Installing Ubuntu 8.04 for use with ESP-r
8 May 2009
Jon W. Hand, ESRU, Glasgow, Scotland

Introduction

These notes apply to Ubuntu version 8.04. There are different disk layouts discussed as well as two sets of packages needed depending on whether software development is going to be done on the machine.

If you are installing Ubuntu 8.04 in a virtual machine or re-using an existing virtual machine for environments such as VirtualBox or Parallels the look at the last sections of the document.

Before you start

Have the following information available before you start:

a) the disk space available and RAM - Ubuntu can take as little as 4GB of space and will run (a bit slowly) with about 400MB of RAM. If you can give it more memory then this will probably increase the speed of your assessments and results recovery tasks. 8GB is a ok for general use in a virtual machine or computer with limited disk space. Some installs will need more space e.g. for larger models, testing purposes, or to support multiple users.

b) the layout of the disk(s) and partition(s) on the machine - you may be wanting to dual-boot or to use a virtual drive so locate where space is available. If you are dual booting with Windows, it tends to work better if you install Linux after windows.

c) the IP address, machine name/domain, router etc.

Install sources

Ubuntu is available either as a live CD/DVD with the option to install or as an install direct from CD/DVD. There also tends to be an alternative install CD that supports a text based install which sometimes works on machines with limited memory or with problematic monitors.

The live variant will load into memory and after booting offers an 'Install' button. After clicking on the Install button the process is identical for both. If you have trouble with the Live or graphic mode install you should also try the alternative install CD which can be run in text mode. If you are installing within a virtual machine and have digested the last sections this is where you will re-join the install process.

When the Install window appears, confirm your language, location, time and keyboard. Typically Ubuntu detects most computer hardware. If there are issues with hardware you may have to go to an earlier release of Ubuntu or, if possible find a slightly older PC.

If the machine is to be dual book and the disk has unpartitioned space then select the option 'largest contiguous free space' in the Ubuntu installer. You also have the option in recent versions of Ubuntu to use the whole disk or to manually edit the partition table. The latter is an option for experts.

If there is no free space Ubuntu will offer to resize the current Windows partition to free up space. Some users might prefer to use something like PartitionMagic on the Windows side first rather than the Ubuntu tool. It might also be useful to defragment the disk prior to a resize.

In guided partitioning a "/' partition will be created as well as a Linux swap partition. You will not be asked for any further information. Tip: if you have more memory to put in your machine, do this before the install so that the swap space is appropriate.

If you do not want to use all of the free space then select the manual method. The partition table will be shown. Select the one marked as free space and create at least two partitions: a root partition and a linux swap partition. Some users prefer to have a separate /home partition.

If you are installing on a Windows PC Ubuntu will offer to import account information and settings from your accounts. This is optional.

You will be asked for an initial account/user. Consider this carefully as there is no 'root' user on a Ubuntu machine. The initial user account is given additional privileges via the 'sudo' command to temporarily acquire the permissions needed for administrative tasks. Remember the initial account name and password! You can use the first user as the owner of the simulation software and then create a second account for normal simulation use.

You will be asked the name of the machine. On a dual-boot PC you will probably use the same name as the other operating system. The name requested is the initial portion of the machine_name.domain internet name.

At the 'ready to install' review the data and take a note of the partitions that are going to be formatted. If in doubt use the 'back' arrow. The installer will first format the disk partitions. It will then install the core of the operating system (the number of packages will depend on whether you are using a CD or DVD).

For the partition manager of Ubuntu, you can use the entire new virtual disk and allow a guided partitioning. The initial phase of the Ubuntu install proceeds as usual. It will load to memory and then present an option to install on the desktop.

If your computer needs to use a proxy server to reach the internet then set of these details first.

Setup the software sources via the administrative tool.

Next select system -> administration -> update manager and this will likely have quite a few packages to update (allow it to do this).

Software needed to work with ESP-r

The base install of Ubuntu has much of what you need. One of the features of Ubuntu is the general ease of installing new software packages. Lets begin with the minimal requirements to run either the X11 or GTK version of ESP-r.

The only required packages tend to be libg2c and ImageMagick. You can search for them in the package manager and once it is in place you should be able to use the standard self-extracting installer for ESP-r on Linux.

