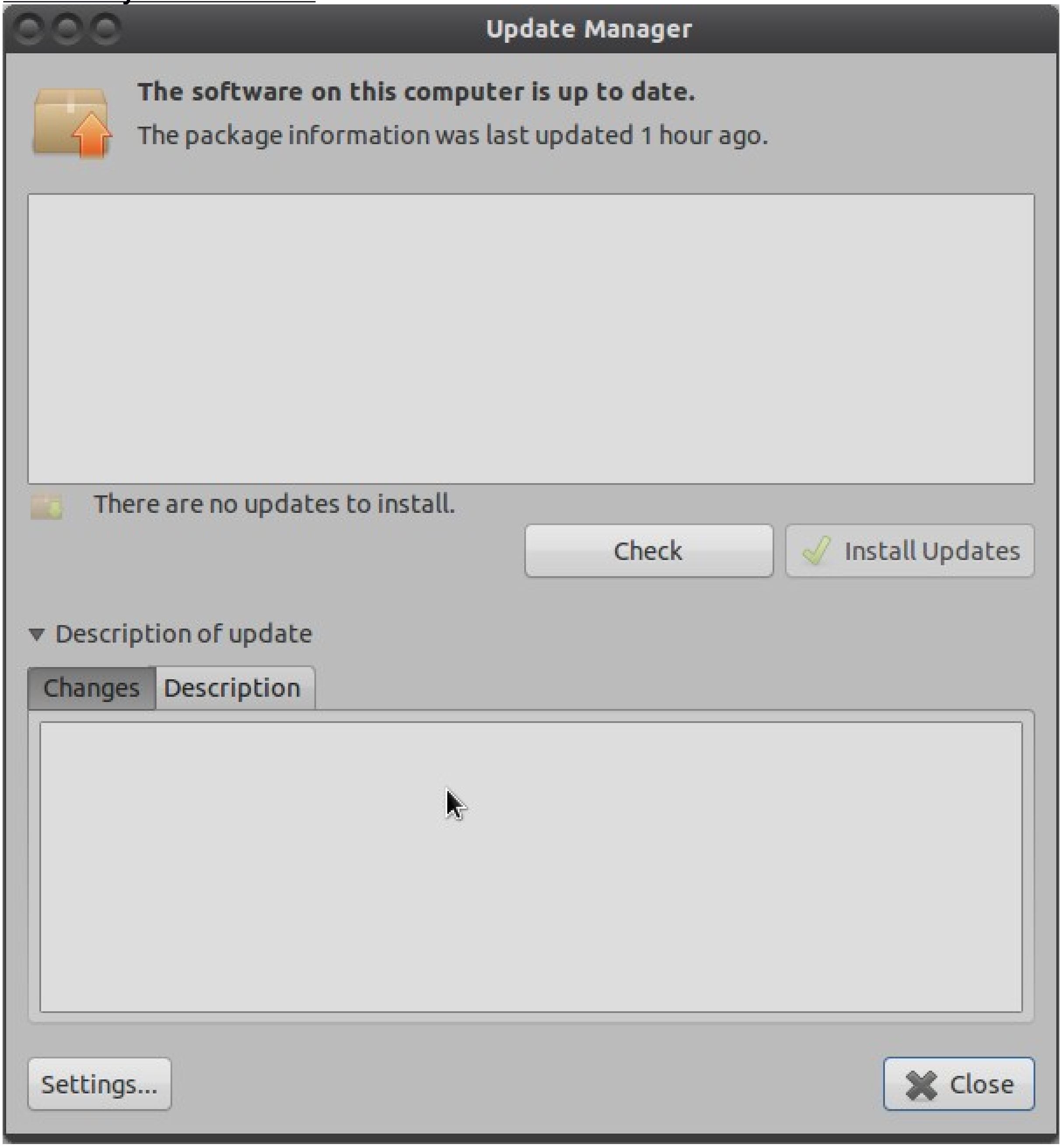
1. Download packages
2. Install them
3. Checking for dependencies between packages
4. Manage them (Such as searching for updates)

Aptitude Repository

In order to find which packages are available to you, Aptitude needs a repository which will tell him where to search for packages.

The repository contains addresses in which aptitude will search for packages.

Easiest way to add sources:

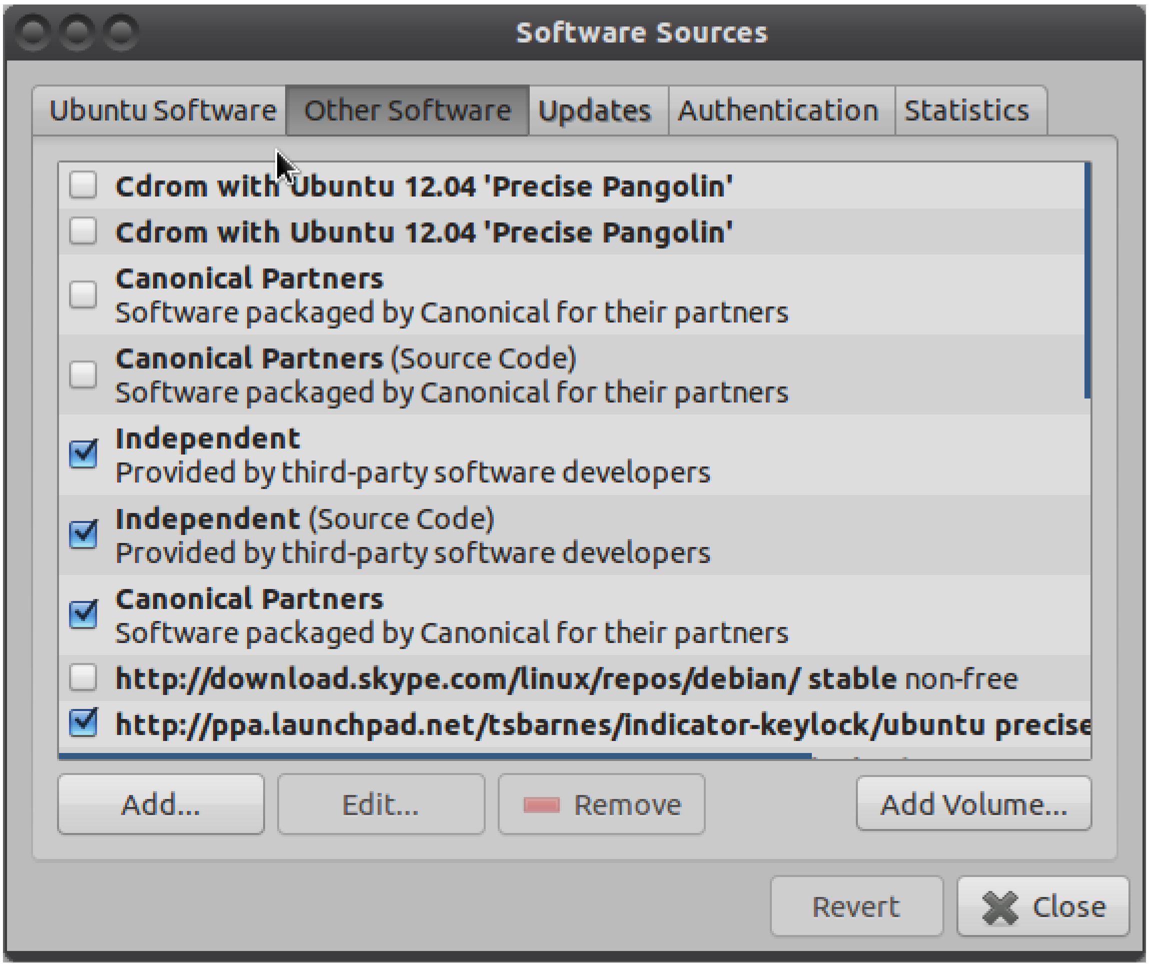


Update Manager

The software on this computer is up to date.

The package information was last updated 1 hour ago.

There are no updates to install.



☐ **Cdrom with Ubuntu 12.04 'Precise Pangolin'**

☐ **Cdrom with Ubuntu 12.04 'Precise Pangolin'**

☐ **Canonical Partners**
Software packaged by Canonical for their partners

☐ **Canonical Partners** (Source Code)
Software packaged by Canonical for their partners

☑ **Independent**
Provided by third-party software developers

☑ **Independent** (Source Code)
Provided by third-party software developers

☑ **Canonical Partners**
Software packaged by Canonical for their partners

☐ **http://download.skype.com/linux/repos/debian/ stable** non-free

☑ **http://ppa.launchpad.net/tsbarnes/indicator-keylock/ubuntu precise**

PPA – Personal Package Archive. An easy way to add sources to your repository only by typing source name and application. Each ppa will add a new file to sources.list.d directory.
In order to find PPAs, go to https://launchpad.net/ubuntu/+ppas?name_filter=
when you find PPA name you want to add, just click on Add button at the software software shown above and that's it!
In addition, sometimes you will search for a new application and you'll find PPAs at some web sites, it is useful.

Aptitude Repository Internals

There are two main files which assemble the repository:

1. **"/etc/apt/sources.list"** - this file contains list of package sources from which Aptitude tool gets the name of the packages. When you're running apt-get install, aptitude uses its local repository
2. **"/etc/apt/sources.list.d/"** -

if your repository contains address from which it can take a specific package you want, all good in the hood. But what if not?
All you have to do is to add a new address to the files above.

Structure of sources.list

```
deb http://chaos/debian    unstable    main local unofficial
```

```
    ------ base -------  distribution  ---- components -----
base – the address from which the packages will be downloaded
distribution – which state of the package do you want to download?
Uncompleted packages? Final packages? Common options:
```

1. stable
2. unstable
3. name of the distribution version
4. testing
5. experimental

component

1. main – officially supported
2. universe - Community-maintained, i.e. not officially supported software.
3. Multiverse - Software that is "not free"
4. restricted - Supported software that is not available under a completely free license.
5. non-free

Ubuntu Components:

1. security - Important Security Updates.
2. Updates - Recommended Updates
3. Proposed – Pre-Released Updates
4. Backports - Unsupported Updates.

This list includes sources from both sources.list and sources.list.d/

For example – [completion needed]

To check from which source aptitude gets a package:
apt-cache policy [package name]
or
apt-cache showpkg [package name]

For advanced using: http://www.webupd8.org/2012/02/how-to-use-launchpad-ppa-add-remove.html

Chapter 7 - Command line theories

What is a Command Line?

Command line is a way to tell your computer what to do by commands, rather than graphical objects.

The command line is non graphical environment, it's a textual environment. All you will see is characters.

For example, if you want to print some file content on the screen, all you need to do is write “cat” following by the name of the file.

Cat resolv.conf

CLI Facts

- GNU/Linux is **Case Sensitive!** It means that running the command PS will not work but running ps will work. Same as for filenames.
- Files and directories started with . Will be hidden. Your home directory contains several hidden directories. Check it! (ls -a)
- length of files are up to 256 characters and do not have to end with a suffix such as exe or doc, although the GUI Refers to it and will run the default application for this suffix

The Virtual File System (VFS)

Introduction

The virtual file system is the way which your files are ordered.

Instead of working with drive letters, all files are descendants of root folder (“/”).

there are directories which comes by default with Ubuntu operating system (and other GNU/Linux based operating systems), described here:

Directory	Details	Can reside on an exclusive partition?
Root "/"	files for boot and for single user mode	Yes
Bin	normal user commands (ls, chmod, cp, etc.)	No, part of “/”
sbin	advanced/administrator commands (parted, swapon, etc.)	No, part of “/”
Lib	Libraries. Shared libraries for reuse of several applications.	No, part of “/”
Boot	boot files, usually GRUB or traditional LILO and the kernel itself. Has to be under a primary partition (as part of “/” or by itself)	Yes
Etc	System configuration files (system config files? documents	No, part of “/”

Directory	Details	Can reside on an exclusive partition?
	and settings application data?)	
Usr	UNIX System Resources, Unchanging files like manuals, application binary files. the "program files" of linux. • /usr/bin - applications program files • /usr/lib - common libraries. the "dll/gac directory" of linux. • /usr/sbin – administrative programs such as display manager. Those programs are not part of the basic kernel/OS. Distributions will put their applications here.	Yes
Var	variable files: temporary files, spool (printer), mail (traditional, not email!), logs. In cotrast to /tmp this directory will usually hold the files for longer time.	Yes
Home	user files and profile. the "document and settings / users" of linux.	Yes
Dev	device files. those are character and block devices usually.	No, it's a special directory that must be available at boot.
Mnt	mountpoint for temporary mounts. Use it for manual mounts.	
Media	mountpoint for removable devices. Ubuntu mounts removable devices automatically there, devices such as CD, USB stick, etc.	
Proc	processes and kernel data	No, it's a virtual directory.
Tmp	Temporary data.	Yes
"/root"	root home directory	No

There are other directories but they are usually not in use so I haven't mentioned them.

