Galarnyk & Stampfli, LLC.

Purdue University, West Lafayette, IN 47906

Installation and Setup of Ubuntu LTS 12.04 On a Partitioned Hard Drive

Collaborators

Noah Stampfli and Kalena Galarnyk

Table of Contents

Installation and Setup of Ubuntu LTS 12.04 on a Partitioned Hard Drive

What to Expect

By the end of this document, you can expect to have the knowledge and tools to install, setup, and administer Ubuntu (Linux operating system) on a partition of your hard drive, separate from your current operating system. This document aims to guide the technical oriented user through the process of partitioning their hard drive, setting up this Linux operating system, and informing the user of common and useful tools once the system is installed.

After following the Ubuntu installation instructions, there is a brief tutorial on command line basics to help you get started harnessing the power of your new Linux system. Also, several packages of free software are recommended for new Linux users, providing an idea of how beneficial it can be to install free and open source software (FOSS) on your new Linux system.

Why Partition Your Disk?

Setting up Ubuntu on a separate partition of your hard drive allows you to try out a new and (possibly much faster) operating system without completely committing to it. That being said, Ubuntu is a free operating system that can be run on most personal computers and laptops (except Apple hardware). Most of the current documentation available does not explicitly help the user install the system on a separate partition, and we intend to remedy this unfortunate situation.

What You Will Need

- A computer with **Windows 7** or **Windows 8** installed
- **DVD drive** and **DVD-RW/DVD+RW**
- **Internet Connectivity**
- **Mouse** and **Keyboard**
- For 32-bit Installations a minimum of:
  - **1 GB** (gigabytes) of **RAM**
  - **20.4 GB** (gigabyte) **hard drive**
- For 64-bit Installations a minimum of:
  - **2 GB** (gigabytes) of **RAM**
  - **24.4 GB** (gigabyte) **hard drive**
- A willingness to learn and grow in computing!

Getting Started . . .

Now that you know the purpose of these instructions and some benefits of choosing to install Ubuntu on a separate partition alongside Windows, you may consider proceeding with the following steps.

It is recommended that you skim through the steps and understand them BEFORE attempting to implement them on your own machine.

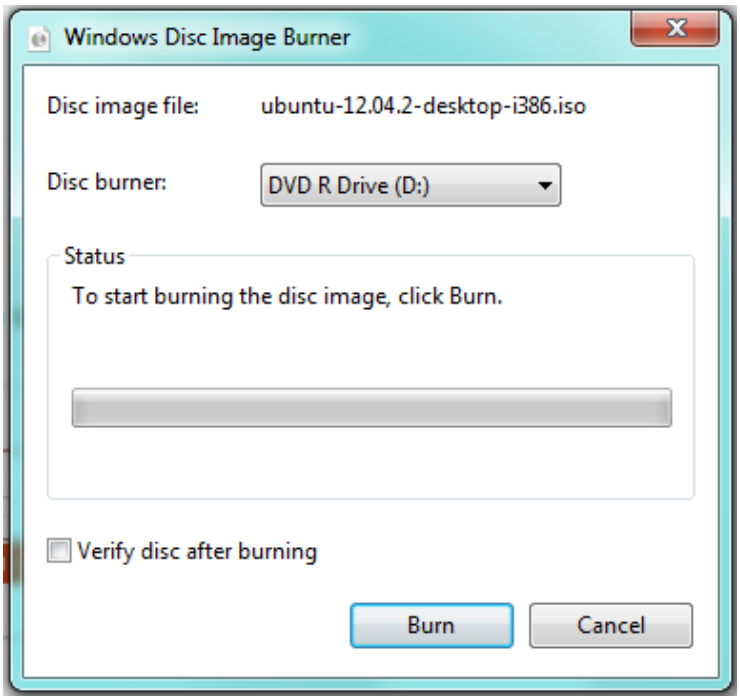
Go to the web address **http://www.ubuntu.com/download/desktop**.

Choose your "flavour": either **32 bit** or **64 bit** depending on your hardware.

Click **Get Ubuntu 12.04 LTS**.

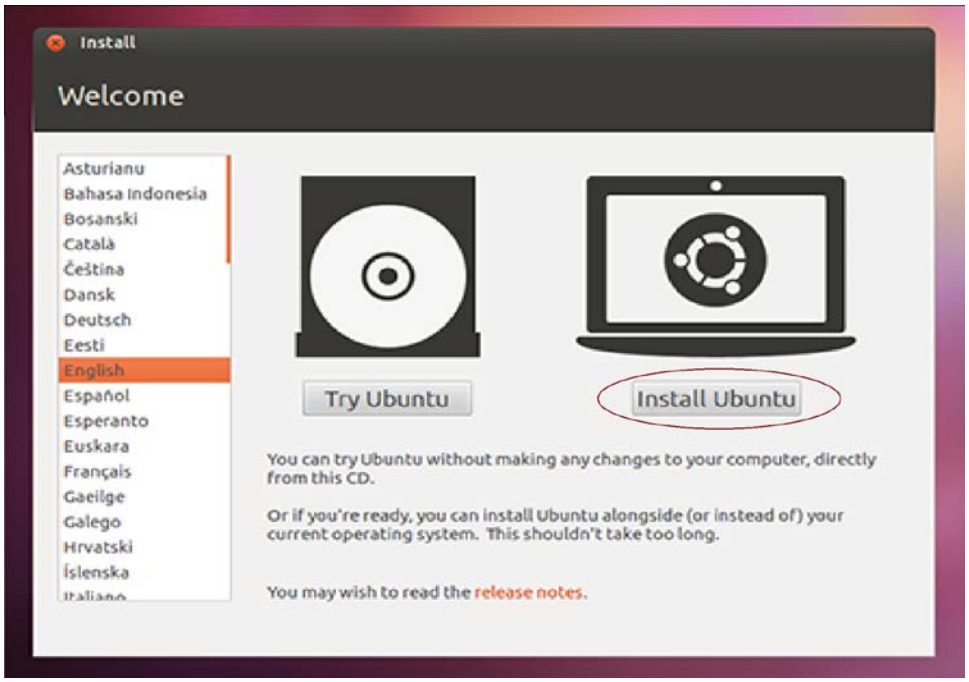
When asked to save or open the download file, **open** the “iso” file.

