Assignment 3

Due on November 17, 2010, at the beginning of class

Please make sure your assignment satisfies the following basic requirements:

- printed *on normal paper* and **stapled**,
- showing your name in the top right corner,
- using 12 pt serif font (like Times New Roman),
- using 2 cm margins,
- showing **the number of words after every answer.**

Assignments that fail to satisfy those requirements may be returned. The students will be asked to resubmit the assignment and will incur late penalty.

When answering the questions, make sure to cite *all sources* on which you are basing your answer. Please note, however, that citations to **assigned readings** may be necessary for full credit.

Use **APA style** and include page numbers. (E.g. "Raymond, 1998, p. 2.") For readings included in the course pack with unnumbered pages, assume that the first page included in the course-pack is p. 1, then count from there. Include an APA-style bibliography section at the end of your assignment.

Any sequence of two or more words copied from any source (readings, slides, other students notes, etc.) and not originally written by you individually **must be in quotation marks**. Failure to use quotation marks will be considered an academic offense and will be handled accordingly. *This rule applies whether or not the source is identified.* (Failure to identify the source makes the offense yet more serious, but is not a necessary component of plagiarism.) Here are some examples of proper and improper use of sources:

According to Raymond (1997, p. 5), with a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone. **[Plagiarism: verbatim copy without quotation marks.]**

According to Raymond (1997, p. 5), if you have enough beta-testers and co-developers nearly all problems will be characterized quickly. The fix will obvious to some person. **[Plagiarism: the text shows only trivial differences from the original.]**

According to Raymond (1997, p. 5), with enough beta-testers and co-developers it becomes possible to quickly characterize the problems: for some person the fix will be obvious. **[Border line: the text shows differences from the original, but it is fairly obvious that the author started with a direct quotation and then made some changes just so that it is not a verbatim copy. Multiple sentences of this kind can constitute plagiarism.]**

According to Raymond (1997), "given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone" (p. 5). **[Not plagiarism: the author uses quotes to indicate that those are Raymond's words. However, this answer does not make it clear the author understands what Raymond actually means. The author may get less than full credit.]**

Raymond (1997) argues that having a lot of people testing the code and contributing as "co-developers" allows all problems to be found, understood, and fixed quickly. **[Good]**

Part A (80 points)

Answer each of the following questions in **100 words or less**. Each question is worth 10 points.

1. What license did Linus Torvalds use for the original version of Linux? What were the key differences between this license and GPL? What lead him to switch to GPL?

2. According to Moody, what was the main difference between Linux and Minix in terms the way the way the projects were run? What was the effect of this difference?

3. Richard Stallman argues that people should use the term "GNU/Linux" instead of "Linux." What is the technical/historical basis for this argument? Why do you think Stallman believes it is worth insisting on the term "GNU/Linux"?

4. Explain the title of Torvalds's book. Why does he say "I probably would have stopped at the end of 1991"? Why did he continue?

5. Who holds the trademark for the term "Linux" and why?

6. What is Fogel's advice for choosing a license for an open source project?

7. According to Fogel, how does one "apply" a license to open source software?

8. Fogel writes: "Making important decisions in private is like spraying contributor repellant on your project." What does he mean by that and what is his advice?

Part B (20 points)

Answer the following questions in **100 words or less per question.** (Each question is worth 10 points.)

1. What is "forkability" and what is the connection between forkability and free / open source licenses? Is free / open source software always forkable?

2. What are the implications of forkability for how free / open source software projects are run?

Part C (30 points)

Pick a free / open source software project. Find that project's website, an archived mailing list, and a source repository (a version control system). If the project you picked appears to lack one of those, find another project that does. Please answer the following questions about the project, citing specific URL (or URLs) for each answer. **Please do not email project members with questions.** Each question is worth 3 points.

1. What is the name of the project and the URL of its main website?

2. What mailing lists does the project have? If there are several, what is the difference between them? How can one subscribe to the "main" mailing list?

3. What software is used to manage the mailing list(s)? Is that software itself free / open source?

4. Where are the archives for the mailing list? How many messages (roughly) are sent to the list in a typical month?

5. What kind of messages are most common? (E.g., is it mostly question-and-answer messages? Is it mostly announcements?)

6. Does the project offer a way to download source code as an archive file (e.g., .zip, .tar, .tgz)? If so, where? What is the latest version available? When was it posted?

7. Does the project offer *compiled* software for download (e.g., .exe)? If so, for what operating systems?

8. How would one get the source code with the history of revisions? What revision control software would one need for that?

9. What was the latest modification to the project's source code? Who made it? What files were affected? Did the developer provide a summary of the changes? If so, what is it? (If you do not understand what it means, just quote it.)

10. Does the project website provide the users with instructions on how to report bugs? How are they asked to do this? Do they actually use this method?

You can use **up to 100 words** to answer each question, though shorter answers would be sufficient in most cases. If some of those questions cannot be answered, explain why not and what you did to try to answer it.

Part D (20 points)

For your project in part C, download the archive file with the latest release of the project's source code. (This should be a file with an extension such as ".zip," ".tar.gz," ".tgz.") If the project you described in part B does not offer source code in this format, find some other free/open source project that does. In either case, state the URL from where you downloaded the code and the name of the file. Find software to open the archive. Examine the content of the archive. You should be able to use a simple text editor (like Notepad or TextEdit) to open individual files, though using a "code editor" such as Notepad++ or TextWrangler will give you a more authentic experience.

Answer the following questions, using **up to 100 words** per question. (Shorter answers should be sufficient in most cases.) Each question is worth 5 points.

1. What's inside the archive? Describe the files and directories that you see.

2. Is there a README file? Describe roughly what it says.

3. Is there a file that describes the installation process? If so, what does it say?

4. Does it say anywhere inside the archive under what license the content is distributed? If so, where? Do individual source code files have copyright notes or license information in them?

*Please include **an APA-style bibliography** at the end of your assignment. Assignments without a properly formatted bibliography **may lose up to 5 points**. This penalty will be subtracted from your total score and will be **in addition to** any points taken off for specific answers that fail to correctly document sources.*