Notes:

most programs resides on:

/bin

/sbin

/usr/bin

/usr/sbin

most of the applications you will install from apt repository manager, will put the application in those directories.

Mountpoints

Mountpoints are directories which points to another device or partition.

For example, I can create the directory /mnt/example and **mount** a partition to it. After this process, /mnt/example will point to this partition. In windows operating system you will use drive letter to access a partition. Here you will use a path to access a partition.

Note:

for single user mode you must have at least bin, sbin and lib. All of them must reside at the same filesystem/partition of root (/) partition.

Usr is not needed for minimal boot. You can run your system with multi-user mode without it.

/usr/local – intended to be a directory for applications which are not part of the distribution. In ubuntu you're barely find any files in here. Most of files will be putted under /usr directories. This directory will not be overwritten on upgrade.

Recommendations

- Create an exclusive partition for home directory
- Create a swap partition. It has better performance than a swap file.
- Make sure your partition on which the usr directory resides will be large enough, I say at least 5GB just for it's data. Recommended: 20GB.

Shell

User Interface. A shell can be either Graphical (GUI) or Textual (CLI).

The shell get commands from the user and communicate with the kernel so the command will be done.

The most common text Shells are:

- Bash – Bourne-again shell (GNU). the most common used. Ubuntu uses it by default.
- Sh – Bourne shell (UNIX)
- csh – C shell (BSD)

each text shell has its **metacharacters** - characters which have special meaning. Wildcards characters for example are metacharacters. (will be discussed later)

- File Types:
  - regular file
  - directory – directory is a file which contains a list of files.
  - Block files? - [completion needed]
    you can use CLI from X server environment by run the terminal emulator. In ubuntu it can be done by clicking CTRL-ALT-T.

Redirection / Descriptors

- Each command gets from the shell its standard input, standard output and standard error descriptors.
  - Stdin (default: keyboard)
  - stdout (default: screen)
- stderr (default: screen)

each descriptor has a unique number:
stdin – 0 “<”
stdout – 1 “>”
stderr – 2 “2>”

- to redirect the standard input use “<” so the input to the program will be other than the default
- to redirect the standard output use “>” so the output of the program will be written to another target such as files. “>>” will append the output to the target instead of overwrite it.
- To redirect the standard error use “2>”

Pipes

The standard output of command will be used as the standard input of the following command.

We put a “pipe” between the commands.

- Use the pipe (|) character to pipe between commands.
  Ls -la | less

filters

Commands which gets an input from the standard input, process them and send the result to the standard output. They use as joints between commands.

Most of time they use to “filter” data before you send it to the standard output.

Other

Chapter 8 - Using Command line usefully (GNU/Linux commands)

Introduction

This chapter will focus on the most useful commands at the GNU/Linux Command Line Interface. It means that they will work on any GNU/Linux distribution.

Note, There are specific sections which will focus on Bash commands.

- Each command gets options and arguments
  - cp -R /home/user1 /home/user2

GNU Basic Commands

Categories: directory handling, text handling, permissions handling, users,

Command	Details	Parameters
Ls	List Directory. • colors: • green executable file. • Blue – directory. • Turquoise – soft link • green background – shareable directory, other users can write to it. • Red background – [completion needed]	• -l • -a • -R
Cd	Change Directory	"cd -" - will move to the previous directory (not one directory up!)
pwd	Print Working Directory. Tells you the absolute path you are in	
mkdir – Make Directory.		Mkdir -p – will create directory hierarchy, a several directorys (mkdir ./hello/world)
Find	Search for files and directories. Default behaviour: search recursively for files and directories.	Find *{from which directory}* -name *{expression to search}*
Alias	Set alias for a full line command	Alias *{name of the new*

Command	Details	Parameters
		alias}='{the command this alias will run}'
Unalias	Will remove an alias	Unalias *{name of the alias}*

Text handling

Command	Details	Parameters
paste / join	perform join between two files by joining corresponding lines.	paste firstnames.txt lastnames.txt
Tr	translate characters to other characters. useful when you want to change text to lower or upper case	
Pr	prepare text for printing and can put header at each page	
fmt	formats your output so the text will be wrapped if it exceeds console width.	
cut	filter output text by choosing specific fields/columns only. You can choose numerous fields. • -f : column number • -d : which delimiter separate between the columns (usually space or tab) • -c : show only characters by the position.	
Sort	sort incoming text. • -t : fields delimiter • +# : by which fields to sort.	
Head	shows only the first lines of a text. Default number of lines: 10. • -# : change number of lines where # is number of lines. • +# - line from which to start printing.	
Tail	shows only the last lines of a text. • -# : change number of lines where # is number of lines. • +# - line from which to start printing.	

Command	Details	Parameters
	• -f : will keep printing new lines that is updated by another process. For example, if you have a log file and you want to keep reading it and see what is new, it is useful.	

General

- Clear – clears the screen
- echo – print a text to stdout
- Tee – gets a text from the standard input, write it to a file and send it to the standard output.

CLI comfort handling

- Alias – allows to create alias to a command so it will be called differently, for example to run ls with -l always (alias ls='ls -l').
  - unalias will remove the alias (unalias ls).
- history – history of commands which has issued in the past.
  - # - number of last commands to show (history 10)

Searching

- egrep – like grep but more sophisticated. Allows to search regular expression's advanced patterns
- grep (without using advanced regular expression)
- useful parameters:
  - -i case sensitive
  - -v non match lines (not)
- grep – Global Regular Expression Print. Search and filter specific wildcard expression or Regular Expression (Will be discussed later).
  - -v : only lines which does not match the expression.

Search & replace command line

- sed – command to search and replace text. Can get the text from the standard input or file as a parameter. The search expression uses regular expression.
  - sed 's/original text/replaced text' – will replace the first occurance.
  - Sed 's/original text/replaced text/g' – will replace all occurances.
  - Sed can get regular expressions.
  - Rule of thumb – sed gets three slashes
    - before the first one - commands
    - after the first one – search for the expression
    - after the second – to what to replace
    - after the third – flags.
  - Sed '/searched text/a text to add after this line' – to add a text in the middle of a file, **after** specific line.
  - Sed '/searched text/i text to add after this line' – to add a text in the middle of a file, **before** specific line.
  - Sed '2i my text' – to add a text **in** the second line.
  - Sed '2a my text' – to add a text **after** the second line.

  - -i – will update the input file.

File and directories handling

- locate – search for files just like *find* but faster. Uses slocate DB which lay at /var/lib/slocate. Updated by a nightly cron job(?)
- Cp – copy file(s) from source to target.
  - cp -R – Copy recursively all the subdirectories inside the source directory.
- Mv – moves a file or directory from one place to another.
- mkdir - create new directory
  - -p - creates directorys hirarchy
- rmdir – Remove Directory. The directory must be empty.
- File *{file with full path}* –tells what type of file it is and which architecture it has compiled.

User and Permissions Handling

- passwd – change your password
- sudo – run a command with superuser permissions
- who – tells who are connected and from which virtual terminal.
- chmod - Changes permission at file or folder. can run recursively.
- chown - Changes the owner of file or folder. useful when you have no permissions to change permission.

Program Return Codes

Return codes – each command return code at the end of the execution.

Return Code	Meaning
0	Success

Bash Commands

- To determine which code has returned run “echo $?” (Bash)
- To run two commands sequentially use the semicolon (;) character. Note that the second command will be executed always. (apt-get update ; apt-get upgrade)
- To run two commands sequentially and run the second only if the first has succeeded, use the double ampersand (&&) characters. (lkfjkljfdsfdkj && apt-get update)
- to run a command only if a previous command has failed, use the double vertical (||) characters (echo || echo “hello”)
- To run command from string, run sh -c *{command string}*

Text viewers and editors

Editors

- Vi – the old UNIX editor. Really complex! It's good when you have to advance editing, but for most users I think it's needless.
- nano – more comfortable editor for me. Not all the distributions install it, but as long as you use ubuntu, you have no worries :)
  - ctrl+\ - search & replace

  - ctrl+x – exit the editor
  - ctrl+o – save
  - ctrl+w – search
  - ctrl+k – cut line (also used to delete a whole line)
  - ctrl+u – paste
  - ctrl+^ - mark text (used to mark text and then cutting it)
- nl – numbering lines. Good when you want to print to the screen a code file or if you have a very long text file and you want to know in which line you can find a specific data.

Viewers

- ➢ more
- ➢ less
- ➢ Cat – Concatenate. Prints out a text to stdout. If the text is long, it won't stop printing can get several files to print or single file.
- ➢ tac

important shortcut keys in most of viewers and editors:

- ➢ / - finds an expression/pattern by regular expression
- ➢ n - next match
- ➢ N - previous match
- ➢ q – quit
- ➢ h - help

Ubuntu Specific Tools

- apt-get
  - autoremove - will remove all the packages that are no longer needed (packages that were installed as a dependency of other package)
  - install - install packages
  - remove - uninstall packages but retain configuration files
  - purge - uninstall packages and include configuration files

Deep Diving

Wanna learn more about each command? RTFM! (remember?)

each command has it's own manual page which can be accessed by the command **man [***command name]*. For example, *man ls* will explain how to use ls and what are the options for this command.

Man uses *less* application. Reuse is good! So If you want to use *man* usefully, learn how to use *less.*

Man is divided to the following sections:

- NAME – general description
- SYNOPSIS – how the syntex works
- DESCRIPTION – deep description about the command
- OPTIONS – which options are available. Sometimes the options section is ommited and the options are included in the description section
- NOTES – special notes you should know.

- FILES – file which used by the command, for example configuration files.
- AUTHOR – who written the manual/command.
- REPORTING BUGS – where you can report about bugs.
- COPYRIGHT – the company whose command/application belongs to.
- SEE ALSO – other manual pages related to this command.

If you don't sure which page is relevant to you, run “man -k *your keyword*” and man will print out which pages can be relevant to you. Try it with *man -k unity!*

Man has several sections, so one keyword can refers to several things:

1. User Commands
2. System Calls
3. Libc calls
4. Devices
5. File formats and protocols
6. Games
7. Conventions, macro packages and so forth
8. System administration
9. Kernel

man {*section number*} {*keyword*}

you can always try running *{keyword}* –help.

CLI Shortcut Keys

Shortcut Key Combination	Result
CTRL+C	Terminate the current commnad
CTRL+W	Erase the last word
CTRL+U	Clear the text in line starting from the cursor to the beginning of the line
CTRL+Left/Right Arrow	Moves between words at the current command
Tab	filename completion. Also completes at some commands the options (double tab). For example, apt-get.
Up/Down Arrow	move between the commands which has issued. (commands which perform at history)
Ctrl+D	Ends the current VT session

Chapter 8 - CLI Power User

Introduction

Working with Text mode only

To work with Textual VT only, you can shut down the GUI VT by the command “service lightdm stop” from one of the Textual VT. Before you do it make sure ALL OF YOUR APPLICATIONS ARE CLOSED.

Wildcards

Wildcards are characters which helps you find something. Most of times you will use it to find files.

- * - zero or more characters.
  - Ls /usr/bin/gnome* - will show all the files starting with gnome.
- ? - one character exactly.
  - Ls *.??? - will find all the files that has three letters extension at the current directory.

Regular Expressions

Regular Expressions are like wildcards but just more sophisticated. You can do more comprehensive searches with it.

background

putting the expression quotes is recommended so special signs will be recognize.

Meta-character	Detail
^ (caret)	the expression exists at the beginning of the line (^[exp])
$	the expression exists at the end of the line ([exp]$)
*	the preceding character exists zero or more times
+	the preceding character exists one or more times
.	any character, fill one character place. (like ? In wildcard) this does not include \n and \r
+	the preceding character exists one or more times.
?	the preceding character is optional (exists zero or one time)
|	match at least one of the expressions in each line. the expressions are separated by the pipe ('he|she|it')
[]	any of the characters within the brackets is valid ('[Hh]ello'). Refers to the current position.

	For example – gr[ea]y will accept *gray* or *grey*. Can also take range of characters such as [a-z] or [0-9a-z]
[^]	All the characters **except** those which are in the brackets. For example, “^[^0-9]+” will return the strings which don't start with digits.
Escape character	\
Special Characters	[\ ^ $. | ? * + ( ) All of there can be produce by putting the escape character “\”
{n}	The preceding character will be perfromed n times
{n,m}	The preceding character will be perfromed between n times to m times

Regular expression which work on specific tools:

Meta-character	Detail
\d	Represents digit
\D	Any character **except** digit
\w	Represents letters (a-z A-Z for example)
\W	Any character **except** letters
\s	Represents space
\t	Represents tab

Advanced Commands:
sed - edit file with regular expressions. for example, to add character at the beginning of each line, or to perform find & replace.