When the download is finished, stick a blank DVD disk into your drive and click **burn**.



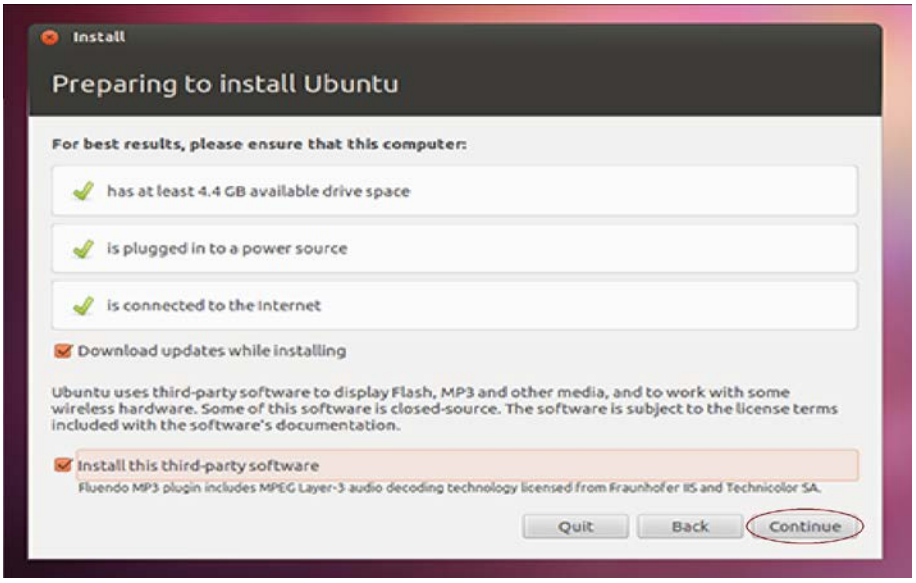
Once the DVD is burned to the disk, boot your computer from disk.

When prompted, select a language and click **Install Ubuntu**.

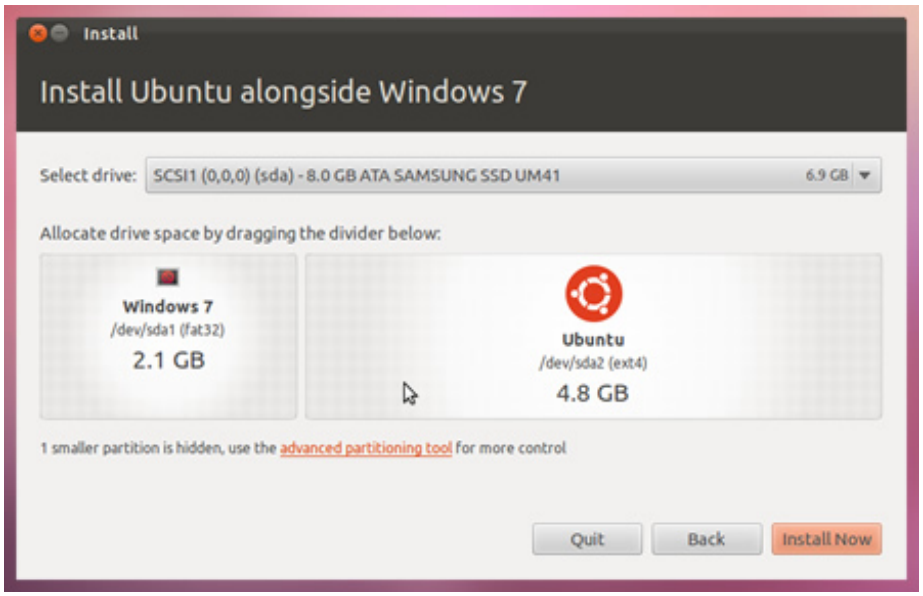


To get the most up-to-date version, ensure that **Download updates while installing** is checked.

If you wish to install the Ubuntu suggested MP3 plugin for MPEGs, check **Install this third-party software**. You will install some software for yourself later.



Follow the Ubuntu Install Manager to complete the installation, and refer to our installation video if you have any questions.

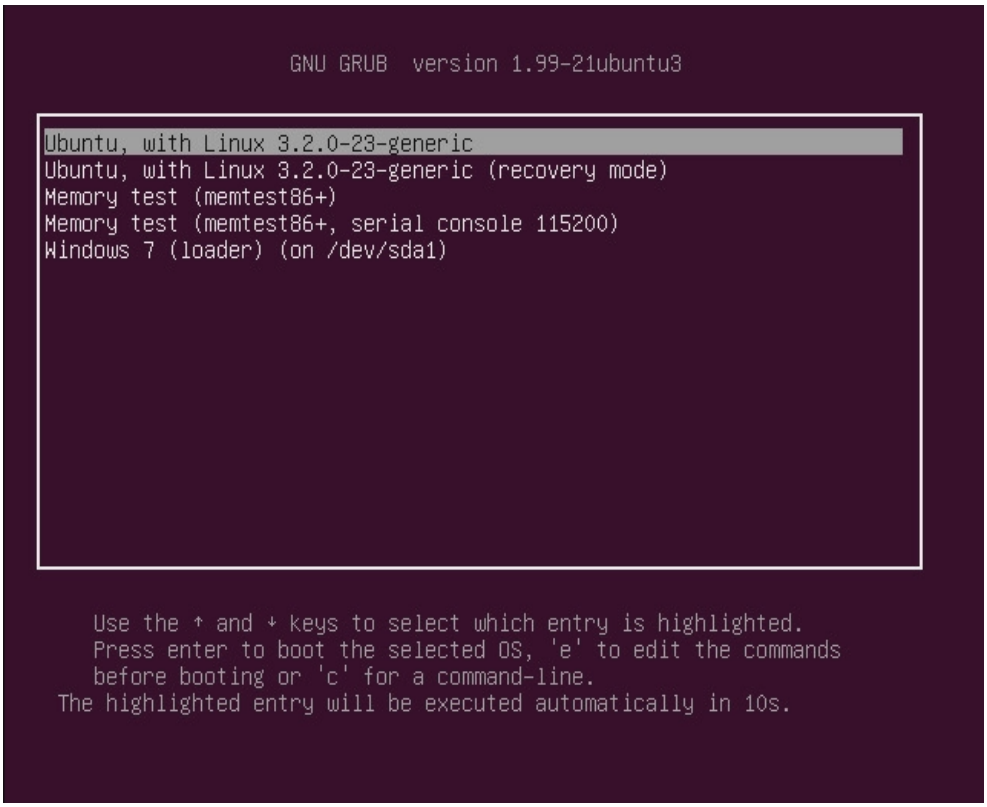


*__Note__ that this screen will appear in the install manager, where you may select the amount of memory you wish to allocate for each drive by **sliding** the divider between the drive spaces.

