Đóng góp cho phần mềm tự do

Nguyễn Thái Ngọc Duy
pclouds@gmail.com

2009-11-23 – Commit 702e152
Nguồn: github.com/pclouds/onlinux.git

Tóm tắt nội dung

Bài viết này bàn về việc đóng góp cho phần mềm tự do[1]. Bài viết sẽ đề cập sơ lược về khái niệm phần mềm tự do, những nguyên tắc đằng sau phần mềm tự do, những lý do nên sử dụng cũng như đóng góp cho phần mềm tự do, và sau cùng, cách tiếp cận/đóng góp sao cho hiệu quả và phù hợp với văn hoá phần mềm tự do.

1 Phần mềm tự do/nguồn mở là gì?

Phần mềm tự do/nguồn mở – Free/Open Source Software – là những phần mềm thoả mãn những điều kiện do tổ chức Free Software Foundation[2] hoặc Open Source Initiative[3] đưa ra. Trên nguyên tắc, phần mềm tự do và phần mềm nguồn mở là hai khái niệm khác nhau. Phần mềm nguồn mở là một khái niệm ít chặt chẽ hơn phần mềm tự do[4]. Về cơ bản, cả hai khái niệm đều bảo đảm những quyền cơ bản cho người sử dụng[5], bao gồm quyền được tự do sử dụng, phân phối và sửa đổi chương trình.

Có rất nhiều phần mềm tự do/nguồn mở trên thế giới. Nổi bật nhất có thể kể đến là Linux kernel (thành phần cơ bản của các hệ điều hành Linux như Fedora hay Ubuntu), Mozilla Firefox (trình duyệt Web), Mozilla Thunderbird (trình đọc thư), OpenOffice.org (bộ phần mềm văn phòng)...

Cách đơn giản nhất để biết một phần mềm có phải là phần mềm tự do/nguồn mở hay không là xem giấy phép sử dụng của phần mềm. Tổ chức Free Software Foundation công nhận những phần mềm sử dụng giấy phép GNU General Public License (GPL) và các giấy phép tương thích GPL[6] là phần mềm tự do, bao gồm: GPL, GNU Lesser General Public License (LGPL), GNU Affero General Public License (AGPL), Apache License, giấy phép BSD...

Phần mềm nguồn mở, do là một định nghĩa nới lỏng hơn của phần mềm tự do, chấp nhận nhiều loại giấy phép hơn[7]. Tất cả các phần mềm tự do đều là phần mềm nguồn mở. Điều ngược lại không hoàn toàn đúng.

Phần mềm tự do không đồng nghĩa với Linux. Trên Linux, đa số là phần mềm tự do/nguồn mở, nhưng vẫn có phần mềm thương mại. Ngược lại, bộ phần mềm từ Mozilla và OpenOffice.org là hai ví dụ cụ thể về phần mềm tự do/nguồn mở trên Windows. Thế giới phần mềm tự do/nguồn mở rộng hơn, bao gồm những hệ điều hành khác như nhóm hệ điều hành BSD, OpenSolaris...

Lưu ý là ngoài những giấy phép chỉ áp dụng cho phần mềm, còn có những giấy phép khác cho những sản phẩm/nội dung khác, theo tinh thần của phần mềm tự do. Nổi bật nhất là nhóm giấy phép Creative

[1] Viết đầy đủ là "phần mềm tự do/nguồn mở", tuy nhiên để ngắn gọn, bài viết sẽ đề cập là "phần mềm tự do"

[2] http://www.fsf.org

[3] http://www.opensource.org

[4] http://www.opensource.org/history

[5] Khái niệm người sử dụng ở đây có thể được chia làm hai nhóm: nhóm người sử dụng cuối và nhóm người phát triển chương trình. Hai nhóm người sử dụng này có những đòi hỏi khác nhau.

[6] http://www.fsf.org/licensing/licenses

[7] http://www.opensource.org/licenses

Commons[8]. Các giấy phép loại này nằm ngoài phạm vi của bài viết này.

2 Ai tạo ra phần mềm tự do?

Vào những ngày đầu của phần mềm tự do, những người tình nguyện trên khắp thế giới đã góp tay xây dựng nên phần mềm tự do. Viết bởi lập trình viên, vì lập trình viên. Hoàn toàn tự nguyện. Ngày nay, điều này không còn tuyệt đối đúng. Những phần mềm cốt lõi (như Linux kernel) được sự đầu tư của các công ty lớn trong lĩnh vực công nghệ thông tin. Ví dụ, bài "Who wrote 2.6.23"[9] của LWN.NET phân tích công sức đóng góp để tạo nên Linux kernel phiên bản 2.6.23 của các cá nhân, tổ chức. Những công ty như Red Hat, IBM, Novell, Intel... đóng góp một lượng code đáng kể.