Job Control

Jobs are managed processes which can run or hold at the background, so you won't notice they exists and in the meantime you will be able to do other things in the current shell.
Process/Program runs by default in the foreground.

- Each process can be run as a job.
- Each process gets a job number in addition to its PID.
- Process/Program runs by default in the foreground, so it's the only thing you can do at the current shell.
- Foreground -
- Background -
- jobs refers by %{*job number*} with the correlated commands.
- to run a process in the background (as a job), add ampersand (&) after the command (skype &).
  - this method can be useful when you run gnome / KDE applications from the terminal window.
- Ctrl+Z suspending the process and putting it in background.
- the "jobs" command shows all the jobs and their states (running, stopped, etc.)

- + represents "current job".
- - represents "previous job".
- fg - command which run a process in case it's in suspended mode and move it to the foreground.
- bg - command which run a process in case it's in suspended mode and move it in the background.
- both of them get the job number as a parameter. you can discover the job's number by "jobs" command. (fg %1)
- kill command can kill jobs in addition to processes.
  - Kill %*{job number}*. For example “kill %1” will kill job number 1.

Processes:

- represents with pid (Process ID)
- ppid - Parent Process ID

* all of the processes are descendants of init process (pid = 1).
* ps - shows processes which run in time. defaultly, shows processes that run in the current terminal and owned by the current user.
* -e all processes
* -F extended fields

- --sort sorts by given fields (--sort=time,pid etc.)
- --pid shows only the processes which has listed (--pid 1,5,200)
- --ppid shows only the processes which are direct childs of given process id (--ppid 1,2)

* pstree - shows processes in tree view hierarchically
* -c doesn't compact similar subprocesses.

* single quotes - disable shell interpretation for special characters such as asterix. for example rm 'h*2' (h*2 is a file).
* double quotes - like single quotes (but not for all special characters) but useful when you want to use single quote in the expression "Don't go".
* interprets - $,
* backslash "\" - disable special character. echo Tom\'s Diner
*

special characters
* $ - variable

Basic Permissions

General

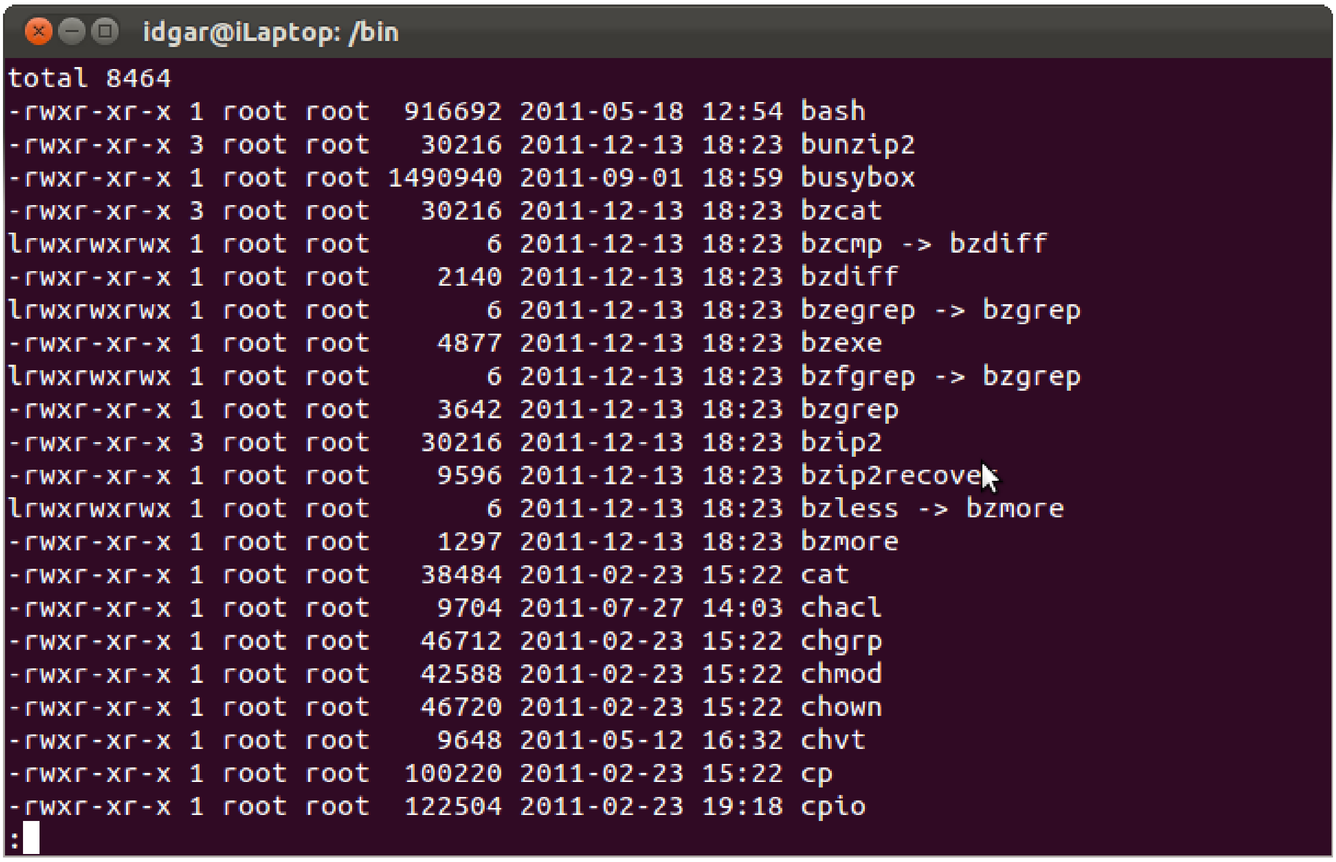
The permissions in linux consists of several components:

- Groups
- Users
- Type of access: Read, Write, eXecute

Each file mange it's permissions in the following way

- there is one Owner to the file (usually the creator of the file)
- the file is assigned to one group
- all other users can get a specific permissions.

The Permissions can only be changed by the Owner or Root user.



```
total 8464
-rwxr-xr-x 1 root root  916692 2011-05-18 12:54 bash
-rwxr-xr-x 3 root root   30216 2011-12-13 18:23 bunzip2
-rwxr-xr-x 1 root root 1490940 2011-09-01 18:59 busybox
-rwxr-xr-x 3 root root   30216 2011-12-13 18:23 bzcat
lrwxrwxrwx 1 root root       6 2011-12-13 18:23 bzcmp -> bzdiff
-rwxr-xr-x 1 root root    2140 2011-12-13 18:23 bzdiff
lrwxrwxrwx 1 root root       6 2011-12-13 18:23 bzegrep -> bzgrep
-rwxr-xr-x 1 root root    4877 2011-12-13 18:23 bzexe
lrwxrwxrwx 1 root root       6 2011-12-13 18:23 bzfgrep -> bzgrep
-rwxr-xr-x 1 root root    3642 2011-12-13 18:23 bzgrep
-rwxr-xr-x 3 root root   30216 2011-12-13 18:23 bzip2
-rwxr-xr-x 1 root root    9596 2011-12-13 18:23 bzip2recover
lrwxrwxrwx 1 root root       6 2011-12-13 18:23 bzless -> bzmore
-rwxr-xr-x 1 root root    1297 2011-12-13 18:23 bzmore
-rwxr-xr-x 1 root root   38484 2011-02-23 15:22 cat
-rwxr-xr-x 1 root root    9704 2011-07-27 14:03 chacl
-rwxr-xr-x 1 root root   46712 2011-02-23 15:22 chgrp
-rwxr-xr-x 1 root root   42588 2011-02-23 15:22 chmod
-rwxr-xr-x 1 root root   46720 2011-02-23 15:22 chown
-rwxr-xr-x 1 root root    9648 2011-05-12 16:32 chvt
-rwxr-xr-x 1 root root  100220 2011-02-23 15:22 cp
-rwxr-xr-x 1 root root  122504 2011-02-23 19:18 cpio
:
```

Each character defines something:

File Type, Owner permissions, group permissions, other Permissions.

We can also see that the owner is root and also the group is root (this is /bin directory so it makes sense).

Pay attention that all of those files are eXeutables in order to execute them. It's different from windows, it doesn't determine if the file is executable or not by the extention (.exe, .com, etc.)

if a directory doesn't have executable permissions, you cannot enter into it.

- Chmod – changes permissions to file. Remember that you has to be root or owner of the file in order to do it.
    - Symbolilc - Chmod *to who(u,g,o), add (+) or remove (-) or reset (=), which permissions (r,w,x).* For example, chmod o+r will allows other users to read this file.
      You can determine different permissions to different user types at one command, chmod u=rwx,go+x
- useradd -
- usermod -
    - usermod -a -G *group_name1,group_name2.... username* -
- groups -

/etc/passwd – list of users in the host. Includes mainly user name, user id, primary group id, description.

/etc/group – list of groups in the host. Includes mainly group name, group id, explicit users which are in the group (it's enough to be connected to the group by putting primary group Id in /etc/passwd). Most of the groups will not have explicit users.

For further reading:

http://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/

http://www.cyberciti.biz/faq/understanding-etcgroup-file/

Manage user accounts

Users

Each account has its own UID, or User ID.

To see what is your id run the command “id”.

Groups

there are also group permissions in GNU/Linux environment. It means that you can link UID to GID.

To see to which group your user belongs to, run the command “id”.

Manage Advanced permissions

- Octal Permissions – you can determine file permissions by octal numbers. The permission consists of three digits: first to owner, second to group and third to other.
  - r = 4
  - w = 2
  - x = 1
  - 7 = 4 + 2 + 1 (permission to read, write and execute)
  - 6 = 4 + 2 + 0 (permission to read and write)
  - for example, to give all permissions only to the owner, chmod 700 will be used.
  - Rule of thumb – even digit will include execute permission, odd won't.
- Umask – remove permissions
  - umask 022 will remove write permissions to group and other.

Ubuntu command line commands

Apt-get upgrade -

will upgrade all of your packages if there are upgrades / new version of the package.

apt-get dist-upgrade –

same as upgrade + replace old packages with new packages. It means that the name of the package has changed.
Most of uses for that is when the kernel version has changed. Each kernel has its own package name (linux-image-3.2.0.29-generic vs. linux-image-3.5.0.18-generic).
Most of the packages retain the same name – for example google-chrome-stable.

Chapter 9 - Basic Administering Linux

- Root – superuser at GNU/Linux environment. Can do anything, so do not use you system as root unless you really need to!
- Top - shows the processes that are alive and other statistics about the system such as CPU usage, Memory usage, etc.
  - Takes from today's Cores around 1% cpu time.
  - z - show in colors. Helpful!
  - x - marks the field by which sorting is perform
  - m - shows memory summary
  - t - shows CPU summary (tasks)
  - l - shows Load Avarage summary
  - d - refresh (delay) time
  - f - fields to show
  - u - choose only specific user processes
  - 1 - shows Single CPU summary / SMP CPU summary

Processes

- A program that resides on memory.
- Each process has it's own Process ID, or PID.
- Each process has it's parent, except init which is the first process and that's why it's PID is 1.
- in order to end a process you have to send it an Interrupt. (will be discussed later)
  the Interrupt is sent to the process through the parent process. (Dear son, please leave or GET THE HELL OUT OF HERE!)

Signals

Signal number	What it does
01(SIGHUP)	End process (happens when the parent is terminated – sends Hang UP signal). Sent by the kernel?
02(SIGINT)	End process (ctrl+C). sent by the kernel.
03 (SIGQUIT)	End process & core dump
09 (SIGKILL)	Kill process by its parent– the process can't end itself properly.
15 (SIGTERM)	End process (default signal, for example when you run kill *pid*

Keep in mind that signals 01,02 and 15 does the same thing but is sent from different places. Most of times you will use Ctrl+C to end process.

- Nohup – the command will not stop until it finishes, even when its parent sends Hangup signal or Quit signal. Those processes will be disconnected from their parent and will be directly under init process (PID 1)
  Nohup cp /home/abc.txt /home/user/

The order you should use:

1. End Process from the process itself – choose quit program of such thing.
2. End Process – by kill command, kill *pid (*sends signal 15)
3. Kill Process – by kill command – kill -9 *pid*
4.

- killall – like *kill* but instead of using PID argument, it will use the process name. Keep in mind that ALL of the processes that has the same name will be end.
- Interrupts - commands from the kernel to the process (by priority)
- TERM - Terminate please (code number 15)
- KILL - Force killing the process without giving him the chance to terminate gracefully. (code number 9)
- Kill - most used to terminate or kill process.
- kill -9 [processname] will kill the process
- kill -15 [processname] will terminate the process
- ~/.config - folder which holds your user desktop configuration
- autostart - startup applications list
- some of the applications saves it's configuration there

How to start application minimized?

Some of the applications support the "-h" switch, which means "hide".