Once the installation is completed, you will be asked to **restart.**

The system will reboot and load the grub boot loader menu.

Select **Ubuntu, with Linux 3.5.0-23-generic** with arrow keys and press **Enter**.



What Next?

Now that Ubuntu is installed and running, you may first want to familiarize yourself with some useful tools to use with your new operating system. Two of the most alluring aspects of Linux are learning to harness the power of the command line, and installing FOSS (Free and Open Source Software). The availability of good, free software on Linux machines for consumer use is mind-boggling.

Command Line Essentials

The command line is one of the most underutilized utilities in Windows. In Linux, it is important to realize the robustness and power of using such a utility. By embracing the command line, a user can recognize the full potential of their computer, customize, and optimize their system with their own preferences. With a bit of learning, the command line is an essential for any budding Linux user. This is by no means a comprehensive guide; however, these basic commands should get you eased into the command line terminal.

To access the default command line interpreter for Ubuntu, click the Ubuntu logo in the task bar and search for 'terminal'.

man *<command>* - man is one of the most important commands to become accustomed with as a first time Linux user. The man command opens up the manual for any command recognized by the system. To open a manual, type man <command> in a terminal window. To exit a manual, press '*Q*'

> *e.g.* man less - *displays the manual for the less command, a command used to view the contents of a text file. In the manual, you will find arguments, or modifiers to the less command, examples of the less command, and descriptions of the less command's functions. Press Q to exit.*

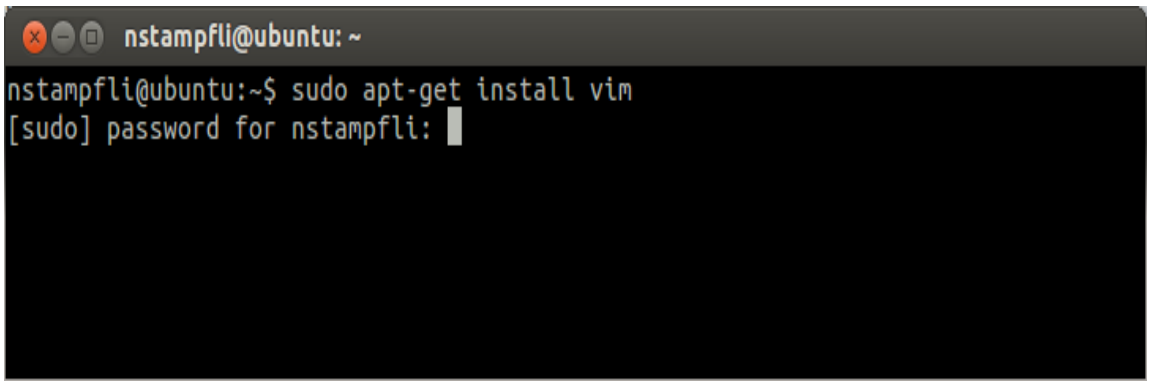
sudo *<command>* - sudo (superuser do) is an extremely important command in Linux. Ubuntu does not allow users to access the root account directly, so to execute a command that requires administrative privileges, prepend your command with sudo. Use sudo with caution, as many commands require administrative privilege for a reason. If you do not know what a command does, do not issue it. Sudo will for the root password. As the password is typed, for security neither the characters nor asterisks are displayed. Do not be alarmed, just enter the password and press **Enter**.

> *e.g.* sudo apt-get install <program name> - *utilizes sudo to allow an account to install a specified program from Ubuntu's 'Aptitude' repository. Apt-get will be discussed in more detail* next.

apt-get - apt-get is a small program in Ubuntu called a package manager. A package manager in Linux is a program that is linked to a specific repository of program packages online. As mentioned before, Aptitude is Ubuntu's repository. Package managers on other platforms include yum (on Fedora/Redhat) and yast (on OpenSUSE). Apt-get can be used to install, remove, and update programs on your computer.

> To install a program: sudo apt-get **install** <program name>
> To remove a program: sudo apt-get **remove** <program name>
> To update a program: sudo apt-get **update** <program name>

> *e.g.* sudo apt-get install vim - *uses the apt-get package manager to search through the Aptitude repository for a small command line text editor called vim (Vi IMproved) and install it to your computer. It is highly recommended that you install a command line text editor to edit any configuration files to further customize your machine in the future. Alternatives include nano and emacs.*

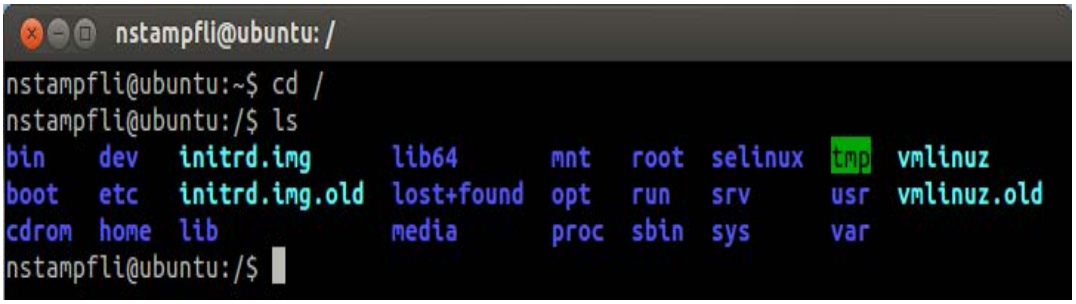


Basic File Management Commands

By the end of this section, you will have the knowledge to navigate the Linux file system, create, edit, move, rename, and remove files.

