University of Delhi
Free and Open Source Software Policy

As per the guidelines from Ministry of Electronics & Information Technology (MeitY) and Ministry of Human Resource Development (MHRD), the University of Delhi has decided to adopt free and open source software (FOSS) and FOSS-based solutions for teaching and research in the University and its colleges. It applies to all software, which includes operating system, productivity software (e.g. word processor, spreadsheet, email, Web browser), graphics and multimedia software.

1. Definition

For the purposes of this Policy, the term “Free and Open Source” or purely “Open Source” relates to the license conditions under which Open Source Software is made available, and employs the definition developed by the Open Source Initiative, available at (attached the copy of this as Annexure A) http://www.opensource.org/docs/definition.php

2. Rationale:

Adoption of Free and Open-source software is in coherence with the University’s vision of openly sharing the results of intellectual endeavors. FOSS also gives benefits through improved product quality, simplicity and performance while allowing the the User to connect to the vast global community

Another prime reason for migrating to ‘Free and open source software’ is for their reliability and stability. FOSS servers are known to function for years without requiring maintenance. Low costs, low administration requirement and not being under the control of proprietary software vendors are major qualifiers for making FOSS suitable to an academic environment. Besides there are many other reasons why the University should aggressively adopt FOSS. These include:

Promote culture of collaboration/sharing: FOSS believes in the culture and collaboration. This is particularly important for software sharing without fear of license infringement - ‘piracy’.

Security : Open Software has a better security record due to

(a) Availability of source code: This makes it easier for developers and users to discover and fix vulnerabilities, often before they are exploited. (b) Security focus: FOSS is more focused on robustness and functionality, rather than ease of use. Before features are added to any major open-source application, it is determined that they do not compromise system security. (c) Roots: FOSS systems are mostly based on the multi-user, network-ready Unix model, hence they come with a strong security and permission structure.

Open standards and vendor independence

Open standards give users, whether individuals or governments, flexibility and the freedom to change between different software packages, platforms and vendors. It is always possible to reverse-engineer and document the standard used by an application. All possible variations are plainly visible in the source code, making hiding a proprietary standard in FOSS systems impossible. FOSS projects actively work to achieve active standards compliance. The development culture, where sharing and working together with other applications are the norm, ensures this.

Reduced reliance on imports

A major incentive for developing countries to adopt FOSS systems is the enormous cost of proprietary software licenses. These funds could be better spent on other development goals.

Developing capacity for local software The FOSS developmental approach greatly facilitates not only innovation but also its dissemination. In particular, this often means that new research ideas are first implemented and available on Linux before they are available / incorporated into other platforms.

Piracy, IPR

Software piracy, a problem in almost every country around the world, becomes acute in developing countries, where lower incomes make software far more expensive. We open ourselves up to litigation and ridicule. A culture of software piracy is also detrimental to local software development.

Localization

Localization involves taking a product and making it appropriate to the target locale - FOSS allows creation of easy local/personal fixes and enhancements. These provide an opportunity to contribute to the ‘common good’

3. Policy Position and Strategy:

In the light of the above, the following policy position and strategy may be adopted:

1. The University endeavors to avoid becoming “locked into” any proprietary software platforms if possible – freedom of choice should be maintained, wherever possible
2. The University recognises that there may be some situations where proprietary software is well-justified
3. The University actively encourages the exploration of Open Source software solutions in all areas of application;
4. The University shall set up a website that provides up-to-date information on Open Source software availability, updates, developments, resources, guides, manuals, compilers etc.
5. The University may take initiatives to encourage academic community to use Open Source software as a mean of their affordability, adaptability and flexibility, to expose them to this alternative form of yet empowering open licensing
6. The University may ensure that it imposes no requirement or expectation on students in any discipline that would require them necessarily to make use of proprietary software when a cost is involved, where there are comparable Open Source packages available
7. The University may also invite suggestions/feedback to explore the inhibitions to wider deployment of Open Source software, and how to overcome any difficulties associated within the University
8. The University may conduct training programs for the faculty members on open source technology and its frame work

Annex A: The Open Source Definition

Source: http://www.opensource.org/docs/definition.php

[This material is reproduced here as permitted by the footnote on the OSI Home Page, which states "*The contents of this website are licensed under the Open Software License version 1.1.*" which states "Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license to do the following: a) to reproduce the Original Work in copies; ..."]

The indented, italicized sections below appear as annotations to the Open Source Definition (OSD) and are not a part of the OSD. A plain version of the OSD without annotations can be found at http://www.opensource.org/docs/definition_plain.php.
A printable version of this annotated page is available at http://opensource.org/docs/def_print.php.

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

Rationale: By constraining the license to require free redistribution, we eliminate the temptation to throw away many long-term gains in order to make a few short-term sales dollars. If we didn't do this, there would be lots of pressure for cooperators to defect.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost – preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate

forms such as the output of a preprocessor or translator are not allowed.
Rationale: We require access to un-obfuscated source code because you can’t evolve programs without modifying them. Since our purpose is to make evolution easy, we require that modification be made easy.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
Rationale: The mere ability to read source isn’t enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.

4. Integrity of The Author’s Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of “patch files” with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
Rationale: Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they’re being asked to support and protect their reputations.
Accordingly, an open-source license must guarantee that source be readily available, but may require that it be distributed as pristine base sources plus patches. In this way, “unofficial” changes can be made available but readily distinguished from the base source.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.
Rationale: In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process.
Some countries, including the United States, have export restrictions for certain types of software. An OSD-conformant license may warn licensees of

applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
Rationale: The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
Rationale: This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program’s being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program’s license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
Rationale: This clause forecloses yet another class of license traps.

9. The License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
Rationale: Distributors of open-source software have the right to make their own choices about their own software.
Yes, the GPL is conformant with this requirement. Software linked with GPLed libraries only inherits the GPL if it forms a single work, not any software with which they are merely distributed.

10. License must be technology-neutral

No provision of the license may be predicated on any individual technology or style of interface.
Rationale: This provision is aimed specifically aimed at licenses which require an explicit gesture of assent in order to establish a contract between licensor and licensee. Provisions mandating so-called “click-wrap” may conflict with important methods of software distribution such as FTP download, CD-ROM anthologies, and web mirroring; such provisions may also hinder code re-use. Conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues.

Origins: Bruce Perens wrote the first draft of this document as “The Debian Free Software Guidelines”, and refined it using the comments of the Debian developers in a month-long e-mail conference in June, 1997. He removed the Debian-specific references from the document to create the “Open Source Definition.”