Các công ty đầu tư vào những phần mềm tự do chủ chốt, có ảnh hưởng đến chiến lược kinh doanh, phát triển của mình. Việc giúp cho phần mềm tự do ngày càng tốt hơn giúp ích cho công ty ở khía cạnh này hoặc khía cạnh khác. Nó có thể giúp công ty tăng lợi nhuận trước mắt. Những ví dụ cụ thể bao gồm các nhà sản xuất "distro"[10] như Red Hat, Novell hay Canonical. Đây là ba nhà sản xuất của Red Hat Linux, SuSE Linux và Ubuntu Linux. Việc cải tiến Linux kernel sẽ giúp hệ điều hành Linux tốt hơn, giúp cải thiện thị phần họ nhắm tới, cả máy chủ lẫn máy để bàn, và tăng nguồn thu từ hỗ trợ sử dụng các phiên bản Linux này.

Ngoài những lợi ích cụ thể, còn có những lợi ích gián tiếp khác. Ví dụ như Oracle và Google bản thân không thu lợi nhiều từ việc bán Linux. Tuy nhiên Linux là một thành phần quan trọng trong hệ thống thông tin của những công ty này. Google sử dụng Linux trong hàng ngàn máy tìm kiếm của mình. Việc cải tiến Linux đồng nghĩa với việc cải tiến hệ thống/công cụ cho chính công ty.

Điều này không có nghĩa thế giới phần mềm tự do hiện nay hoàn toàn bị điều khiển bởi các công ty. Những phần mềm được đầu tư từ các tập đoàn vẫn giữ quyết định độc lập. Ý kiến của lập trình viên tình nguyện được tôn trọng như bất kỳ lập trình viên nào khác.

Phần lớn còn lại của phần mềm tự do vẫn được phát triển bởi những người tình nguyện trên toàn thế giới. Họ là những lập trình viên độc lập, viết phần mềm để phục vụ cho nhu cầu của mình và chia sẻ nó với mọi người. Họ là những người yêu thích việc viết phần mềm cũng như chia sẻ thành quả của mình với người khác. Đối với họ, viết phần mềm, tham gia cộng đồng là một thử thách và cũng là niềm vui. Họ là những người sử dụng và cổ suý cho tinh thần phần mềm tự do, bỏ thời gian ra để giúp phần mềm tự do ngày càng tốt hơn (lập trình không phải là việc duy nhất để tạo ra phần mềm nói chung, phần này sẽ được đề cập chi tiết hơn ở sau).

3 Tại sao sử dụng phần mềm tự do?

Lý do đầu tiên để sử dụng phần mềm tự do là nó miễn phí (quyền tự do sử dụng và phân phối). Tuy nhiên không nên nhầm lẫn phần mềm tự do với phần mềm miễn phí. Đối với phần mềm miễn phí, người sử dụng chỉ được quyền sử dụng (có giới hạn) chương trình, và đôi khi được quyền phân phát lại chương trình đó. Phần mềm tự do nhiều hơn chỉ đơn giản là "phần mềm miễn phí".

Phần mềm tự do được gọi như thế bởi vì nó nhấn mạnh sự tự do của người sử dụng. Với phần mềm tự do, chính người sử dụng nắm quyền điều khiển phần mềm mình dùng. Khi bạn dùng phần mềm tự do, phần mềm tự do là *của bạn*. Bạn được quyền làm gì tuỳ thích với những gì mình sở hữu. Bạn có thể kiểm tra, tìm hiểu nó. Bạn có thể phân phát nó cho bạn bè. Bạn có thể chỉnh sửa nó và phân phát những chỉnh sửa của mình.[11]

[8] http://creativecommons.org/

[9] http://lwn.net/Articles/247582/

[10] Linux distribution – bản phân phối Linux

[11] Như đã đề cập ở trên, "người sử dụng" bao gồm hai nhóm đối tượng khác nhau, có những đòi hỏi khác nhau. Quyền lợi của nhóm người dùng này có thể xung đột với quyền lợi của nhóm người dùng khác. Vì điều này, giấy phép phần mềm tự do thường được phân làm hai nhóm, nhóm ưu tiên người sử dụng là các nhà phát triển chương trình (điển hình là giấy phép MIT/BSD) và nhóm ưu tiên người sử dụng cuối (giấy phép GPL). Với giấy phép MIT/BSD, người phát triển có toàn quyền, bao gồm cả quyền không công bố các thay đổi mình tạo ra. Điều này làm giới hạn quyền được xem mã nguồn của người

Trên thực tế, những quyền này không hẳn được phát huy tác dụng một cách trực tiếp. Nếu bạn là một người sử dụng phần mềm, bạn không phải là một người phát triển phần mềm, quyền được xem và sửa đổi mã nguồn có thể nói là không cần thiết đối với bạn. Không hẳn là như thế. Bởi vì bạn có quyền xem và điều chỉnh phần mềm, bạn có thể tận dụng một nguồn lực khác (ví dụ như thuê lập trình viên) để sử dụng những quyền của mình. Điều này không thể xảy ra với các phần mềm thương mại.