ls - ls is used to list all the files in the working directory a.k.a. the folder you are currently in. Two useful arguments are ls -l and ls -a. ls -l lists the file permissions of all files in the directory in addition to the file names. ls -a lists all files in the directory, including any hidden files. Arguments can be combined under a single flag and are applied in the order in which they are listed, though often order does not matter. (e.g. ls -la, or ls -al). Furthermore, you can use ls <directory path> to list the files in a directory other than the working directory.

e.g. cd / ; ls *- changes to the root directory and displays its contents (See Below)*



e.g. ls -a /etc *- displays all files, including hidden files in the /etc directory*

pwd - pwd (print working directory) is used to display the full path of the current directory.

e.g. in the /home/username directory, type pwd. *Terminal outputs '/home/username'*

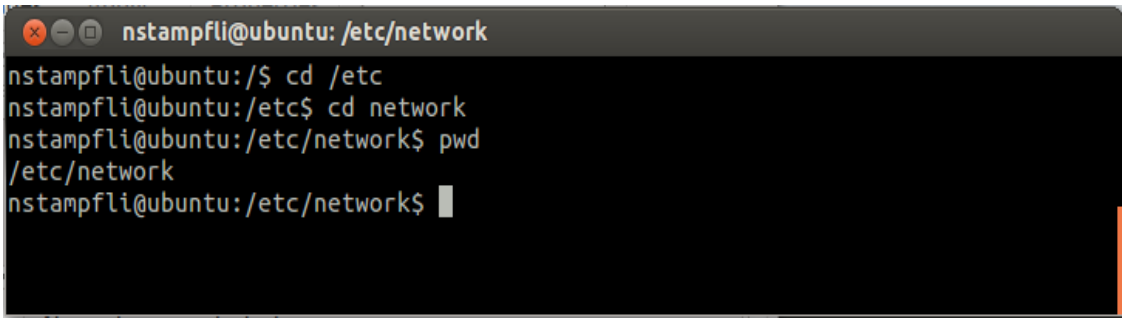
cd *<directory name / directory path>* - cd (change directory) is used to change the working directory, or switch the folder you are currently in.

Note: there are 4 special directories**:**

/	root directory
~	current user's home directory
.	working directory
..	parent of the working directory (up one folder)

Unlike windows, any directory path prepended with a **/** will look in the root directory. If no / is specified, cd will look for the specified directory or the path of directories within the working directory.

> *e.g.* cd /etc *- this command takes you to the etc directory, located in the root directory. Now, from the etc directory, type* cd network*. This command takes you to the network directory , located in the etc directory. Type* pwd *to verify this.*



mkdir <directory name> - mkdir (make directory) makes a directory in the working directory with the specified directory name.

> *e.g. in ~, type* mkdir TestDocuments*. This creates a directory in ~ with the name 'TestDocuments'. Type* ls *to view the contents of ~ and TestDocuments will be among the system output.*

rmdir <directory name> - rmdir (remove directory) removes the specified directory from the working directory and all its contents.

> *e.g. in ~, type* rmdir TestDocuments*. This removes the TestDocuments directory AND recursively removes all the contents of the TestDocuments directory from the current user's home directory. Type* ls *to verify that TestDocuments no longer exists.*

touch <file name> - touch does one of two things. If the file specified already exists, it updates the timestamp of that file. Secondly, if the file does not exist, a blank file of that name is created.

> *e.g.* touch File1 *- this command creates a new file in the working directory called File1. If File1 exists, rather than overwriting it,* touch *will update the timestamp on the file. Use* ls *to confirm its existence.*

rm <file name> - rm removes the specified file from the working directory. To remove a folder or directory using rm, you must use the -R argument (rm -R) which recursively removes all files contained in the directory. **This does not ask for user confirmation so use with caution.**

e.g. rm File1 *- removes File1 from your working directory.*

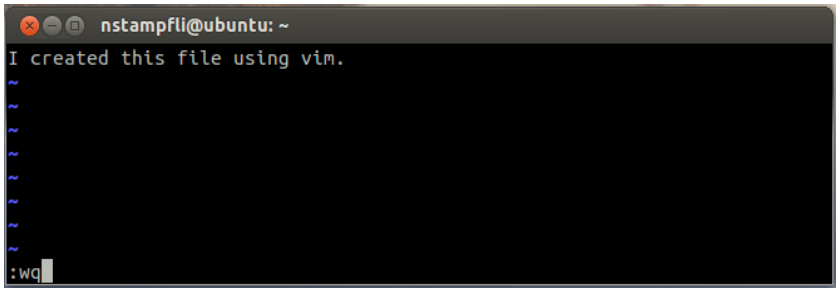
mv <file name> <destination/newname> - mv does one of two things to the specified file. If the destination exists, the file is moved to the destination. If the destination does not exist, the specified file is renamed to the specified newname.

e.g. mv testfile .. *- this moves a file named testfile to the parent of the working directory. Also try* cd ..; mv testfile testfile2 *to change to the directory that testfile was moved to, and rename testfile to testfile 2. Use* ls *to ensure that testfile was renamed to testfile.*

vim <file name> - if vim is installed (**see apt-get**), the vim command will allow you to open and edit a file, or create a new file with the specified file name. Vim only uses the keyboard navigation and can be intimidating for first time users. However, it is a very robust program and invaluable when editing configuration files. Many tutorials are available online.

- Vim has two basic modes, *insert mode* (insert or edit text) and *command mode* (issue commands to the program which is the default mode). To enter insert mode or command mode, press **'i'** or **esc respectively**.
- To exit vim, in command mode type either **:wq** to save and quit, or **:q!** to force quit without saving.

e.g. vim testfile2 *- opens the file testfile2 that we created previously. If testfile2 does not exist in your working directory it will be created.*



As stated before, these commands are by no means comprehensive, but they should get you started using the command line and being productive in Linux. Remember, if your journey on the command line takes you somewhere that you do not understand, utilize the man command and online resources to gain better understanding of the Linux file structure and contents.