For example, Empathy supports it:

empathy -h

Deamons

Deamons are services which run in background and are long lived process.
They run all the time... or of course until you shut them off.

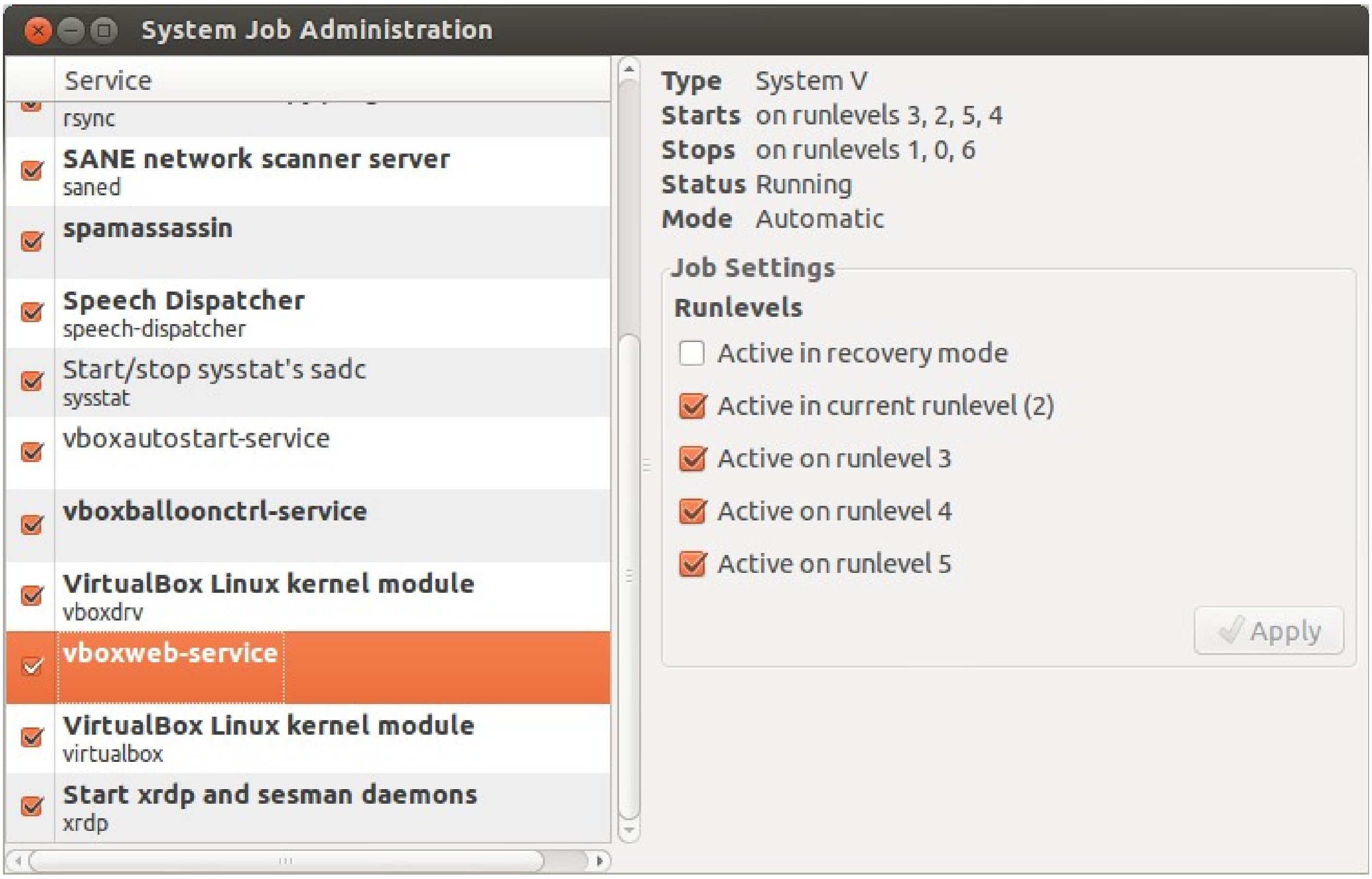
For example, the display manager is a deamon that take care of your login sessions. When you log off, it still runs.

- configuration files usually located in /etc

to control daemons, use the command *service* [daemon name] [start/stop/restart].

~/.config/autostart – all the applications which will run at startup. Pay attention! You have to add the applications first from GUI or else no one will call the files.

Another tool to manage daemons: jobs-admin
sudo apt-get install jobs-admin



Service

☑ rsync

☑ **SANE network scanner server**
saned

☑ **spamassassin**

☑ **Speech Dispatcher**
speech-dispatcher

☑ Start/stop sysstat's sadc
sysstat

☑ vboxautostart-service

☑ **vboxballoonctrl-service**

☑ **VirtualBox Linux kernel module**
vboxdrv

☑ **vboxweb-service**

☑ **VirtualBox Linux kernel module**
virtualbox

☑ **Start xrdp and sesman daemons**
xrdp

Type System V
Starts on runlevels 3, 2, 5, 4
Stops on runlevels 1, 0, 6
Status Running
Mode Automatic

Job Settings

Runlevels

☐ Active in recovery mode

☑ Active in current runlevel (2)

☑ Active on runlevel 3

☑ Active on runlevel 4

☑ Active on runlevel 5

to understand how they work, go to chapter 10 (sysVinit, upstart)

Chapter 10 - Advanced Administering Linux

General

[completion needed]

To read about change log for kernel versions:

http://kernelnewbies.org/Linux_3.6 – example for page to see changes about Linux 3.6 kernel.

http://kernelnewbies.org/LinuxChanges – example for page to see changes about the last Linux kernel.

Boot Process

Boot Loader

Boot loader is an application which load the kernel into the memory and enable running the whole Operating System, Choose which Operating System to load (in case of multiple boot) and so on.

Boot loader usually resides on MBR, Boot Sector and files in a filesystem..

MBR has 3 parts:

1. 446 bytes of boot loader code (for example GRUB)
2. 64 bytes of partition table.
3. 2 bytes refer as magic bytes (BIOS check those bytes to verify that this sector is really MBR)

GRUB

Note: i'm not going to describe all the details about GRUB but just the main ones. At the end of this section there is a very good link for a web page which explain more about it. You are more than welcome to read it and deep dive.

Intro
GRUB stands for Grand Unified Boot loader. It's actually GNU's boot loader and that's why most of distributions use it.
It has Graphical menu in which you can choose what do you like to boot – specific operating system or a program's image (like memtest86)
GRUB can boot other operating system such as Microsoft Windows. In fact, GRUB can boot almost any operating system!
Total size between 20KB-30KB.

Ability to decompress compressed kernels (vmlinuz for example).

The newest version of GRUB that comes with the newest ubuntu versions is GRUB 2.
Its structure is a bit different than previous versions.

Stages

- Stage 1 – MBR code. It's only 446 bytes so it's not enough to run GRUB. The following code resides at stage 1.5. **all Stage 1 does is to point to stage 1.5**.

- Stage 1.5 – Filesystem specific. Loading your "/" filesystem code (ext3/4 by default) so GRUB will be able to run Stage 2. Was in the past. Now it's kind of not exist because it resides right after Stage 1, at the following blocks to Stage 1 at the beginning of the disk.
- Stage 2 – GRUB main code. From here the graphical menu is loaded and displayed. After choosing the OS, GRUB will load the Kernel into the memory and run it, either from local disk or from network. Reading configuration files.
- initrd – Initial RAM Disk, contains kernel modules for accessing root filesystem.
- GRUB Does not chooses boot sector by active partition.

Flow Structure

BIOS → MBR (Stage1 - 446bytes) → Stage 1.5 (about 7KB) next blocks between MBR to 1st partition → Boot Sector (Stage2 - 20KB-30KB) → Loading the Kernel / Loading other Boot Loader (which will load its Kernel) through Chainloader and loading initrd (initial ramdisk)→ running the Kernel → mount filesystem -> /sbin/init (switch to multiuser - init level 3) -> getty*7 -> shell (bash, etc.)

Structure and Syntax

The reference to Disk and partitions goes by number of the hard drive and the number of the partition. The count starts with 1 (although older versions started with 0) so for the first Hard drive and first partition I will use (hd1,1). There is no difference between Ide, SCSI and Sata. You will reference all of them with hd#,#.

- Extended/Logical partition always starts at 5 and up.

Important files

GRUB files located in "/boot/grub". The most important files you can find there are:

- grub.cfg – Configuration files which is loaded when GRUB starts. Mainly contains menu entries, the entries you will see when you get to GRUB graphical menu.
  This file is automatically generated
- "/etc/grub.d" - contains GRUB scripts, from which grub.cfg is generated
- "/etc/default/grub" - contains general settings for grub such as which entry is the default.

Important Parameters

Initrd location -
linux – where is the vmlinuz file location and which partition holds the root file system.
root location – where the configuration files located.
Default – default boot option.
Timeout – time to wait until booting default.

Rootnoverify – a partition in which other boot loader resides, windows for example.
Chainloader –tells to replace the current boot loader in memory with other one, because the current boot loader cannot load other's kernel image to the memory. It's also called **indirect boot**. The boot loader main code resides most of times at the boot sector, so the chainloader will call a specific boot sector.

Install and update

To install GRUB at MBR, run "grub-install *{device full path}*". Device full path for example: /dev/sda.

To update grub entries and settings, run "update-grub". it will regenerate the grub.cfg file.
The generate process reads all GRUB scripts (from /etc/grub.d) and /etc/default/grub, then builds grub.cfg file from them.

Further Reading

http://www.dedoimedo.com/computers/grub-2.html
http://www.pixelbeat.org/docs/disk/

LILO

LILO – Linux Loader. An old boot loader.

Main Processes

Init

The father of all the processes. Has Process ID 1. starts right after the kernel has loaded.

Path: /sbin/init

Startup Managers

Ubuntu uses two startup methods: sysVinit and upstart.

SysVinit is the old method, but some of the services are still written to use it.
Upstart is the new method, so canonical services might work with it and some others. It's more advanced.

SysVinit

Introduction

The first startup manager in linux based Operating System.
In fact, it replaces sysVinit init process!
Upstart init process runs both sysVinit and upstart jobs.

Where is this startup manager in use?

First, sysVinit comes with every GNU/Linux distribution, so ubuntu includes this system, although it isn't widely in use.

The distributions which uses sysVinit:

- redhat / fedora / centos

Run levels

There are 6 run levels:

0 – halt

1 – single user

2 – multi user

3 – multi user with networking (default for server distributions)

4 – optional – not in use

5 – garphical (x11 based) (default for desktop distributions)

6 - reboot

Important Files

/etc/init.d – contains deamon scripts used by System V init tools (SysVinit).
/etc/rc*.d – each runlevel has its scripts here. each script is a symbolic link to /etc/init.d directory.

/etc/rc*.d → /etc/init.d

<u>init.d</u>
each script in /etc/init.d can get one parameter which determines what the script will do: start, stop, status, restart. (there are more values but they are script specific)

if one service depends on another service, it will call it inside this script/job. Each job manage its own dependency.

<u>rc*.d</u>
Script that starts with K (aka kill) are the scripts for stopping the service. (symbolic link)
when you will change the init level, all those scripts will be called as "*scriptname* stop"

Script that starts with S (aka start) are the scripts for starting the service. (symbolic link)
when you will change the init level, all those scripts will be called as "*scriptname* start"

/etc/init/rc-sysinit.conf

[completion needed]

/etc/init is not containing scripts in sysVinit

commands

Runlevel – will show who was the previous run level and which runlevel you are inside now.

Init *runlevel number* – will change the run level to *runlevel number*.

Chkconfig (is not included in Ubuntu) – will show which services will run in which runlevel.

Upstart

After you understood how sysVinit works, now we'll move to upstart.

Introduction

Upstart is an event based startup manager.

Instead of running scripts sequentially in SysVinit startup manager, the scripts are running in parallel.

Upstart is event driven, that means that some scripts will start running just after an event has occurred. For example, applications which depends on the net, will start running only after network connection has been established.

Upstart has the ability to build dependencies between scripts. (just like services mechanism in windows)

upstart is used mainly for startup and shutdown processes.