Có một thực tế khác là do cách phát triển của phần mềm tự do, bạn không nhất thiết phải có tiền thuê lập trình viên để có thể sử dụng quyền xem và sửa đổi mã nguồn của mình. Bạn có thể *thuyết phục* những lập trình viên đã tạo ra phần mềm xem xét thay đổi theo nhu cầu của mình, chừng nào đó là nhu cầu chung, có thể cần thiết cho nhiều người dùng khác.

Với những người dùng là các tổ chức có nhu cầu đặc biệt, việc điều chỉnh các phần mềm cho phù hợp nhu cầu sử dụng hoàn toàn nằm trong tầm tay. Nếu bạn sử dụng phần mềm thương mại, khả năng thuyết phục công ty tạo ra phần mềm đưa ra sản phẩm phù hợp thường khá mong manh và hao tốn.

Phần mềm tự do, với lực lượng phát triển ở quy mô thế giới, có thể thực hiện được những việc mà các công ty phần mềm bị hạn chế về nhân sự. Một trong những ví dụ là hỗ trợ ngôn ngữ bản địa cho phần mềm. Trong khi đa số các phần mềm thương mại chỉ có thể hỗ trợ dưới mười ngôn ngữ phổ dụng trên thế giới, phần mềm tự do có thể hỗ trợ thậm chí những ngôn ngữ chỉ vài triệu người sử dụng.

Có những lợi ích gián tiếp từ những quyền đã làm nên phần mềm tự do. Do mã nguồn phần mềm tự do được công khai, mọi người đều có thể xem, bạn có thể an tâm là phần mềm mình kiểm tra bởi một lượng lớn lập trình viên trên thế giới. Nếu một lỗi an ninh được phát hiện, nhiều khả năng nó sẽ được khắc phục hết sức nhanh chóng. Dĩ nhiên điều này chỉ đúng với những phần mềm có tầm ảnh hưởng lớn, như Linux kernel. Một ứng dụng nhỏ, với lượng người dùng ít, sẽ không có được những ưu thế này.

Đối với người sử dụng là những tổ chức, doanh nghiệp, hay lớn hơn là một quốc gia, quyền được sở hữu phần mềm tự do trở thành điểm then chốt. Hãy hình dung, công ty của bạn phụ thuộc vào một phần mềm thương mại. Điều gì sẽ xảy ra nếu một ngày kia công ty đó quyết định thôi không bảo trì, phát triển phần mềm đó? Nhiều khả năng bạn phải đối mặt với việc thay thế phần mềm này bằng một phần mềm khác, ảnh hưởng đến việc kinh doanh của công ty mình. Nếu điều này xảy ra với phần mềm tự do, bạn sẽ cần bỏ một số tiền để thuê một công ty/cá nhân khác tiếp tục phát triển phần mềm theo nhu cầu của bạn.

Đối với các quốc gia, việc sử dụng các phần mềm được tạo ra bởi các quốc gia không thân thiện là xâm hại an ninh quốc gia. Do không thể kiểm tra mã nguồn, một quốc gia khó lòng biết được nếu quốc gia khác có cài mã độc hoặc mã theo dõi vào những phần mềm mình đang sử dụng hay không. Với phần mềm tự do, một quốc gia sẽ hoàn toàn làm chủ công nghệ mình đang sử dụng.

4 Tại sao không sử dụng phần mềm tự do?

Nói thế không có nghĩa là phần mềm tự do hoàn hảo. Phần mềm tự do cũng có những điểm yếu, và đó là lý do tại sao có bài viết này: để cải thiện những điểm yếu của phần mềm tự do.

Như đã nói ở trên, phần mềm tự do có thể được viết bởi những chuyên gia phần mềm, được các công ty thuê để viết phần mềm tự do, hoặc những người lập trình độc lập. Với những phần mềm nhận được đầu tư từ các công ty, chất lượng phần mềm hoàn toàn không thua kém phần mềm thương mại, thậm chí còn vượt hơn do những ưu điểm trong mô hình phát triển phần mềm tự do. Ngược lại, những phần mềm do sự đóng góp tự nguyện của các lập trình viên trên thế giới có độ ổn định kém hơn, lỗi nhiều hơn.