Recommended (free) Software

The software packages below are a taste of some of the vast quantities of free and open-source software (FOSS) available on Ubuntu Linux in addition to other Linux platforms. By the end of this section, you will have an idea of what is available to you and the ability to install several recommended software packages.

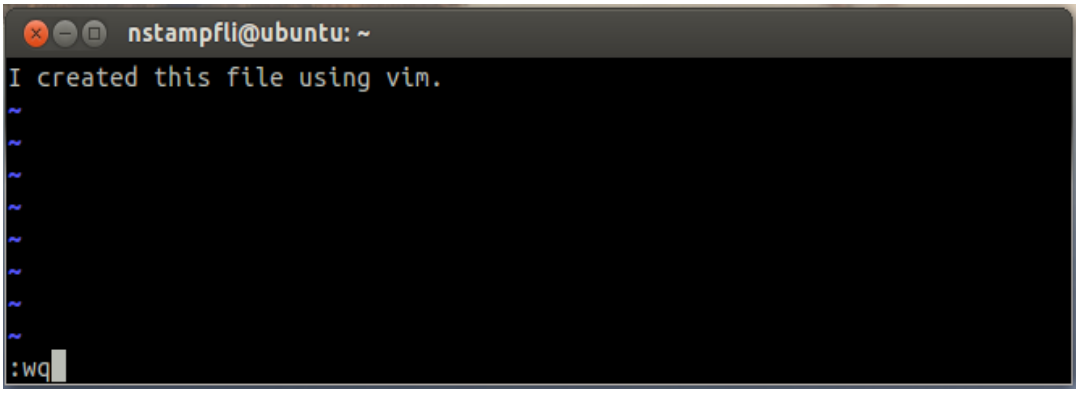
In order to install the below programs, open Terminal and type:

sudo apt-get install <program name>

VIM (Vi IMproved)

Vim is a powerful command line text editor that is used mainly to write program code, script, or edit Ubuntu configuration files. If you are a technical user, and wish to be able to truly take control of your system and make it your own, you should familiarize yourself with a command line text editor. Vim in particular has a fairly high learning curve if you wish to utilize all of its functionality. To get started using Vim, see the vim command in Command Line Basics

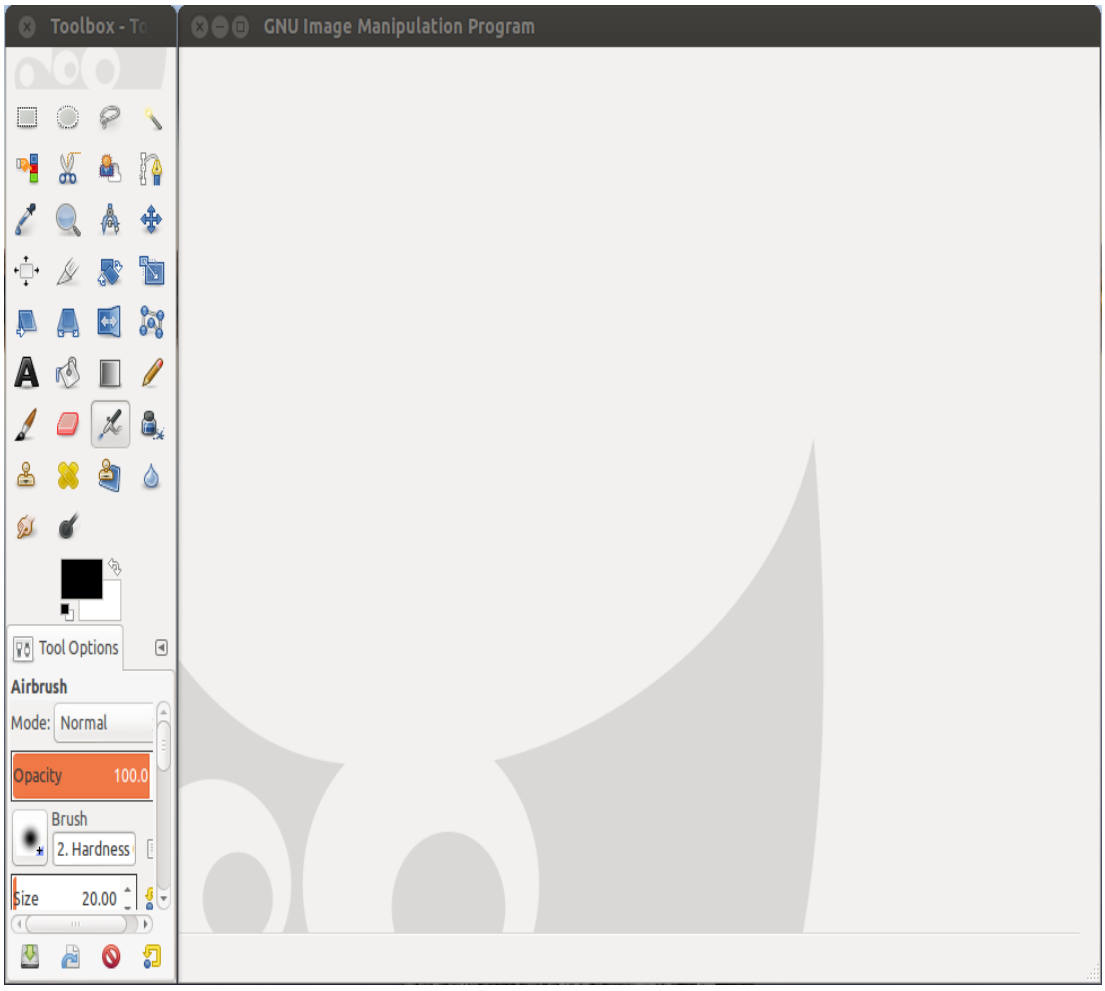
To Install: sudo apt-get install vim (see Command Line Basics)



GIMP (GNU Image Manipulation Program)

GIMP is an open-source raster-based image manipulation program similar to Adobe Photoshop. GIMP is mainly used for editing photos and creating raster (pixel-grid) images. GIMP is a powerful tool that comes with a variety of image manipulation tools and filters to assist you in bringing your imagination to life.