How does it works

When you use upstart, it works in conjuction with sysVinit.
In fact there are 3 runlevels -

0 – halt

1 – single user mode

2,3,4,5 – full multi-user mode with graphical mode. (**runlevel 2 is the default)**

6 - reboot

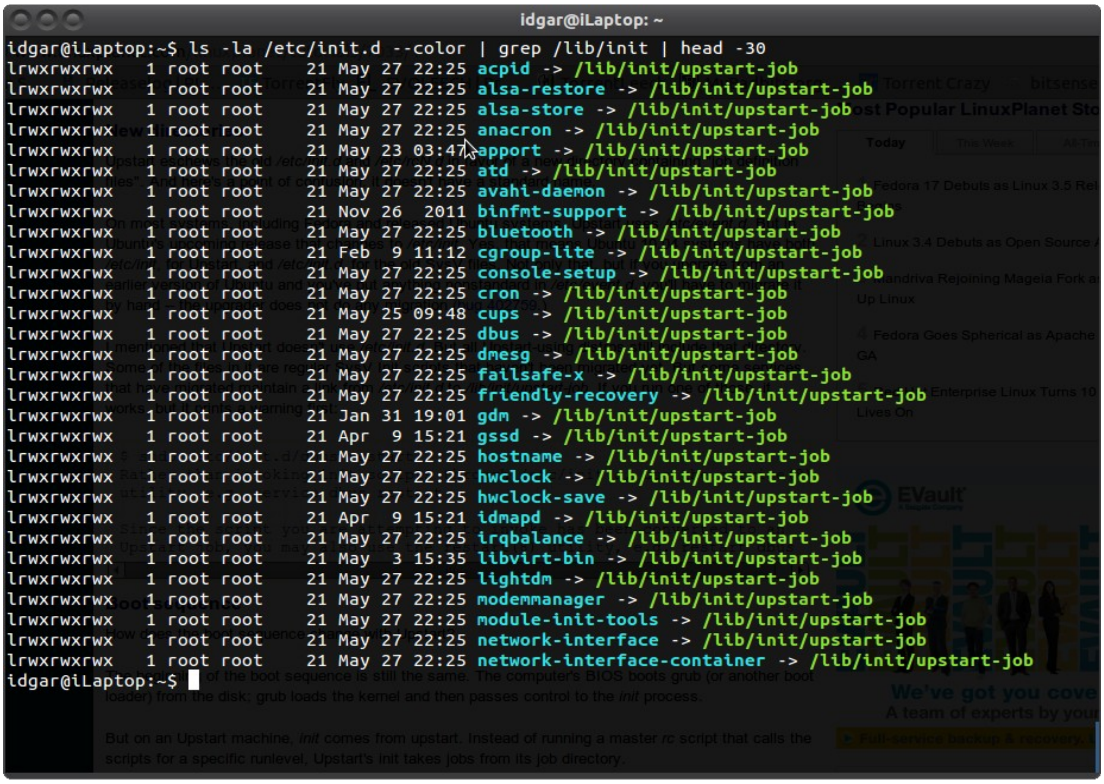
Important files

/etc/init – configuration files used by Upstart to start, stop and reload deamons.
Those are .conf files. **Those jobs replaces /etc/init.d jobs! Not using /etc/init.d scripts!**

Each upstart definition in the .conf files called **stanzas**.

To change when a service starts/stops – you need to change those files.

some of the scripts which has migrated to upstart scripts, are linked from /etc/init.d to /etc/init (linked through /lib/init/upstart-job):



```
idgar@iLaptop:~$ ls -la /etc/init.d --color | grep /lib/init | head -30
lrwxrwxrwx   1 root root    21 May 27 22:25 acpid -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 alsa-restore -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 alsa-store -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 anacron -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 23 03:47 apport -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 atd -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 avahi-daemon -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 Nov 26  2011 binfmt-support -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 bluetooth -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 Feb  9 11:12 cgroup-lite -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 console-setup -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 cron -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 25 09:48 cups -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 dbus -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 dmesg -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 failsafe-x -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 friendly-recovery -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 Jan 31 19:01 gdm -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 Apr  9 15:21 gssd -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 hostname -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 hwclock -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 hwclock-save -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 Apr  9 15:21 idmapd -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 irqbalance -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May  3 15:35 libvirt-bin -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 lightdm -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 modemmanager -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 module-init-tools -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 network-interface -> /lib/init/upstart-job
lrwxrwxrwx   1 root root    21 May 27 22:25 network-interface-container -> /lib/init/upstart-job
idgar@iLaptop:~$
```

/etc/init.d – keeps sysVinit scripts, but for those who has converted to upstart jobs, they are referenced to upstart jobs. This referencing is for cases when someone tries to call upstart job the way he used to call sysVinit job (if someone got confused). Anyway, upstart will give you a comment about that you running upstart job in sysVinit way in order to avoid it next time.

The route for call services that has upstart job, in the way you used to call sysVinit job:
/etc/init.d/*service command* → lib/init/upstart-job → /etc/init/*service.conf command*

Configuration

For those who worked with SysVInit before, if you want to get similar runlevel 3, stop lightdm service (service lightdm stop).

[completion needed]

How to use

Command:

start *service*

stop *service*

status *service*

initctl list – list of services and their status

Another option is to use service command:

initctl version – upstart version.

Service -

Further Reading:

http://www.linuxplanet.com/linuxplanet/tutorials/7033/1

http://upstart.ubuntu.com/wiki/Stanzas – stanzas list

http://upstart.ubuntu.com/cookbook/ - detailed explanation about upstart structure and how .conf files works.

Summary

The relevant things:

Init – **upstart** process management. Runs both **upstart jobs** and **sysVinit** scripts.

Start/stop/status *service (*shortcuts for initctl start/stop/status) – manual commands for upstart jobs.

initctl list - list all the **upstart** services.

Service –status-all – list all **sysVinit** services (and also shows upstart services).

/etc/init – upstart scripts (configuration files)

start on runlevel [2345] – will run the service at runlevel 2,3,4,5

stop on runlevel [016] – will stop the service at runlevel 0,1,6

exec/script – what to do when starting the job

pre-start – what to run before the main service. If there is no exec/script, will refered as start event.

pre-stop – action before terminating the service

post-stop – action after terminating the service

stopping services performed as sending signal (default – SIGTERM. After timeout period sending SIGKILL)

<u>init that still works</u>:

/etc/rc*.d – jobs that will start and stop when changing runlevels. Only services that haven't been written to work with upstart.

/etc/init.d – sysVinit scripts (and links for those who converted to use upstart method).

Manage automatic start/stop

For those who have upstart script – edit /etc/init/*script*.conf script.

For those who don't have upstart script – add or remove /etc/rc*.d symbolic links.

To change default runlevel:

edit /etc/init/rc-sysinit.conf – env DEFAULT_RUNLEVEL=*desired_runlevel*

Terminals

General

In Linux there are Virtual Terminals or VT. The physical machine run by default 6 Text VT and Graphical VT (most of time at desktop/workstation distributions). Graphical VT also called Virtual Desktop.

Virtual Terminals Enables connecting several users concurrently.

How to switch between Virtual Terminals?

You can switch between them by clicking Ctrl-Alt-F# (1-6).

To get to the graphical terminal, click Ctrl+Alt+F7.

Another option is to run the command chvt [terminal number]

for example: chvt 1

Textual Terminals

To start a textual terminal, the application “getty” starts.

/sbin/getty

by default, init calles to 6 textual terminals: tty1 to tty6, that means you will see 6 processes of getty.

change resolution:
to change resolution all you need to do is to add vga=[number] to the kernel line in one of the following:

1. at the grub.conf file (not existed in grub2 hence not exist in Ubuntu 12.xx)
2. at grub menu, you can edit the line for the menu entry.
3. Change /etc/grub.d/10_linux script (complicated)

vga=791 – 1024x768 16bit
vga=792 – 1024x768 24bit
vga=794 – 1280x1024 16bit
vga=795 – 1280x1024 24bit
For example:

```
title           Debian GNU/Linux, kernel 2.6.18-4-k7
root            (hd0,0)
kernel          /vmlinuz-2.6.18-4-k7 root=/dev/hde2 ro vga=791
```

```
initrd          /initrd.img-2.6.18-4-k7
```

Graphical Terminal

/usr/bin/X

- X - starts X server (without any window manager or Desktop Environment)
- X :# - Start X server in other VT, starting at 0 == VT7 and so on. (X :1)
- startx - To start your default X window manager / Desktop Environment [determined by – completion is needed] run “startx – :#” where # is the number of graphical virtual device (0==7 and so on).
- First, startx will search for ~/.xinitrc. This is a file which runs the window manager.
  The only line in this file: “exec [window manager]”.
  For example “exec gnome-session”.
- /etc/X11/xinit/xinitrc – the file which startx runs at the beginning. By default, it only calls /etc/X11/Xsession.
  - /etc/X11/Xsession - Commands:
- X - starts X server (without any window manager or Desktop Environment)
- X :# - Start X server in other VT, starting at 0 == VT7 and so on. (X :1)
- startx - To start your default X window manager / Desktop Environment [determined by – completion is needed] run “startx – :#” where # is the number of graphical virtual device (0==7 and so on).
  - Xinit – run X server with only xterm.
    Xinit – :#

Processes hirarchy:
*   lightdm -> Xorg (X11)
*   -> gnome session -> unity

Display Manager

Changing your default display manager

to change the default display manager, change the file /etc/X11/default-display-manager. It holds the display manager's path.
the display manager's execution files reside on /usr/sbin. For lightdm - /usr/sbin/lightdm

You can also run the command “sudo dpkg-reconfigure gdm” or “sudo dpkg-reconfigure lightdm” (same result on both of commands) and choose the default display manager. This is the TUI style to edit the file above.

dpkg-reconfigure [name of the display manager]

Note – you have to install the display manager package first.

Examples:

dpkg-reconfigure lightdm

dpkg-reconfigure gdm

Display Manager Configuration

/usr/share/xsessions/ - this folder keeps the list of all desktop environments that each user can choose to log in to from any display manager. For each desktop environment there is a file with settings for it such as the command to run this desktop environment.

each file in this folder is a session which you can choose at the display manager screen.

The file describes the name of the session, comments and how to run the desktop environment.

/etc/xdg/autostart – the programs which will be started with each desktop environment. Each program has its own file that describes it and runs it. (.desktop file)

fields:

[Desktop entry]
Version=
Encoding=
Name=

Name[language two letters]=
Comment=
Comment[language two letters]=
Icon=
Exec=
Terminal=
Type=
OnlyShowIn= / NotShowIn=
Categories=
GenericName=

X-GNOME-Autostart-Delay=
StartupNotify=
NoDisplay=
will hide from startup applications program if the value is true (good for system/default applications)

Manually restarting display manager

sudo service [display manager name] restart

Shells

Bash

/etc/profile → $HOME/.bash_profile → $HOME/.bash_logout

- /etc/profile – the first file the textual shell loads after login.
- ~/.bash_profile -
- ~/.bash-logout -
- login messages
  - /etc/issue – the first heading appears before logging in.

applications:

- /sbin/getty
- /bin/login
- /bin/sh - bash shell

Unity

unity-panel-service -

hud-service -

indicator-application-service -

indicator-session-service -

indicator-datetime-service -

indicator-messages-service -

indicator-sound-service -

indicator-printers-service -

unity-applications-daemon

unity-files-daemon

unity-gwibber-daemon

unity-music-daemon

unity-lens-photos

unity-shopping-daemon

unity-lens-video

unity-scope-gdocs

unity-scope-video-remote

nm-applet – network manager applet which shows you the network connection status and let you change your network connections. It's the graphical front end to network manager service.

Bluetooth-applet – applet to manage bluetooth connections.

Inter-process Communication

Signals

Signals (like SIGHUP, SIGTERM and SIGKILL) are sending through IPC mechanism (which implemented in the **kernel**).

Pipes

Pipelines between commands are also sent through IPC mechansim (which implemented by the **shell**)

System V IPC

Further reading: http://www.tldp.org/LDP/tlk/ipc/ipc.html

message bus system

Called D-Bus. It's one of the Inter-process communication (IPC) mechanism which allows applications and daemons to communicate with each other. It does not require network. (it works in conjunction with linux kernel IPCs)

Processes

dbus-daemon

dbus-launch -

Devices and Hardware

Hardware

lspci

list all your PCI devices. Useful when you need to gather information about your hardware.

lshw

dmidecode

print data about your hardware such as motherboard, CPU, BIOS.

For example, you can print out the data about your memory DIMMs,number of DIMM slots on your motherboard and maximum memory capacity possible:

```
sudo dmidecode –type memory | less
```

[completion needed]

lsusb

Udev

Udev – linux device manager. Manages devices in /dev. It's dynamic and can add or remove devices from /dev.
Listens to uevents the kernel sends out if a new device is initialized or removed from the system.

- /dev -
- Hda/sda – the first sector in the hard drive which contains MBR
- sda1,sda2... - partitions.

Udevd – udev daemon [completion needed]

Upower

system wide power management.

One of the uses is to monitor laptop batteries.

processes

Upower – command line tool to monitor upowerd.

Upower –monitor-detail – will show the data fetched from upowerd at the moment it happens.

upowerd – system message bus service for upower.

Deamons / Services

All the daemon scripts resides on /etc/init.d (System V Init) directory or /etc/init (Upstart) directory.

It means that when you use the command service, it will call one of those scripts.

Upstart has precedence on System V init scripts.

Security

Manage user accounts

Users

Each account has its own UID, or User ID.

To see what is your id run the command “id”.

