LAB 1

Free and Open-Source Software

Full name: **Lâm Thế Vinh**

Student ID: **B2206022**

- Note: Screenshots need to be clear and good-looking; submissions must be in PDF format.

1. Using OSS Methods to Produce Better Products

Some reasons why adopting OSS methods can improve and accelerate a product's success:

- **Develop applications faster:** Leveraging existing OSS components and libraries can speed up development because you build on pre-existing, tested, and optimized solutions.
- **Collaborative Development:** Collaborative development enables software projects to build better software. When progress is shared, not everyone has to solve the same problems and make the same mistakes. Thus, progress can be made much faster and costs can be reduced.
- **Security and Quality of Source Code:** There are more eyeballs examining code and looking for security weaknesses before they are discovered by bad actors. On the other hand, Code published openly tends to be cleaner because It is embarrassing to show ugly, sloppy code; coding standards and styles tend to be cleaner and more consistent on community projects.

2. What OSS Products Do You Use?

5 Proprietary software and the FOSS equivalent for each:

	Proprietary software	FOSS equivalent	FOSS product's license	Comparison
1	**Microsoft Office**	**LibreOffice**	**MPL (Mozilla Public License) 2.0**	- **UI**: Microsoft Office has a polished, modern UI with a ribbon interface that many users find intuitive - **Documentation**: Microsoft Office offers extensive official documentation and support resources. LibreOffice has comprehensive community-driven documentation - **Buggines:** Microsoft Office is generally stable, with bugs quickly addressed by regular updates. LibreOffice is regularly

				updated and fixed by its active community.
2	**Windows OS**	**Linux OS**	**GNU General Public License version 2 only (GPL-2.0),**	- **UI:** Windows offers a consistent and user-friendly interface with extensive support for a wide range of applications. Ubuntu's UI (using GNOME by default) is clean and efficient & highly customizable. - **Document:** Windows has extensive official documentation and support channels. Ubuntu provides thorough community-driven documentation. but it may require a deeper technical understanding. **- Bugginess:** Windows is generally stable, but it's prone to occasional bugs, particularly after major updates. Ubuntu is also stable but may require manual intervention to resolve issues, especially with hardware compatibility or after upgrading.
3	**Microsoft Outlook**	**Thunderbird**	**MPL (Mozilla Public License) 2.0**	**- UI:** Outlook has a polished, integrated UI with a focus on productivity, and it integrates seamlessly with other Microsoft services. Thunderbird's UI is functional but feels more dated, with a focus on flexibility and customization through add-ons. - **Documentation**: Microsoft provides extensive, well-organized documentation and support. Thunderbird's documentation is community-driven, detailed, but sometimes harder to navigate, especially for troubleshooting. **- Bugginess:** Outlook is generally stable, with minor bugs that are usually quickly resolved. Thunderbird is also stable, but users occasionally encounter bugs, especially with add-ons, though issues are typically addressed by the active development community.
4	**Adobe Photoshop**	**GIMP**	**GPL (GNU General Public License) 3.0**	**- UI:** Photoshop's UI is sleek, highly customizable, and designed for professionals. GIMP's UI, while powerful,

				is often considered less intuitive and has a steeper learning curve, especially for those accustomed to Photoshop. - **Documentation:** Adobe provides professional-grade documentation and tutorials. GIMP has good community-generated documentation, but it can be inconsistent and less user-friendly compared to Photoshop's resources. - **Bugginess**: Photoshop is robust and stable, with rare critical bugs. GIMP is generally stable, but users sometimes report issues with plugins or when handling very large files, though these are usually addressed by the community.
5	**VMware Workstation**	**VirtualBox**	**GPL (GNU General Public License) v2.0**	**- UI:** VMware Workstation has a polished and user-friendly UI with advanced features for managing multiple virtual machines. VirtualBox's UI is functional but less refined, though it offers a wide range of features. - **Documentation:** VMware offers extensive official documentation and support for enterprise users. VirtualBox's documentation is comprehensive, with strong community support and third-party guides. - **Bugginess:** VMware Workstation is highly stable and optimized for professional environments. VirtualBox is stable but might encounter more bugs or performance issues, especially with less common configurations or operating systems.

3. **Contribute to OSS projects**

Notepad++:

- **The project leader:** Don Ho.
- **Project Start:** the development of Notepad++ began in 2003. The first release was in 25 November 2003.
- **License:** the GNU General Public License.
- Answer the following questions using README file
    + **What does this project do?**

- Notepad++ is a free (as in “free speech” and also as in “free beer”) source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment.

+ **Why is this project useful?**
  - Based on the powerful editing component Scintilla, Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment.
+ **How do I get started?**
  - To start with the Notepad++ You can visit the official Notepad++ website https://notepad-plus-plus.org and go to the download section. Here, the website will display the available versions, allowing you to choose one that is compatible with your computer and download it easily.
+ Where can I get more help, if I need it?
  To get more help: You can access the Online Help section on the official Notepad++ website. Here you can find sections such as: User Manual - A collaboratively built online document; Community - Where you can ask fellow users for help and suggestions; Issue Tracker - Where you can search through and make official feature requests and bug reports; UDL online documentation - Notepad++ User Defined Language online documentation.