There are a few additional packages which provide helper applications that ESP-r may wish to access. Unless you tell it differently, ESP-r will invoke the 'nedit' text editor and so it is worth installing this package.

ESP-r can also export its wire-frame images into the Xfig format as well as VRML. Xfig and transfig work together to allow the creation of vector graphics which can be useful in reports.

Software needed to develop ESP-r

If you are going to compile ESP-r you are going to need a number of tools and if you want to develop ESP-r there are even more tools that are required via the package manager. The package manager will list current and available packages in various formats. You can also search for specific packages by name e.g. ddd (a debugger) and groff (to convert some ESP-r documents into PS files).

Development: build-essential, console-tools-dev, expect-dev, g++-4.1, gcc-4.1, gfortran-4.1, gdb,gettext, libtool, linux-headers, linux-source, subversion, subversion-tools, valgrind, libxml2, libxml2-dev, libxml2-utils

Gnome desktop: dia-gnome (a useful helper application)

Graphics, netpbm (convert between graphics formats)

Libraries: libx11-6-dbg, libx11-dev,

Miscellaneous - Graphical: gsfonts-x11

Networking: gfrtp-gtk, openssh-server, ssh

Shells: csh, zsh

Word processing: enscript, psutils

For GTK development selecting the gtk+2.0 package will identify most of the dependencies you are likely to need.

Setting up your account and folders

There are several strategies for setting up user accounts on a Ubuntu computer, whether or not it is within a virtual environment. The first strategy is to remember that there is a *standard version* of ESP-r and *your version* of ESP-r. The community puts in a lot of effort to make the standard version as robust as possible for production work.

There are a number of arbitrary conventions in the ESP-r community, one of which is the use of a custom Install script instead of the usual automake based system used in many Linux packages. Another convention is that traditional users of ESP-r tend to expect to find the ESP-r distribution in /usr/esru or in /home/esru with a link from /usr/esru to /home/esru. Other users prefer to place the ESP-r distribution in /opt/esru. ESRU have not tested the option of installing into /usr/local.

People who need to keep multiple versions of ESP-r on their computers for testing and development tend to install ESP-r into their home folder e.g. esru_test. It is a good idea to install the standard version of ESP-r to the default location e.g. /usr/esru so you have a benchmark version. If you have a subversion branch and/or just want to make your own version of ESP-r, for example to support more complex models, you should place that distribution of ESP-r in a separate location. This will allow you to test a custom version against the standard distribution. The Install script supports this via command line directives.

One manual step is needed after a custom install - you need to copy the 'esprc' preferences file that was generated during the compile process into you home folder as .esprc ESP-r will scan this file in order to determine where standard databases and example models are held on the computer.

The following steps will create a standard distribution in the traditional location. Note the commands assume that you have permission to create the folder */usr/esru* and that your computer will be using the version 4 of the GCC compilers. If you are unable to create the folder you may need to alter the command to **sudo mkdir /usr/esru**

To find out about the compiler give the following commands:
which gcc
which gfortran
which g77
gcc --version
gfortran --version
g77 --version

Probably gcc will be located in /usr/bin/gcc. There may actually be several versions of gcc on your machine. The gcc --version command tells you. If the version is 4.0 or greater you need to include an additional directive **--gcc4** in the Install script below.

Create a folder for the standard distribution via:
mkdir /usr/esru
Create a folder structure for Source code in your home folder, checkout the current development_branch from the svn server and use the Install script to build the

distribution in the folder /usr/esru with the GCC version 4 compilers:

cd
mkdir Src
mkdir Src/cvsdude
cd Src/cvsdude
svn checkout https://esp-r.net/espr/esp-r/branches/development_branch
cd development_branch/src
./Install -d /usr/esru --gcc4

(Say yes to XML if you have that library, say yes to databases and yes to example models and say yes to debug. Go and have a cup of coffee).