Groups

there are also group permissions in GNU/Linux environment. It means that you can link UID to GID.

To see to which group your user belongs to, run the command “id”.

Manage Advanced permissions

- Octal Permissions – you can determine file permissions by octal numbers. The permission consists of three digits: first to owner, second to group and third to other.
  - r = 4
  - w = 2
  - x = 1
  - 7 = 4 + 2 + 1 (permission to read, write and execute)
  - 6 = 4 + 2 + 0 (permission to read and write)
  - for example, to give all permissions only to the owner, chmod 700 will be used.
  - Rule of thumb – even digit will include execute permission, odd won't.
- Umask – remove permissions
  - umask 022 will remove write permissions to group and other.

Hardware

- Lspci - list all your PCI devices. Useful when you need to gather information about your hardware.
- Dmidecode – print data about your hardware such as motherboard, CPU, BIOS

Storage

Udisks/Udisks2

Package which contains daemon and tools to query data about storage devices and manipulate.

It can mount devices, gather SMART information, put drive into standby,

Packages: libdisks2-0, udisks, udisks2.

kernel (discover hotplug events of devices)→ udev (mount devices to /dev) → udisksd (mount devices to /media for example).

daemons

Udisksd – udisk system daemon. The communication is through D-Bus (system message bus) API. Also available through C/C++ by libdisks2. It starts by D-Bus and not with upstart/sysVinit script. (**is this part is the main?)**

(belongs to org.freedesktop.Udisks2) **the newer one?**

udisks-daemon – service for D-Bus. Automatically starts by dbus-daemon. (belongs to org.freedesktop.Udisks service in D-Bus) **the older one?**

Tools

Udisks – command line interface for Udisks Daemon to query and manipulate storage devices.

Gnome-disks – udisks GUI tool.

udisksctl – interacts with the udisksd daemon. Perform operations on the disks,

Linux FileSystems (ext)

- superblock - block which contains filesystem metadata
- inode table – [completion needed]
- Block bitmap
- inode bitmap
- inode -file table, contains unix permissions and pointer to 12-15 data blocks. tree
- inode pointers
  - 12 direct data blocks
  - 1 indirect block
  - 1 double indirect block
  - 1 triple indirect block
  - there can be numerous inodes for a file (hard link) or link to the inode (soft link)
  - indirect block
  - double indirect block
  - triple indirect block
  - inode -> direct blocks (data blocks)
  - inode -> indirect blocks (each pointer is 4 bytes) -> data blocks
  - total indirect pointers determined by block size. for example, at 4KB block size indirect block contains 1024 pointers.
  - datablock can be file or directory
  - directory datablock contains list of inodes.
  - total inode quantity at filesystem can be fixed or dynamic.

  - advanced filesystem features:
    - compressed
- directories by list or by binary tree

linux and defragment

file de-fragmentation – file's blocks are contagious. It means that the whole file is in one area.
The ext3/4 filesystems will not create fragments for files when you create or edit it.
If you edit a file and add data to it, if there is no space after the file, ext3/4 will move the file to another area.

- When file is created or changed, the filesystem will make sure it will be contagious, even if it needs to move to another location.
- After each file, there will be empty space.

http://geekblog.oneandoneis2.org/index.php/2006/08/17/why_doesn_t_linux_need_defragmenting

contagious related files – related files are put close to each other.
It's better to order the files at the filesystem in a way that if you read hundred of files at the same time and sequentially. For example, to order boot files at the same area in an particular order.
Linux doesn't do this.
http://blogs.kde.org/node/2270

Mounting FileSystems

Mount

location: /bin/mount

additional filesystems mounting: /sbin/mount.*

umount

/etc/fstab – filesystem table. You can put there your permanent mountpoints and each time your system will boot you will see them. [completion needed]

how to know the UUID of a partition?
Just run ls -la /dev/disk/by-uuid and you'll see which partition each UUID is linked to.

- device label - abbrebriate to hard drive name instead of /dev/sda*

FUSE

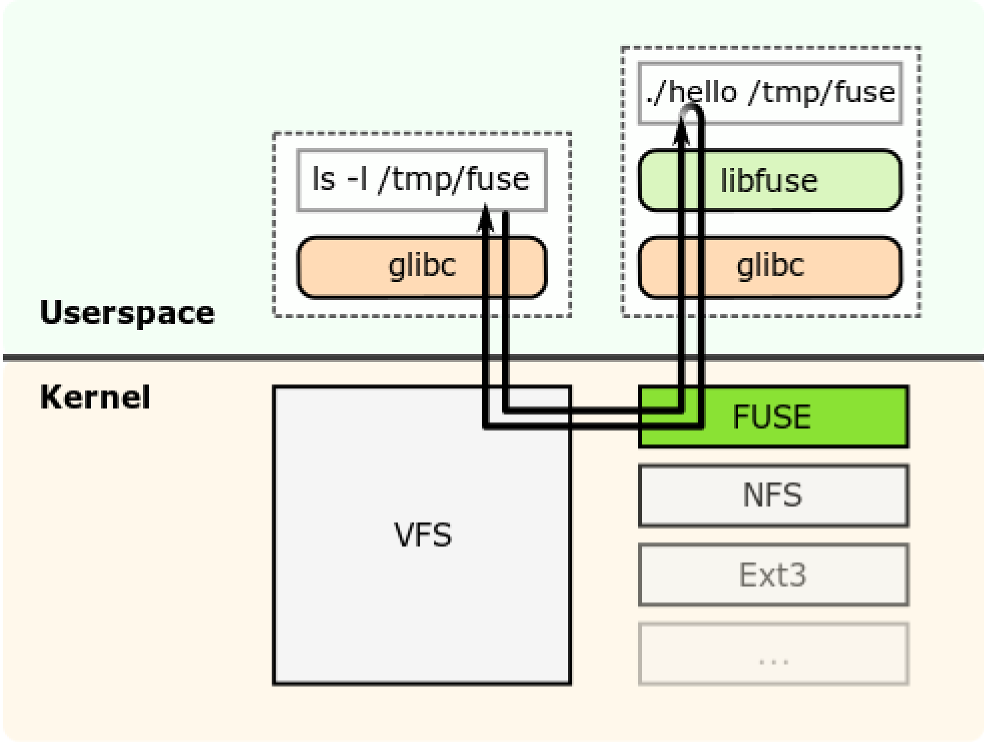
Standard linux filesystem such as ext2 to ext4 are compiled into the kernel. So how do we use other filesystems that are not compiled into the kernel?

Of course, one option is to recompile the kernel, but it makes it complicated.

The other way is to use programs that doing it.

FUSE is Filesystem in Userspace, is a program which allows us to use variety of filesystems.

Non privledge users can mount filesystems.



As you can see in the diagram, FUSE itself compiled into the kernel, but the specific filesystem is in the userspace (in this example libfuse filesystem)

Filesystems implementation that uses FUSE:

- ntfs-3g

Notes:

- filesystem in userspace is less efficient than kernel filesystems hence there is performance degragation, although the panelty is not high.

Mounting NTFS filesystems

/etc/fstab:

```
UUID=<Partition UUID>           <mountpoint>    ntfs-3g  uid=<your user id>,gid=<your group
id>,umask=002 0          0
```

notes:

types *ntfs* and *ntfs-3g* are the same.

DMRAID

A driver to manage fake raids. It's actually perform raid operations such as write a data to sevaral disks (raid1) or calculate parity (raid5). Uses the cpu.

When do you use it? Especially when you configure raid at home with intel chipsets.

In intel you may see the terms “Rapid Storage Technology” or RST. Another term is “Intel Matrix Storage Technology” - the old name.

When you configure disks with raid from the BIOS, it will take several MB from the end of your disks to write metadata (usually about 3MB).

The common commands:

dmraid -s : status

dmsetup status : ?

dmraid -r : drive attached to raid

dmraid -ay : activate raid (USE CAREFULLY!)

dmraid -r -E : remove device from raid (USE CAREFULLY!)

Networking

NetworkManager

This application manages all network configuration, starting from connection to SSID in wireless and to get configuration from dhcp and so on.

It replaces ifup/ifdown scripts.
Plus, you can see in /etc/resolv.conf that is pointing to 127.0.0.1, that's because he's pointing to network manager which will point to the right DNS by network manager settings.

Nm-connection-editor – The basic editor for network configurations in Ubuntu. This is the easiest way to configure your NICs.

Nmcli – get detailed information about interfaces. Will give you information about configurations from dhcp server.
nmcli dev list iface [interface]

nm-tool – get information about interfaces. This tool will give you in one command your IP Address, Subnet Mask, Default Gateway and DNS Server.

Nm-applet – an applet which loads by default and can be found at the unity panel in the indicators section.

Nm-tool | tail – 8

for more information about network manager - https://help.ubuntu.com/community/NetworkManager

DNS Client

By default, domain names identifies as computername.local. If you tried for example ping another computer in your home network and haven't succeeded, adding to the computer name the suffix “.local” might work.
.local is a pseudo-top-level domain, a default domain when there isn't a domain.

Dnsmasq – dns resolver daemon. Starting at Ubuntu 12.04, this daemon is been installed by default.

To use this daemon, /etc/resolv.conf is always pointing to “nameserver 127.0.0.1”, although dns server that has configured by dhcp will get higher priority.

this daemon enable using various dns servers depending on suffix. So one domain will go to DNS x, while other will go to DNS y.

this daemon has caching capabilities (as opposed to the era before, Ubuntu hasn't used dns cache mechanism).

Dnsmasq using /etc/hosts.

/usr/sbin/dnsmasq

Dhclient -

To resolve windows computer names, follow the instructions in http://that-matt.com/2009/12/how-to-get-ubuntu-to-ping-a-windows-hostname/

and after changing configuration files, run sudo service winbind restart

/etc/resolv.conf – configure your DNS Servers. DO NOT CHANGE IT MANUALLY! This file is being updated by resolvconf application.

Resolvconf – Starting Ubuntu 12.04, there is a new service which automatically update /etc/resolv.conf file.

If you want to add static dns servers, you can do it by adding values to /etc/resolvconf/resolv.conf.d/
for more information: http://www.stgraber.org/2012/02/24/dns-in-ubuntu-12-04/

nslookup – a command which searches for ip address for domain name from dns servers listed In /etc/resolv.conf.

DHCP Client

Dhclient – controls dhcp client. Can release and renew addresses.

```
Sudo dhclient -r
```

will release ip addresses which has assigned by dhcp.

```
Sudo dhclient [interface]
```

/sbin/dhclient

Note – those commands will update /etc/resolv.conf file and routing table (can be shown by “route -n” (direct/indirect by dnsmasq? – completion needed)

will renew/get Ip address and other data from the dhcp server. **I recommend noting the interface as a parameter,** if you have numerous of interfaces, it will try to renew addresses at all the interfaces and if some of them are not connected, it will take time.

http://en.wikipedia.org/wiki/.local

Avahi

A daemon which implements ***zeroconf*** (zero configuration networking) protocol. It's a protocol for use when you don't have a network with DHCP or/and DNS.

Each device connected to network without central services (DHCP / DNS), will allocate a randomize address in segment 169.254.0.0/16.

instead DCHP. this technique enables to communicate peer-to-peer between two devices.
This technique calls ***link-local. (also IP4LL)***

Instead DNS, zeroconf protocol uses ***mDNS*** (multicast DNS) protocol. Each device choose a domain name in the local DNS namespace (e.g - .local). For example, if your computer is mycomp, the FQDN will be mycomp.local .

After determining FQDN, the protocol announce this name by multicasting it. Each device manage its own DNS list.
When a device wants to know an IP address of a domain name, it sends a reqeuest to 224.0.0.251 (multicast), the computer which owns this name will answer with its IP.

This technique is actually distributing domain names on the network instead of using central domain name server.

Also, the implementation of zeroconf include the protocol **DNS-SD** (DNS Service Discovery).
Each host has services consist of: service's protocol/type, transport type (tcp,udp...) and the port.
When a device wants to discover a service type, it sends a packet with the first two parameters above:
<service>.<transport>.<domain>
If there is a device which serve this, the DNS or mDNS will return a packet:
<device name>.<service>.<transport>.<domain> .

Uses 2 ports: 5353 (mDNS) and 32768. both UDP.

For IP4LL, You might be familier with APIPA from microsoft, which refers to addresses 169.254.0.0/16, a segment for address autoconfiguration.

Apple impementation to this protocol calls bonjour (previously rendezvous)

Notes:

- all .local FQDN will not go to DNS server but will use mDNS!
- <u>Without this daemon you will probably will not able to resolve computer names in your LAN.</u>
  Ping work with avahi along with dns.

<u>daemon</u>: avahi-daemon

commands:

avahi-browse – discover services with mdns-sd protocol. The simplest one is “avahi-browse –all”.