- Answer the following questions using the project contribution guidelines
  + How to file a bug report
    Before filing a new bug report, check the Notepad++ GitHub Issues page to see if the issue has already been reported. If it’s a new issue, Open a new issue on the GitHub repository and fill out the template provided, Description: Explain the issue clearly, Steps to Reproduce: List the exact steps to recreate the bug, Screenshots or Logs: Attach any relevant screenshots or log files…
  + How to suggest a new feature
    You can create a new issue on the GitHub repository with your feature request.
  + How to set up your environment and run tests
    You can follow the steps, Clone the Notepad++ repository from GitHub, Build the project: Instructions for building Notepad++ from source are typically included in the repository’s documentation. Follow these instructions to compile the code and Run tests.
  + The types of contributions the project is looking for

The Types of Contributions the Project is Looking For: Bug Fixes, Feature Enhancements, Code Quality Improvements, Documentation, Testing

+ The roadmap or vision of the project

  Project Goals: Notepad++ aims to be a versatile and efficient text editor for Windows. It focuses on performance, feature richness, and user-friendly design.

  Future Plans: The roadmap for Notepad++ is typically outlined in its GitHub repository or discussed within the community. Major updates and feature plans might be shared through issues, pull requests, or discussions on their GitHub page.

4. Selecting a License

The [OSSWatch] tool attempts to help its users understand their own preferences about free and open-source software licenses. There are 7 choices that you need to make.

The question I need to answer when making a choice:

- Do you want to limit the results to licences that the Open Source Initiative describe as being "popular and widely used or with strong communities"?
- Do you want to include licensing conditions on reuse?
- How would you like your licence to handle the issue of jurisdiction?
- What is your attitude to the issue of patent grants in relation to your desired licence?
- What is your attitude to patent retaliation in your desired licence?
- Do you want your licence to specify enhanced attribution?
- Do you want your licence to address the 'privacy loophole'?
- Do you want your licence to include such a 'no promotion' feature?

- What license(s) match ALL requirements below (other choices are “I don’t care”)?
  - Permissive, exclusively patent grants, specify enhanced attribution

> **Summary**
>
> This is your selection:
>
> 1. Popular and widely used: No
> 2. Licence type: Permissive
> 3. Jurisdiction: Don't care
> 4.a Grants patent rights: Yes
> 4.b Patent retaliation clause: Don't care
> 5. Specifies enhanced attribution: Yes
> 6. Addresses privacy loophole: Don't care
> 7. Includes 'no promotion' feature: Don't care

  -

The licenses match ALL requirements:

Academic Free License 3.0	[6 out of 7]
Affero GNU Public License	[6 out of 7]
Apache License 2.0	[6 out of 7]
Artistic License 2.0	[6 out of 7]
Attribution Assurance Licenses	[6 out of 7]
Common Public Attribution License 1.0	[6 out of 7]
Lucent Public License Version 1.02	[6 out of 7]
Microsoft Public License	[6 out of 7]
Reciprocal Public License 1.5	[6 out of 7]
Adaptive Public License	[5 out of 7]
Boost Software License	[5 out of 7]
Common Development and Distribution License	[5 out of 7]
Common Public License 1.0	[5 out of 7]
Eclipse Public License	[5 out of 7]
Educational Community License Version 2.0	[5 out of 7]
Eiffel Forum License v2.0	[5 out of 7]
European Union Public License	[5 out of 7]
Fair License	[5 out of 7]
GNU General Public License v3.0	[5 out of 7]
GNU Library or 'Lesser' General Public License v3.0	[5 out of 7]

Historical Permission Notice and Disclaimer	[5 out of 7]
ISC License	[5 out of 7]
MIT license	[5 out of 7]
Microsoft Reciprocal License	[5 out of 7]
Mozilla Public License 1.1	[5 out of 7]
NTP License	[5 out of 7]
New and Simplified BSD licenses	[5 out of 7]
Non-Profit Open Software License 3.0	[5 out of 7]
Open Software License 3.0	[5 out of 7]
Qt Public License	[5 out of 7]
University of Illinois/NCSA Open Source License	[5 out of 7]
zlib/libpng license	[5 out of 7]

Take screenshots to show that you finish this exercise

5. Experimenting with FOSSology (optional)

The FOSSology project offers easy-to-understand options for learning the basics of how to use the tools. The easiest way is to use the project's online testing facility. To do this:

1. Point your browser to https://fossology.osuosl.org/repo/
2. Login with username=fossy and password=fossy.
3. Download the source code of the project in **exercise 3**, then upload it to FOSSology.

4. After the source code is analyzed, In the select action box, click onCopyright/Email/Urlin in the drop-down menu. You see the copyright, license, and URL information highlighted in appropriate colors.
5. Try clicking on some of the other fields in the top ribbon banner to see more useful information.

Take screenshots to show that you finish this exercise

---END---