Lets say you have your own source code branch. The process is similar:

cd
mkdir esru_my_branch
cd Src/cvsdude
svn checkout https://esp-r.net/espr/esp-r/branches/my_branch
cd my_branch/src
./Install -d /home/my_login_name/esru_my_branch --gcc4

Substitute you home folder name and your branch name as required. Now there are two versions of ESP-r on your machine. To get the version that you want you will have to manipulate the PATH environment variable to include the folder there the executables are located. The PATH environment variable is usually defined in the ./profile file in your home folder. You might find a line which includes
PATH="$HOME/bin:$PATH"

You could use a text editor to alter this line: **PATH="$HOME/bin:/usr/esru/esp-r/bin:$PATH"**

You could use a text editor to alter this line:
PATH="$HOME/bin:$HOME/esru_my_branch/esp-r/bin:$PATH"

Or, you could create a bin folder in your home folder and place it in links to the relevant executables of ESP-r. Examples of such scripts can be found in the source distribution bin folder.

If you want to be able to use both the X11 and GTK versions of ESP-r you are going to have to use the Install script twice and place the X11 version in one folder structure and the GTK version in another. Remember to issue a **make clean** command in the src folder prior to invoking the installer script the second time.

There are more documents related to managing your source code and working with the subversion source code control environment within the archive folder of the source distribution. You may also find some useful materials in the folder src/manual/OS.

The VirtualBox environment

The package VirtualBox from Sun is currently a free download and supports a number of combinations of host computer and guest operating system. This allows you to get Ubuntu running on a variety of computers. There are other packages from vendors such as VMWare and Parallels for OSX and the procedures will be similar.

Download the installer for VirtualBox2.2 (check site via google search). If this is for a machine used by several people you might choose to do it as Administrator.

You can either make use of an existing virtual disk (e.g. from another computer) or install the OS from scratch. Using an existing virtual disk tends to require only that you copy the virtual disk file to an appropriate location and then use the disk manager options within the virtual environment to identify the virtual disk file. There will be minimal adjustments once the new virtual computer is up and running,

To install from scratch you must create a virtual disk drive. With most virtual environments you are given two options - one is a fixed size and one expands to a maximum size. The latter is a good choice, it will only be as large as it needs to be. But the maximum size cannot be changed so think a bit before you select it. The default size for a new hard disk is 8GB which is sufficient for development work and a few moderate sized models. If you are going to make lots of models or are going to try and run the full ESP-r testing suite then you might want to go for 10GB.

You also need to assign a level of memory for the virtual machine. 600-800 MB is ok for normal work if you have 2GB available on your computer. If you only have 1GB available you probably want to squeeze the virtual memory down to 400-512 MB. Additional memory can speed up simulations and reduce the need for disk access during some results recovery tasks.

Check the preferences of the tool to find where the new virtual environment will be storing the files. You might want to place it in a less obscure location so it will be easier to backup. By default VirtualBox will attempt to use what they call host networking and this may require you to log into your computer a second time.

Once you have the details completed you can start up the new virtual computer, it has no operating system so there will be a wizard that will ask some questions. Place your Ubuntu CD in the drive or identify an iso file to use for the installation. Remember there are several versions of Ubuntu. Consider choosing a 'lightweight' version to limit its demands on your computer (so there is more power for compiling and running simulations). Go back to the main instructions for the rest of the process.

Using an existing Ubuntu virtual disk under VirtualBox

Install VirtualBox as above. Decide where you want to locate your virtual disks and ensure that the VirtualBox preferences are updated. Copy an existing virtual disk into that location and use the Virtual Media Manager to add this disk to the list.

To create a new machine click on the 'new' button and give the new virtual computer a name (see above) as well as defining the OS type and vendor. Allocate memory

(see above) and when it asked indicate that you wish to use an existing disk. After you click finish you can power up the new virtual computer. Usually only minimal adjustments will be necessary to get a working computer (setting network preferences and download sources and fire-walls).

Parallels on OSX

One of the frustrations of VirtualBox is limited connections to the host machine. If manual file transfers are an issue then Parallels on OSX is an option. To start from scratch have the Ubuntu CD ready, ask for a new virtual machine and it will identify typical defaults from scanning the CD and create a virtual disk for Ubuntu with minimal interactions. Parallels defaults to a disk image that will grow. Once the new virtual machine is up and running then the standard install process with an alternative Ubuntu installer can follow the standard instructions (above). The 'Live' CD assumes that there is already a working operating system (e.g. Windows) running so that is less appropriate.

Parallels does not unfortunately seem to be able to reliably read in current VDI files in use under VirtualBox.