Avahi-resolv – will resolv host name to IP address and vice versa. For example, “avahi-resolve –name <host name.domain> -4” or “avahi-resolve –address”.

Interface Configuration

Ifconfig – [completion needed]

you can edit configuration through “network connections” GUI tool.

Routing Configuration

Gateway

sharing protocols

Nfs

NFS Server

nfs server is available through the service/package nfs-kernel-server.
In order to export nfs volumes/directories you have to edit /etc/exports file.
This file contains the directories you are about to share and to whom.
The nfs version at Ubuntu 12.04 by default is 4.

Sudo apt-get install nfs-kernel-server

To see which exports are already exposed you have two options:

1. read /etc/exports (the hard way)
2. showmount -e

NFS Client

nfs-common package – this is the package you need in order to mount nfs shares.
Sudo apt-get install nfs-common

showmount -e [server name] – will show all the shares the server exports.

To mount, you have two choices:

1. by adding a new entry to /etc/fstab
    1. [servername]:/[export path] [mountpoint] nfs defaults
2. sudo mount -t nfs -o [options] [servername]:/[export path]

sources: http://www.cyberciti.biz/tips/ubuntu-linux-nfs-client-configuration-to-mount-nfs-share.html

cifs / samba / smb

Samba server

If you want to share linux files to windows clients you have to install *samba server*.
Installation: sudo apt-get install samba

to configure which folders will be shared, you have to edit the file “/etc/samba/smb.conf”.
Easier way is to install the tool system-config-samba which is the GUI version and edit smb.conf file.
Sudo apt-get install -y system-config-samba
For further information, go to https://help.ubuntu.com/11.04/serverguide/samba-fileserver.html

important things to know:

1. to use an authentication, the user you are using in the windows side has to be samba's domain. For example workgroup\tux. You can configure which domain is associated with your ubuntu by editing smb.conf or by the tool.
2. You have to associate samba users with linux users, you can do it by the tool or else you cannot access remotely with credentials.

Services:

1. smbd – samba server
2. nmbd – netbios server

Samba Client

samba client comes built in with ubuntu.

If you want to access windows shares, you can do it by nautilus → browse network

another option from file manager / nautilus:

smb://[computername]/[share] (like \\[computername]\[share] in windows)

you can connect to smb share through command line:

smbclient //[computername]/[share]

and to list available shares:

smbclient -L //[computername]

another option is to mount cifs shares by:

1. *mount* command.
2. Adding a mountpoint to fstab. The filesystem type is smbfs.

smbfs package?

GVFS

Daemon:

gvfsd (GNOME Virtual File System – GnomeVFS)

Gvfs uses fuse to mount **network shares**. It has gvfs mountpoint which uses FUSE that resides on /run/user/<username>/gvfs (you can see that with mount command). Any share will be add as a directory inside this mountpoint (they are not real mountpoints, just virtual mountpoints which resides on gvfs filesystem).

So for the comparison, you can mount cifs shares in two main ways in linux:

1. mount.cifs - mount.cifs → FUSE → linux kernel
2. gvfs – gvfsd-smb → gvfs → FUSE → linux kernel

as you can see, mounting with gvfs is less efficient, so why using it?

The answer is simple – it's easier to use it and the user doesn't need root privileges in order to mount network shares.

allows:

- automount devices through file manager (aka nautilus as default) to read content from remote share. The mountpoints will be found at /run/user/<username>/gvfs but for simplicity you will see them at the Sidebar.
- Support Protocols: ssh, ftp, cifs, WebDav (HTTP), Secure WebDav (HTTPS)
- to mount network share use URI. For example, for ftp – ftp://*server*.

Each time you mount a new type of protocol, a new daemon/process will be loaded into memory.

For example, if you've using nautilus and now you put URI of ftp site into the location bar, the ftp site will be mounted inside gvfs. If you will look at your processes, you will find a new process - /usr/lib/gvfs/gvfsd-ftp

Processes

gvfsd – the main daemon.

gvfsd-fuse - process that responsible to mount gvfs filesystem into linux mountpoint (the default - /run/user/<username>/gvfs

gvfsd-smb-browse – process that responsible to search smb/cifs shares (or browse network – nautilus).

gvfsd-trash – trash process (like recycle bin in windows). For each filesystem, a directory named .Trash-<UID> is created it its root directory (not “/”). contains two main directories:

1. info – info about the deleted file/directory (path, deletiondate).
2. Files – the actual files/directorys that have deleted.

This process gathers all the .Trash-<UID> directories and show them in one virtual directory.

The trash can be accessed through URI: trash://

gvfsd-burn – CD/DVD creator virtual filesystem. When you want to send files to burn through the file manager. This process gathers all the paths you want to burn and in the end allows you to burn it into a disc or image file.

The burner can be accessed through URI: burn://

gvfsd-archive – mount archive files (as much as i've seen, it can mount iso)

per-protocol processes (per share):

gvfsd-ftp

gvfsd-smb

gvfsd-http

Unknown

gvfsd-metadata -

not daemons?

gvfs-gphoto2-volume-monitor -

gvfs-udisks2-volume-monitor -

gvfs-afc-volume-monitor -

command

gvfs-mount protocol://user@password ? - will mount a remote filesystem to the current directory.

Gvfs-mount -l

old methods

Gvfs-fuse-daemon – before Ubuntu 12.10, this daemon mounted your network shares into ~/.gvfs (as opposed to /usr/run/<username>/gvfs).

Notes

/media/<username>/share is not mountpoint which uses gvfs!

http://blog.colovirt.com/2009/12/07/linux-filesystem-gnome-virtual-file-system-gvfs-remote-connectivity-cli/

http://askubuntu.com/questions/61196/why-do-my-gvfs-mounts-not-show-up-under-gvfs-or-run-user-login-gvfs

VPN Server

The easiest way to configure vpn server is to install pptpd server.

```
Sudo apt-get install pptpd
```

after installing the package/daemon, you have to configure it:

sudo nano /etc/pptpd.conf
add two lines:

1. localip [server ip address]
2. remoteip [vpn clients ip addresses range]. The range can be specific address (192.168.0.20), range of Ips (192.168.0.20-25). all of there can be separated by colon (192.168.0.20,192.168.0.30-33,192.168.0.50)

next thing is to set authentication:
sudo nano /etc/ppp/chap-secrets
here you need to users with passwords

for example:

```
testuser        pptpd   Password1       *
```

the last column is set to ip addresses / subnets which can connect. If you connect to the vpn through internet it's better to put *.

now restart pptpd service:
sudo service pptpd restart

that's it! All you need to do is to configure VPN Client

for advanced settings, you can edit /etc/ppp/pptpd-options
there are settings like which authentication method are available.

Don't forget to open port 1723 to the server.

Processes

/usr/sbin/pptpd

VPN Client

Network manager has built in VPN client for PPTP connection.

Choose gateway (the VPN server's address)

put user name and password as you putted in chap-secrets file.

At the advanced settings choose only MSCHAPv2 (this is the default authentication for pptpd and the only choice).

Choose use point-to-point encryption(MPPE)

recommended setting:

IPv4 Settings → Routes → Use this connection only for resources on its network (check)

Remote connections

Protocols

Vnc -

rdp -

Telnet

SSH

openssh-server – package required to be able to login to command line shell remotely. **This one is not installed by default!**

When connecting to servers, there is a ECDSA key (public key) which is saved in ~/.ssh/known_hosts.

If you have changed the ssh server and it has a different public key, then you might have problems. To solve this you can remove server's public key from known_hosts (on the client computer) by the following command:

ssh-keygen -R [hostname]

or

ssh-keygen -R [IP]

processes:

/usr/sbin/sshd – openssh server daemon.

RDP Clients

Rdesktop – can connect to windows versiggons up to XP. Run from command line.

Remmina – the default remote desktop client. Easy to use.

xvnc4viewer – vnc client which supports RDP, VNC and other protocols.

RDP Servers

Xrdp – the easiest installation RDP Server.

dpkg-reconfigure [name of the display manager]

Sudo apt-get install xrdp

Note – you have to install the display manager package first.

Examples:

dpkg-reconfigure lightdm

dpkg-reconfigure gdm

processes

/usr/sbin/xrdp

/usr/sbin/xrdp-sesman

the file which determines who is the default display manager: /etc/X11/default-display-manager

the display manager's execution files reside on /usr/sbin. For lightdm - /usr/sbin/lightdm

and that's it! The daemon is starting right after installing the package.

Manually restarting display manager

Notes –

1. can connect with 8bit or 16bit color depths only if you're connecting with x11RDP.
2. The connections is made by using sesman-Xvnc (vnc protocol). The rdp protocol is used as a proxy.

sudo service [display manager name] restart

Display Manager Configuration

/usr/share/xsessions -

each file in this folder is a session which you can choose at the display manager screen.

The file describes the name of the session, comments and how to run the desktop environment.

General tools

Gnome-nettoolIptables – completion needed.

Nm-connection-editor – The basic editor for network configurations in Ubuntu. This is the easiest way to configure your NICs.
Gnome-nettool – completion needed.

/etc/resolv.conf – configure your DNS Servers.
/etc/network/interfaces – configure your Ethernet NICs.
/etc/hosts -

/etc/nsswitch.conf

ifconfig [interface]– shows information about ethernet interfaces. Information such as:

IP Address
Subnet Mask
MTU
MAC Address

Ifup -
ifdown -
ifconfig -

traceroute – like tracert in windows. **Does not come by default.**

```
Sudo apt-get install traceroute
```

Route [-n] – shows the routing table for networking. From there you can get information about default gateways.

Winbind – completion needed

you can edit configuration through “network connections” GUI tool.

Iptables – completion needed.

NetworkManager uses:

dnsmasq
dhclient

resolvconf uses the following to build /etc/resolv.conf:

dhclient
NetworkManager
/etc/network/interfaces (which uses as default to configure loopback address – 127.0.0.1)

DNS Server

Bind9

sudo apt-get install bind9

Monitor and control

Power Management

```
Sudo Pm-hibernate
```

hibernate the computer.

```
Sudo pm-suspend
```

making the computer going to standby mode / suspend.

Etherwake

tool to wake a computer by network packet to NICs which support Wake On LAN.

```
Sudo apt-get install etherwake
```

etherwake -i [interface] [NIC MAC address]

Network

Iftop – showing all the connections between your interface and other addresses with data about amount of bytes transferred or received.

```
Sudo apt-get install iftop
```

Nethogs – showing the traffic of each application (not by connection)

```
Sudo apt-get install nethogs
```

Wireless

wavemon – wireless monitoring tool.

Compile Programs

You will usually get programs' source code from

1. a web site which has tarball file with all the source code files.
2. Subversion

Tar xvzf [package name].tar.gz

./configure – script which creating Makefile file, the file which will be in use when you run make command.

Make – building the binary files.

make install – installing the program after the binary files are ready. It copies the files to the appropriate folders such as /usr/bin.

make clean – deleting the binary files from the current directory. Because make install already copied the files, you won't need them anymore.

Make uninstall – sometimes there is an option to uninstall the program.

For more information: http://www.tuxfiles.org/linuxhelp/softinstall.html

more about pptp VPN:

http://www.serverhostingsecrets.com/tutorials/ubuntucentos-pptpd-howto/

Other services

Modem-manager

bluetoothd -

Memory

Flushing swap data

```
Sudo swapoff -a && sudo swapon -a
```

Clearing cache data from physical memory

```
Sync && echo 3 | sudo tee /proc/sys/vm/drop_caches
```

you will see that the buffers will be flushed.

```
free -h
```

the first line shows

1. total memory
2. used memory – **includes cached data**
3. free memory – **after taken cached into account**

the second line shows:

1. used memory – **without cached data**
2. free memory – **without taken cache into account (actual memory free for applications)**

linux always uses cache mechanism to save data from the hard drive for example as long as it has unused memory.
Whenever application needs this memory, the cache memory releases (proportional to the amount needed).

buffers – for use when reading data from block devices. (for example, when an application read files from one of your devices, it will remain at the memory). **Most of times this will be the additional memory that will be in use compares to cache.**
Cache – application data and other things. When application wants so save buffer/cache, it will save it in this area. For example, when you open your browser and watch youtube, the movie buffer/cache will be saved in this area.

For summary: buffer is used for kernel mode applications as cache is used for user mode applications.

Sound

asound

Alsa project.

[completion needed]

Pulseaudio

The default sound card manager in ubuntu.

/usr/bin/pulseaudio

Printer

CUPS

Common Unix Printing System. A subsystem for managing printers. It's actually a deamon in your computer.

to install a printer go to address *localhost:631/admin*

Processes

/usr/sbin/cupsd – cups daemon.