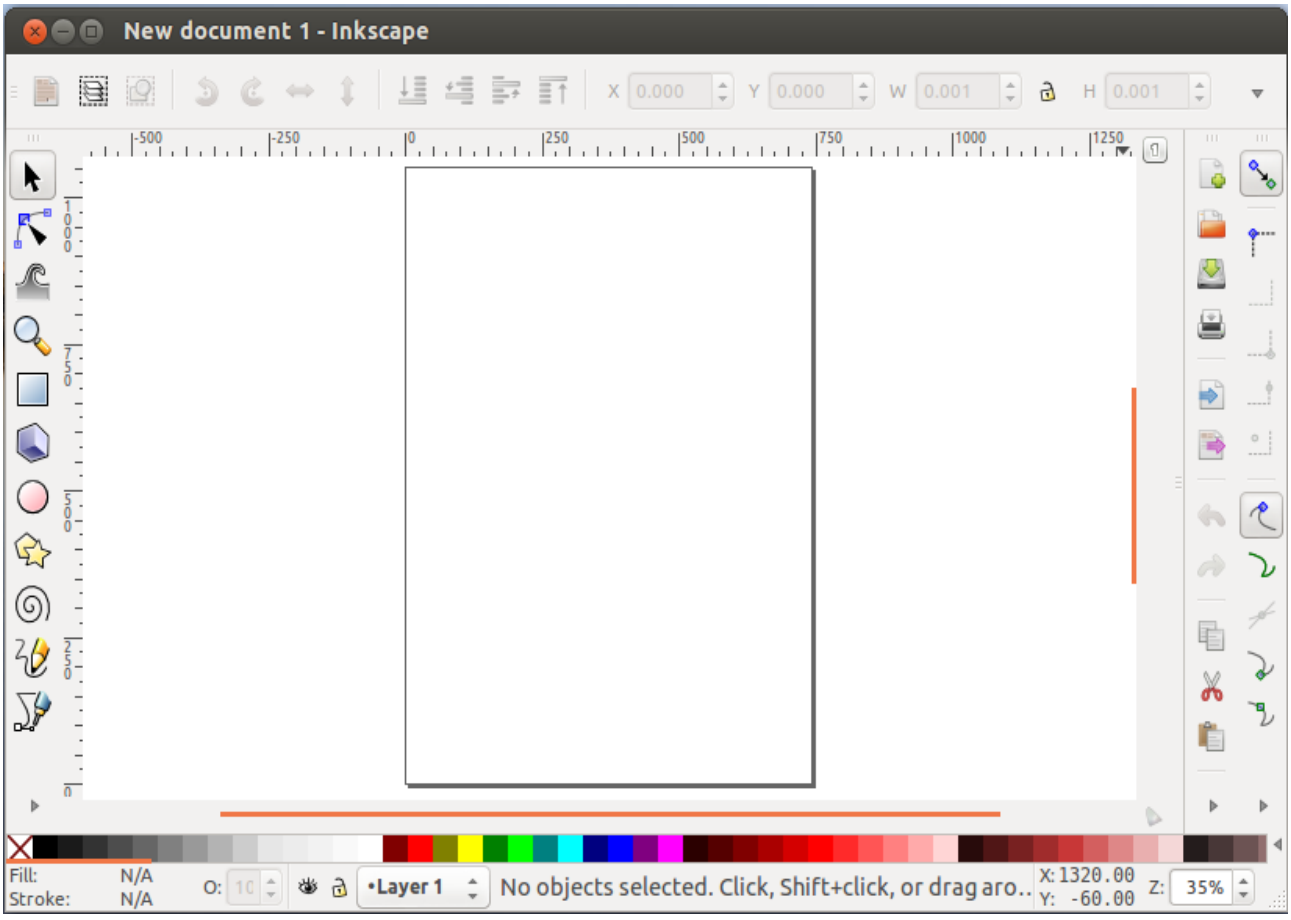
To Install: sudo apt-get install inkscape (see Command Line Basics)



Inkscape

Inkscape is a vector-based image manipulation tool similar to Adobe Illustrator. Vector images are made up of lines that can be calculated mathematically. This means that any images drawn in a vector-based program are infinitely scalable without loss of image quality. For this reason, vector-based imaging is often used to create logos and other type of images that need to be scaled to a variety of sizes. Vector imaging is clear and precise, making a professional looking image without requiring as much artistic skill as raster imaging.

To Install: sudo apt-get install inkscape (see Command Line Basics)



LibreOffice

LibreOffice is the leading free and open-source office suite. For Windows users switching to Linux, the LibreOffice equivalents are listed below. It should be noted that the LibreOffice suite offers the basic functionality of the Microsoft Office suite. Some advanced functions are left out or located in other places. LibreOffice allows for all the functionality that *most* users utilize in Microsoft Office.

To Install: sudo apt-get install libreoffice (see Command Line Basics)

LibreOffice Writer - Similar to Microsoft Word. Writer is a word processor similar to Microsoft Word allowing for an array of formatting and documentation options for writing documents.

LibreOffice Calc - Similar to Microsoft Excel. Calc is used to create and edit of spreadsheets.

LibreOffice Impress - Similar to Microsoft PowerPoint. Impress is used to create and edit of slideshow presentations.

LibreOffice Draw - Similar to Microsoft Visio. Draw is used to design a variety of diagrams.

LibreOffice Base - Similar to Microsoft Access. Used to create, edit, and manage databases and database programs.

