

Free and Open Source Software (FOSS), Part I

With the advent of Information and Communications Technology (ICT), it is becoming increasingly apparent that we cannot do away with software. Our lives are becoming more and more complicated each day with the introduction of one software or the other or one technology or the other. For many in developing countries, commercial software packages are not an option because they are expensive, do not come in local languages and cannot be shared. The fact however remains that software and for that matter ICTs are becoming increasingly crucial to socio-economic development. So how can we as a country utilize software and not incur too much cost? For me, the answer is Free and Open Source Software (FOSS). This article is the first of a two-part article which seeks to define FOSS, make a case for FOSS and also enumerate steps that should be taken to implement such software so that what is being advocated does not become our nemesis.

FOSS stands for 'Free and Open Source Software'. It is a movement started in 1983 by Richard Stallman and the Free Software Foundation (FSF). All the software that is run on computers has been created to work on operating systems. For Microsoft Windows users, software is 'packaged' to run on Windows and Apple's OSX or even Linux then the same rule applies. Software that is 'packaged' and installed onto your computer generally exists in what is called 'binary' form or, as an 'executable'.

Binary software is 'locked' in a sense because it is in computer language. This means that it is virtually impossible for humans to see how it operates and what it is actually doing! A good example of this is spyware or virus software. When these applications are installed onto a machine it is very hard to work out exactly what they are doing. They could be sending your banking information across the internet, or grabbing all of your important passwords. You just don't know, because they are locked.

All computer programs are first written by humans in readable language, which is called 'source code'. Microsoft owns the source code to Windows and their products, as does Apple to theirs, etc. Think about a computer virus. If we had the source code to the virus, we would know exactly what it does, and could easily stop it. The virus creators as well as Microsoft and others do not release their code to the public, which means no-one can tell what their software is actually doing.

FOSS creators however, believe that everyone has the right to know exactly what the software on their computer is doing. Not only do they release 'packaged' versions of the software but more importantly, these people make the source

code to their software freely available for everyone. This is called 'open source'. One great example of open source software is Mozilla Firefox a free web browser for Linux – itself an open source operating system, Windows and OSX, which is safe and secure. In essence you can only trust open source software, because it is freely available for everyone to read in human form and see exactly what the software does. For this reason, governments, other agencies, schools and individuals all around the world are switching to the security of Open Source Software.

In developing countries, in particular, the stability of FOSS distributions has enabled low-cost deployment of sophisticated IT infrastructure. FOSS clearly provides access to excellent primary sources for learning material for students, while the transparency of the technology has re-invigorated the technological and philosophical debate on private versus public right of access to information. Throughout, the FOSS movement has been both the technical backbone as well as the main beneficiary of the exponential growth of the Internet, especially in developing countries.

The UNCTAD E-commerce and Development Report 2003 presents a comprehensive overview of the FOSS phenomenon and explains why it is important for governments, business and civil society to seriously and openly debate and consider the benefits of a positive and pro-active approach to FOSS. The report describes FOSS as a type of software as well as a methodology, a philosophy and a movement. How can this country and other developing countries take advantage of this movement and make sure it reaps the full benefits of this.

The following paragraphs shall explore the advantages of FOSS. The first and foremost advantage of using FOSS in our computer systems is to minimize communication cost, maximize local potentials and creative capacities. It is estimated that the software industry generates yearly revenue in excess of \$300 billion. FOSS can however save billions of dollars, which can be used for other beneficial projects. Because of the huge potential ICTs present in Ghana, we should be concentrating our efforts at procuring local solutions and with FOSS. This would save this country the millions of dollars we would otherwise spend on software.

Another merit in using FOSS is the fact that it has a community of developers and supporters who have social, economic and political motivations instead of commercial. “It is fundamentally rooted in principles of sharing and, hence, promotes the flow of communication and knowledge”, (Panos Media Toolkit on ICTs – No. 5) Sharing software programs with the public can promote local capacity, creativity, localization and independence from profit makers while

reducing costs. Developing countries like Ghana would benefit from FOSS rather than purchasing licensed software.

As there is increasing pressure on countries by the WTO not to use pirated software, FOSS will definitely become an alternative. It offers an affordable and legal alternative to using illegal pirated software.

Finally, FOSS is also better at promoting local language content. English language is the main computer language. FOSS would enable better participation of non-English speaking developers. In our own case in Ghana, it is imperative that our educational reforms do not leave out our local languages so that students coming out of our schools who would become these same developers can develop software in our local languages. Because proprietary software is closed, it is difficult to change into local languages.

Part II of this article would discuss the some of the disadvantages and highlight steps that the government should take to implement FOSS.