Monitoring

Performance monitoring

Top

Load avarage – 1, 5 and 15 minutes cpu load checking.

http://blog.scoutapp.com/articles/2009/07/31/understanding-load-averages

Htop

Similar to top, but more sophisticated. For example, you can go up and down between the processes.

iostat

tool to measure I/O operations on devices and CPUs. I personally use it for drive statistics.

```
Iostat [device name] -d [update interval] -m
```

for example: iostat sda -d 1 -m

installation:

```
sudo apt-get install sysstat
```

Sensors monitoring

In order to monitor sensors (voltage sensors, temperature sensors and fan sensors) you have to install the basic package *lm-sensors:*

```
sudo apt-get install lm-sensors
```

This package will let other tools to get the monitoring data.
In order to make lm-sensors work, the first thing you need to do is to detect the sensors in your computer:

```
sudo sensors-detect
```

this progress will ask you which type of sensors to detect.
At the end it will save the sensors in /etc/modules.

After that, you have to restart lm-sensors by:

```
sudo service module-init-tools restart
```

now, run the command *sensors* to verify that you can get monitoring data, at least from your CPU.

and from here you'll just to the next step – installing tool to watch this data.

For example, you can use *indicator-sensors* package.

```
sudo apt-get install indicator-sensors
```

Hdparm -

logging

Syslog

Syslog is a protocol which based on client-server architecture for logging messages from clients into a server. It's similar to *windows event log* concept. It's a protocol.

Each message can be up to 1024 bytes (1KB).

Udp / tcp -

Default format:

[timestamp] [host] [application/process]: [message]

Fields

- Facility -
- Severity -
- Timestamp -
- Host -
- Tag -
- Message -

syslogd

messages collector. It listen to incoming messages from applications / network and writes it on files. It's actually uses “Push” method. Clients has to know the server and sent it the data. The server does not gather the data (“Pull” method).

Syslog deamon can also filter messages, so it won't save them.

Klogd

sysklogd

combination of syslogd and klogd.

This service can only filter messages by the regular fields. It means that if you want all your messages that came from specific ip will go to another file than /var/log/syslog, you won't be able to do it.

Rsyslog

The default log service of Ubuntu 12.04.

the advantage of rsyslog is that you can filter by a lot of criterias.

Processes

Rsyslogd -

Configuration files

/etc/rsyslog.conf – rsyslog deamon configuration file. This is the main configuration file.

/etc/rsyslog.d/50-default.conf – rules files. Here you can tell rsyslog to split specific messages to other log files.

Rule Sets

Configuration files

/etc/default/syslogd

Configuration file for running sysklogd service.

If you want to get log messages from another computer/device through the network, you have to edit this file in order to enable the listener on UDP 514.

this value has to be: SYSLOGD=”-r”

/etc/syslog.conf

facility.severity		*file to put the logs*

Log Files

/var/log/syslog – the main log file.

Services

Service sysklogk – the main service which write logs to /var/log/syslog file.

Task Schedule

Cron

Job scheduler which came from unix Operating System

```
sudo apt-get install gnome-schedule
```

how to use it?

For tasks of

Crontab -e

processes

Cron -

System Settings

Changing host name

In order to permanently change your host name, you need to edit **/etc/hostname** file, which contains the host name.

Another thing to do is edit /etc/hosts file, which contains your host name, connected to 127.0.0.1 (alongside with localhost)

After changing, reboot the host.

Application services

<u>Ubuntu One</u>

ubuntuone-syncdaemon

<u>Software Updater</u>

update-notifier – the process which notifies about package updates.

Application configuration files

User Applications

You can find configuration files in the following directories:

- ~/.config/*appname*
- ~/*.appname*
- ~/.kde/share/apps/*appname*

usually application will save the configuration only in one of those directories.

System Applications

- **/etc/*appname***
- /usr/share/*appname* (sometimes)

Unknown

1. gnome-keyring-daemon – keep passwords and other secrets for users
2. gnome-session – starts gnome desktop environment.
3. gnome-settings-daemon -
4. gnome-screensaver

hardware?

1. acpid – power management

2. irqbalance -
   /usr/sbin/irqbalance

3. Rpcbind -
4. rpc.* -
5. polkitd -
6. colord -
7. whoopsie -
5. atd -
6. gdomap -
7. accounts-daemon -
8. console-kit-daemon -
9. rtkit-daemon -
10. winbindd -
11. ssh-agent -
12. gtk-window-decorator -
13. goa-daemon -
14. geoclue-master
15. ubuntu-geoip-provider
16. signond
17. signonpluginprocess
18. pxgsettings
19. system-service-d
20. bamfdaemon
21. signon-ui

Chapter 11 - Scripting within the textual shell (Bash)

$? - exit status/code from the last command / application.

Meatcharacter - "\", when you just want to print special character and not to interpret it.

When you put a command argument within single quotes (which means ' '), the shell will not intrepret special characters. On the other hand, when you use double quotes (which means " "), the shell will interpret special characters.

GUI

Gksudo

Gtk+ sudo command. Instead of asking permission (aka password) at the command line, a new window will be shown to ask for the password.

Gsettings

An API to access key/value properties. Its uses are for applications

gsettings [schema key] [ ] [ ]

for example:
gsettings set com.canonical.notify-osd

gconf

Dconf

Dconf-editor

Notify-send

A command that sends a message to the notify OSD balloon.

Chapter 12 – Troubleshooting

The Problem:

You've installed something for your Desktop Environment and now you got stuck – your windows have no borders, the mouse moves but cannot do anything or something like that.

The Solution:

1. press ctrl+alt+T, run “unity –reset”.
2. Change to one of the Textual VT, run “sudo service lightdm restart”
3. if you're using compiz / Unity 3D – Try delete directory ~/.config/compiz-1
4. delete from your home directory a directory name “.config” (LAST RESORT!)

Chapter 13 - Backup Your Data

Deja-dup

The default ubuntu backup application.

deja-dup-monitor – process [completion needed]

duplicity – command line backup tool. Deja-dup uses it?

Rsync

Rsync is a program which backup data between two sides.

It can be locally at your machine between two directories (on the same device or another device), or it can be between your computer and a server (the remote machine).

One Way Backup

Most important parameters:

a – archive mode. Will transfer recursively all the directories from the source to the destination.

v – verbose mode. Will print out the actions.

z – compress the data.

c- skip calculating checksum at the destination.

h- human readable output

--progress – shows progress bar

--exclude-from=*[list file]*

Note:

"-c" and "-z" require write permissions at the destination's directories!

rsync -avz *[source directory] [destination directory]*

Rsync Local

Rsync remote

Rsync -avz *[source directory] [remote machine name]*:*[destination directory]*

Rsync Deamon

Chapter 14 – Advanced Working with X Server

Ubuntu is based on x.org server since version 5.04 (the second release).

Before, it used xfree86.

Xfree86 isn't develop much, for this time (2013), the most recent version is from december 2008, means that it hasn't been improved for 4 years...

why ubuntu moved to x.org? Mainly because xfree86 has changed the license in february 2004 and since then it's not compatible with GPL.

Machine_name:*display_number*

VDU – Virtual Display Unit.

$DISPLAY – the environment variable which tells each X application where to show its graphics.

display number 0 is 7!

xhost *x_client_machine_name* – tells your X Server that the client machine name can run applications under your X Server.

Running Graphical Applications Remotely

More Materials

Vmware workstation

To uninstall

```
sudo vmware-installer -u  package-name
```

recompile vmware workstation modules:

```
sudo vmware-modconfig --console --install-all
```

in order to run ESXi inside vmware workstation, you have to change permissions to vmnet network device.
For example, if you're using vmnet2, do the following:

```
sudo chgrp your_username /dev/vmnet2
sudo chmod g+rw /dev/vmnet2
```

and that's it!
Note: this change will be not permanent. So it's better to run hosted ESXi as shared vms (vmware-workstation-server service will run them)
Vmware kb about this issue: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=287

Performance optimization:
http://faq.sanbarrow.com/index.php?action=artikel&cat=78&id=76&artlang=en
http://iosadchuk.blogspot.co.il/2011/09/tuning-vmware-workstation-8-performance.html
http://vmfaq.com/entry/25/
https://www.nnbfn.net/2009/07/vmware-and-ntfs-3g-poor-performance/ - problems with vmem files which resides on ntfs partition (uses FUSE ntfs-3g)

the parameters for vmware workstation memory management: (in /etc/vmware/config)
(the first two parameters are those whom you can configure through the GUI)

prefvmx.minVmMemPct = "100"
prefvmx.allVMMemoryLimit = "16384" - how much maximum RAM (& host swap?) vmware workstation will be able to use (memory limit)

prefvmx.useRecommendedLockedMemSize = "TRUE" - this tells VMWare whether to use a fixed sized memory chunk or balloon and shrink memory as needed?

mainmem.backing = "swap" – will open vmem file only when suspending the virtual machine. That means that if the virtual machine needs swap, it will use your host swap. RECOMMENDED IF YOU HAVE ENOUGH MEMORY, ALTHOUGH USES SWAP (BETTER SWAP PARTITION USAGE THAN VMEM FILES)

another option:
mainmem.backing = "named" – will open vmem file for each vm. It will be used for both supsended virtual machine and for virtual machine swapping. DO NOT USE IT IF YOUR VIRTUAL MACHINES RESIDES ON NTFS PARTITION! WILL BE EXTREMELY SLOW!

Note: when using “swap”, vmware workstation allocate memory at the beginning from the “cached” area (not from “buffered”). If in the actual moment it doesn't have enough space to allocate more cache, it will not remove other application's cache but will allocate space in the swap area.

Music encoders

```
sudo apt-get install lame
```

Unlock cdrom

```
sudo fuser -km mountpoint
```

Extract tar.gz file

```
tar -zxvf filename.tar.gz
```

Switch off cache for device (not permanent)

```
sudo hdparm -W 0 /dev/devicename
```

Mono

Implementation for “.NET” framework. Implement CLR (Common Language Runtime).

Just like you need to install .net framework software in windows to interpret MSIL (Microsoft Intermediate Language, the .net machine code language which is not CPU instruction set, it's one layer above), you need to install mono to be able to run MSIL programs.

Mono cannot run standard windows programs that are not written with .net!

Also, .net applications which uses win32api won't work either.

- Mono does not run windows installer / MSI. Thus, you can run unpackaged .net applications.

Wine

[completion needed]

[correct terms]

A suite that imoplemets:

- Win32API (windows nt kernel system calls) – implements it in user space rather than kernel space.
- installation manager
- virtual registry
- alternative dlls

daemon calls wineserver (/usr/bin/wineserver) – implements windows kernel.

Wincfg

Wine Configuration tool -

Wine registry

Wine implements its own registry.

important files:

- system.reg – HKEY LOCAL MACHINE
- user.reg – HKEY CURRENT USER

to run regedit tool, just write in the command line – wine regedit.

Important files

~/.wine/drive_c – your virtual drive c. will contain all windows applications that you installed through wine.

Run wine from command line

Wine *windows executable file.*

Add / remove programs

Just run the command “wine uninstaller”

Transmission

Transmission-common

Common files for transmission bittorrent client. (library files?)

installed by default

Transmission-gtk

Bittorent client. Uses transmission-common libraries.

installed by default

```
configuration files - $HOME/.config/transmission/settings.json
```

Transmission-cli

Uses to send commands to transmission-daemon.

Transmission-daemon

The service / server which runs in the background.

Sudo service transmission-daemon stop/start/restart

once you've installed transmission-daemon, you can run the web client in localhost:9091.

User: transmission

password: transmission

configuration file (you don't have to configure it through command line. Some GUIs have configurations):

sudo nano /etc/transmission-daemon/settings.json

edit this file only when the daemon is down/stopped! Or else the file will be overwritten when it will restart (so it will never get your new configurations)

1. the path in which you are going to save/download the files into, has to be own by *debian-transmission* group!
   Change the group owner of the folder and add your user to the group (usermod -a -G *groupname username*)

transmission remote GUI

An application to connect to transmission-daemon instead of webGUI.

One way to install (ppa, but old package)-

```
sudo add-apt-repository ppa:ajf/trg
sudo apt-get update
sudo apt-get install transmission-remote-gtk
```

the other way:

download it from *http://transmisson-remote-gui.googlecode.com/files/transgui-4.1-x86_64-linux.zip*

Notes:

1. Transmission-gtk and transmission-daemon saves their configurations in different places!

Glossary

Term	**Explanation**
SuperUser	The root user
Console	Group of keyboard, mouse and screen which consist together a **physical terminal** which most of time attached **directly** to the machine. It always exists.
Command Prompt	The sign which tells you that the terminal is waiting for a command. In ubuntu it's $ when you use a regular user and # when you use the root user. for example: user@computername:~$
Pty	Program Terminal
PID	Process ID
PPID	Parent Process ID
POSIX	Portable Operating System Interface (uniX) – standards by IEEE for shells and utilities in variants of unix operating systems. So each program will act the same at all the operating systems and will have the same parameters. Sometimes better to use POSIX commands and parameters so you will be able to use it the same at other variants.