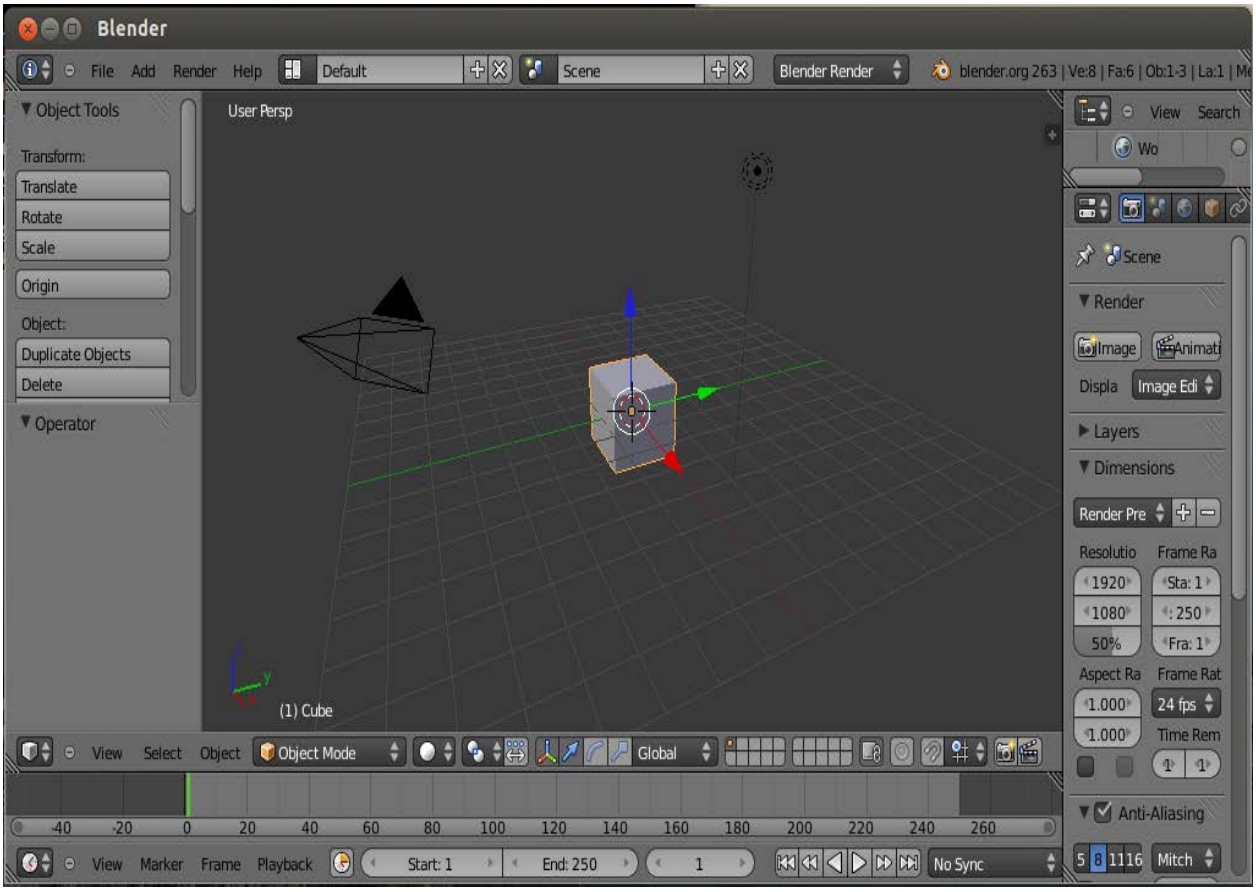
LibreOffice Math - Exclusive to LibreOffice. Math is used to create formulas that can be used in embedded in other LibreOffice Documents.

Blender

Blender is a fairly powerful computer graphics program that allows a user to create 3d models, render images based on 3d models, texture models, rig models, and animate scenes. Blender has a high learning curve but is powerful and can be used to create stunning models and animated films.

The Blender Foundation has a variety of short films online that were created and edited with this amazing program. Sintel, an award-winning short film was created using Blender to model, animate, texture and render the film.

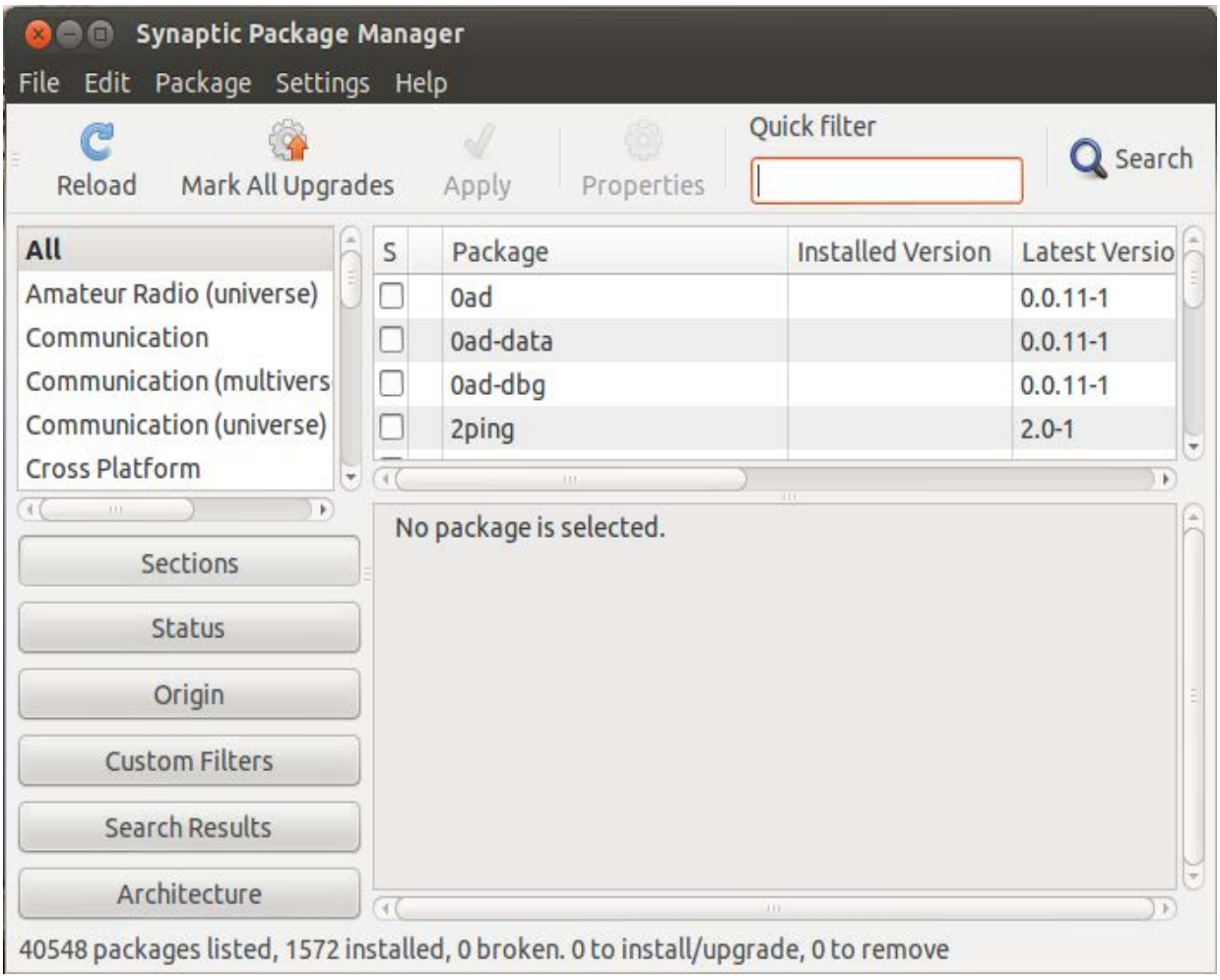
To Install: sudo apt-get install blender (see Command Line Basics)



