

## **What! More Computers!**

Before we launch into this months thrilling installment of Project OSS, it is probably of value given the large number of potentially new concepts that were introduced if we have a quick recap over the key points from last month before delving any deeper.

We identified that cheap systems were not necessarily the bargains they were cracked up to be. Often these low end systems struggled under the weight of the current crop of operating systems and the software that comes with the machine means that in some cases a significant crop of additional software has to be obtained (by hook or by crook) to provide a fully functioning machine.

There are a considerable number of systems around from 5 or so years ago that may not be very current in terms of specification but which can be perfectly serviceable and offer more than adequate performance to handle the typical tasks that we would put our computers to.

Remember, from the survey Steve and I undertook at the start of this year we know that most of our computers are only used for word processing, spreadsheets, handling digital photos, some limited web work, and a few other less than demanding applications that do not by themselves demand particularly special hardware provided that the software is designed to match the hardware platform.

The biggest problem with modern PCs is that as they have become increasingly capable the software from the operating system through to the applications which run within it have become increasingly feature rich and increasingly automated, attempting to second guess (sometimes very badly) your each and every whim.

This is why despite the blistering increases in the speed of processors, the amount of RAM, the performance or hard drives that have occurred over the last few years the honest truth is that when sat back to back the machines running today are not really any different in terms of real world performance than they were 5 or even 10 years ago. They just look prettier and in a few cases might have a little bit of increased functionality.

Project OSS is about a different approach to software. Open Source software is unique, in a world where everyone tries to jealously guard each and every development and sell each and every enhancement as an upgrade OSS does the opposite baring its verysource code to the world at large allowing you the user the opportunity to see how and why a particular function operates should you be so inclined. Similarly in a world where each corporation seeks to have its own proprietary file format locking you, the user, into countless rounds of upgrades by ruthlessly changing the format with every revision of the program OSS again does the opposite by adopting open formats it allows all uses of that type of program the capability to exchange data with each other regardless of the actual application they use to word process or create spreadsheets.

Open Source Software is by no means a lesser product than closed source software. Sure it doesn't have a pretty box most of the time, sure if you get the downloaded version the manuals will be lacking. Yes there are even quite polished OSS applications where the manuals are still not particularly clever. But it is not a lesser product. There are commercial open source offerings that have either developed from the community or which regularly feed back to the community.

A good example of this Star Office/Open Office. Open Office is a community fork of Star Office which Sun Computers gave to the Open Source Community. Developments within Open Office are then fed back into subsequent releases of Star Office. You can download Open Office for the cost of the bandwidth and media. If you do you'll get a very competent Microsoft Compatible office suite with all of the key components you would expect and which has been successfully been ported from GNU/Linux to both MS Windows (most versions) and Apple OS X (with a few tweaks). What you won't get is a box, a printed manual or a help line. Go out and buy Star Office from Sun (and it really isn't that expensive compared to offerings from Microsoft) and you'll get the same product, but this time in a box, with a printed manual, some extras like clip art disks and notably access to Sun's support line. You pays your money you takes your choice, but it is essentially the same product.

There are other examples where companies have a similar relationship with the Open Source Community where both sides benefit from supporting each other and the commercial side seek to benefit not so much from the software itself as from supporting that software.

The strength of Open Source Software lies in the very fact that the source code, the heart of the program, is there for all to see and examine. Code gets reviewed by many different people, bugs can be investigated by anybody who understands code, can be fixed by anyone with the talents and time and can then be redistributed to the community.

But you don't have to be a geek to use Open Source software. You don't have to be programmer or even be remotely interested in the workings of a program to use it. You can just support it by using it, by reporting bugs if they occur and promoting it to others. You will still benefit from stable mature peer reviewed software, and above all you will have the choice of this or that of Open Office Writer or AbiWord. You will be able to choose software that suits both you and your machine.

OK, enough preaching. Let's get down to business. We promised last month to set up this machine to perform the usual functions that we all put PCs to using Open Source Software. You will recall that I put my money where my mouth is and bought (yes lashed out real folding money) an elderly laptop of reasonable specification. Interestingly, or not depending on your perspective, whilst this machine is by no means cutting edge it is only a little behind my own desktop in terms of specification, the main areas being the CPU speed and the video card. But that is by the by.

Having bought the machine we had to obtain an operating system. Although it came to me with Windows XP there was no certificate of authenticity so there was no way I could maintain that particular install even if I had wanted to.

We selected, as you will recall, an Open Source Operating System called UBUNTU which is a specific distribution of GNU/LINUX. It is quite respected for its ease of use and is very much more focused on desktop operation rather than trying to be a server Operating System. Although there is a server version available if anybody feels their home needs a server!

Anyway as you will recall we slapped our downloaded CD in the CD-Drive, booted from there and simply installed it having answered only 6 sets of questions about Username & Password, Country of Use, Time Zone, Keyboard Layout, Language, Hard Drive Partition Layout ended up with a working machine.

Did everything work? Out of the box no. Remember we promised to make this as plain as possible so that you could see and evaluate for yourselves the pros and cons of Open Source software.

The basic system worked.

The screen and video card were correctly identified and configured so nothing to change there.

The on board network adapter was correctly detected and configured. I have taken this machine into the office and once I configured for our proxy server it was able to perform a variety of network related tasks including surfing the internet.

Sound was correctly configured. A test movie clip supplied with the OS played back without fault.

USB works just fine. I have now tried it with external hard drives, external card readers, memory sticks and my digital camera and all have worked just fine.

Things I haven't tested yet.

The IR port. I have very little use for IR transfer so this facet of the machine can wait if it isn't already correctly setup.

The on board microphone. According to the specification there is a mic built into the case. Haven't got a clue if that's there or working. But if it is then as sound was OK it should be OK too. Not really a show stopper at this stage.

PC Card slot. The machine has 1 Type II slot, the sort of thing that could take a wireless adapter, or a modem or network adapter if these things weren't already built into the PC. I've got nothing to test it with at present so although the diagnostic reports suggest that this will work I cannot verify it.

Things that didn't work.

The keyboard functions for volume up/down, mute, screen brightness up/down and so forth. This is no big loss although it would be nice to maintain them. According to some reports they do work under other flavours of GNU/Linux. It could be just a case of assigning a keyboard code but it might be more complex. It's something that I might be able to solve once I sit down and understand the use of keyboard codes, but it's not a pressing matter.

The internal modem. I had taken the liberty of reading up on this machine and this type of modem at some length and to be honest I would have been more surprised if the modem had worked. Sadly practically all laptops and most desktops that have been supplied with modems installed at the factory in the last 10 years have been supplied with some form of software modem.

I don't want to get bogged down in technicalities here, but software modems are not true modems. They rely on the computer's main CPU to handle the processing required to generate tones for the phone lines and the drivers are only reliable under Windows.

I'm going to lay it out fairly plainly here. GNU/Linux does not work well with software modems. In truth in my experience some software modems are not that great under earlier versions of Windows. With GNU/Linux systems there are some work around drivers that seem to have varying degrees of success depending on the machine, driver, modem, version of GNU/Linux and the name and star sign of the operator. For me whilst these drivers were able to detect the modem and allow me to communicate with it dialing remains unreliable.

GNU/Linux is great with internal or external hardware modems but software modems are one of the biggest false economies I have had the misfortune to come across. One could argue, of course, that in these days of broadband and Wi-Fi a modem is really not essential. I disagree, I like the fall back of being able to dial and in a laptop where it could be used anywhere dial up connectivity remains a useful standby. Also the modem has a fax capability which is still a useful communication medium. Finally I have yet to go broadband at home (strange as that may seem) and still rely on a modem to retrieve my e-mail. The lack of a reliable modem on the Project OSS machine is all that prevents me from transferring all of my typical home computer uses to this machine.

So tonight we're going to look at Office Suites. Well to be precise we are going to look at what is probably the most widely used Open Source Office Suite available – Open Office.

We've chosen to look at this area first because in our original survey indicated that over 80% of machines were used for word processing and spreadsheet activities. What I hope to do tonight is offer a quick demonstration of some of the functionality of the Open Office suite.

Whilst I've been rambling on you've been looking at the UBUNTU desktop which is the default configuration. I'm not a particular fan of eye candy and tend not to waste my time dressing up my desktop. It is pretty easy to change the desktop appearance through the available themes that are pre-installed or by add in themes. There are even those that can make GNU/Linux look and feel like XP or OS X if that would make you feel at home. If there's interest we could look at this sort of thing in a wrap up session sometime.

So Open Office is an Office Application which is located in the APPLICATIONS menu under OFFICE (surprising huh!) . You can see that there are 4 components, Database, Presentation, Spreadsheet and Word processor. For now you'll have to take my word for it that all of these components are able to read and write Microsoft Office files with a very high success rate. They also read and write several other formats including Open formats based around XML I believe, although Steve can probably tell us a little more about these afterwards.

Now although I know that many of us make use of word processors and spreadsheets, and that quite a number of us make use of presentation programs and databases I don't know exactly what we use these things for. Therefore I have no idea if the little demonstration that I'm about to launch into is going to be of any interest at all. But what I aim to do is create a short report style of text import a couple of graphics that could be clip art or photos or any form of image, add a table from a spreadsheet and also a graph. The whole thing is of course totally fictional and any resemblance to persons or organisations living or dead is of course purely coincidental.

This will require two components of the office suite – Writer and Calc. I'm leaving the presentation package alone for now and I'm afraid that my knowledge of databases is rather limited so if you're really into databases you'll have to have a play for yourself.

So starting in WRITER, you'll see once it opens that the layout is very similar to WORD. There is the familiar menu structure across the top of the screen and below that a fairly recognisable set of icons. The icons have tool tips if you hover your mouse pointer over them. These are the default set of menus and toolbars, like WORD it is possible to stack other toolbars onto the screen or to customise the available buttons.