Synaptic Package Manager

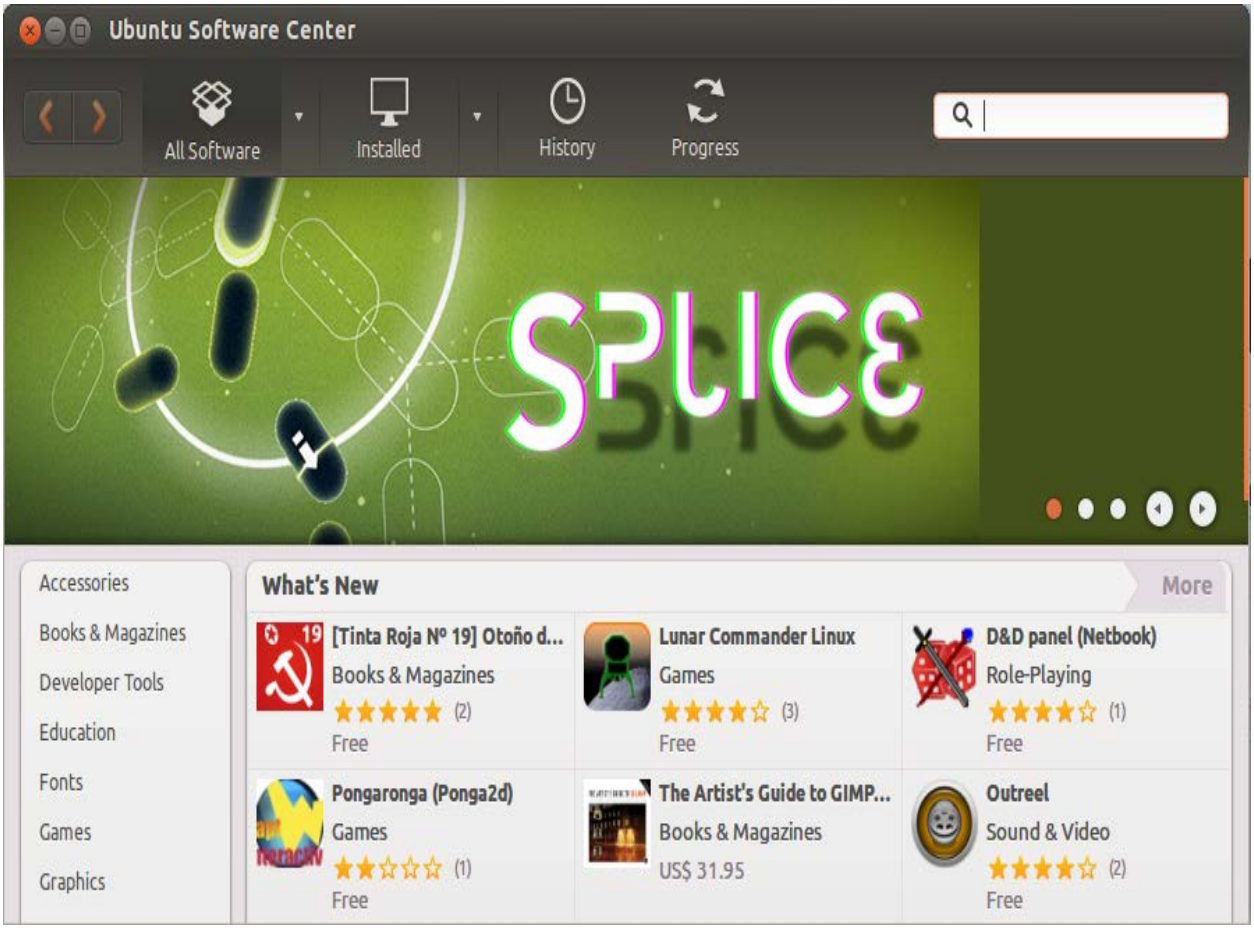
Synaptic Package Manager is a tool that allows you to install new software packages to your computer. With Synaptic Package Manager, you can browse through or search through a number of software packages that are available to install onto your system. It may be easier to find the packages you want using Synaptic and it may show you packages to install along with that package. At times it is a convenient alternative to installing everything via terminal. For those users less inclined to use the command line, Synaptic can be a good alternative for installing programs and software packages.

To Install: sudo apt-get install synaptic (see Command Line Basics)



Ubuntu Software Center

Another package manager worth mention, which was conveniently packaged with your operating system and installed during the initial operating system installation, is the



Concluding Remarks

Here at Galarnyk & Stampfli, LLC we aim to lower the barriers for users to experiment with alternate technologies that may enhance their computing experience and our combined educational background enables us to accomplish this goal. By guiding the user through this installation of Ubuntu on a partitioned disk, we hope that the user will find the transition or experimentation with a new operating system enjoyable.

If you have any questions or comments regarding this instructional document please feel free to email kgalarny@purdue.edu or noahstampfli@purdue.edu.

References

Download Ubuntu for your desktop. (2013). Retrieved March 21, 2013, from Ubuntu: http://www.ubuntu.com/download/desktop