Một khuyết điểm khác của phần mềm tự do là thiếu tài liệu hướng dẫn. Do lập trình viên là nguồn lực chủ yếu tạo nên sản phẩm, họ đôi khi không chú trọng vào việc tạo ra các tài liệu hỗ trợ sử dụng cũng như phát triển chương trình. Một câu nói khá phổ biến trong giới phần mềm tự do, “source is document”, phản ánh điều này.

sử dụng cuối. Giấy phép GPL, ngược lại, bảo đảm người sử dụng cuối có toàn quyền, nghĩa là cấm quyền đóng mã nguồn đối với những người phát triển phần mềm.

Cũng lý do "viết bởi lập trình viên, dành cho lập trình viên", phần mềm tự do thường khó sử dụng đối với người dùng cuối, không hiểu rõ về hệ thống. Giao diện đồ hoạ, các thông báo lỗi, đôi khi khó hiểu và không hiệu quả.

Việc giao mọi trọng trách trong quy trình phát triển, bảo trì, hỗ trợ sử dụng phần mềm cho lập trình viên trong một số trường hợp dẫn đến hiểu lầm trong giao tiếp bởi cách trả lời thô lỗ của lập trình viên khi hỗ trợ người dùng sử dụng phần mềm.

Phần mềm tự do ngoài ra có thể không đáp ứng mọi lĩnh vực trong đời sống. Do được tạo ra theo nhu cầu của đa số lập trình viên, những lĩnh vực ít được lập trình viên quan tâm cũng đồng nghĩa với việc thiếu vắng các phần mềm tự do, ví dụ như các ứng dụng hỗ trợ xây dựng. Việc thiếu các thông tin cần thiết cho xây dựng phần mềm cũng có ảnh hưởng, rõ ràng nhất là viết các trình điều khiển cho các phần cứng mà nhà sản xuất không cung cấp đặc tả phần cứng. Cũng có những lĩnh vực mà công sức bỏ ra để xây dựng phần mềm đơn giản vượt quá khả năng của một nhóm nhỏ lập trình viên, như việc viết lại toàn bộ thư viện hỗ trợ Flash 10. Trong những trường hợp này, sử dụng phần mềm thương mại là điều không tránh khỏi.

5 Tại sao phải đóng góp cho phần mềm tự do?

Những lý do không sử dụng phần mềm tự do cũng đồng thời là những lý do phải đóng góp cho phần mềm tự do. Đó không phải là những nguyên nhân không thể khắc phục. Và những ưu điểm mà phần mềm tự do mang lại khó có thể tìm thấy ở phần mềm thương mại.

Nếu bạn không đồng ý, dĩ nhiên bạn không cần phải đóng góp cho phần mềm tự do. Đóng góp cho phần mềm tự do, đó vừa là nghĩa vụ, vừa là quyền lợi.

Đóng góp cho phần mềm tự do là quyền lợi, bởi vì nhờ đó bạn có thể can thiệp vào quá trình phát triển phần mềm, gợi ý cho nhà phát triển những tính năng mình cần mà nhà phát triển không thấy (chừng nào vẫn còn phù hợp với mục tiêu dự án).

Bằng cách đóng góp vào phần mềm tự do, bạn mở rộng các mối quan hệ, quen được nhiều người (từ khắp địa cầu). Hãy thử và bạn sẽ thấy. Một số rất khác người. Một số lập dị. Tuy nhiên đa phần thân thiện và dễ hoà đồng. Một số là những người rất tài năng.

Bằng cách cùng làm việc với một trong những chuyên gia phần mềm hàng đầu thế giới, bạn có thể nâng cao khả năng tư duy, kỷ luật khi làm việc, cách giao tiếp sao cho thực sự hiệu quả khi không thể trò chuyện mặt đối mặt.

Đóng góp vào phần mềm tự do, bạn đang giúp một ai đó trên thế giới này. Những lời cám ơn và các món quà nhỏ luôn là động lực tuyệt vời.

Đóng góp, mặt khác còn là trách nhiệm. Là một người tử tế, nếu bạn nhận ơn người khác, bạn có trách nhiệm phải trả ơn. Sử dụng phần mềm tự do là bạn đang tận dụng công sức của người khác. Vậy khi bạn có khả năng, bạn nên đóng góp, coi như một cách để đền bù công sức những người khác đã bỏ ra, một cách để thể hiện sự biết ơn.

6 Tiếp cận việc phát triển phần mềm tự do

6.1 Giao tiếp

Mối quan hệ của người phát triển và sử dụng phần mềm tự do không phải là mối quan hệ nhà cung cấp/khách hàng. Người sử dụng là một phần trong quy trình phát triển phần mềm. Người sử dụng và người phát triển cùng nhau tạo nên một cộng đồng chung quanh phần mềm. Mọi người đều bình đẳng, không có "khách hàng là thượng đế", cũng chẳng có độc quyền dẫn đến không quan tâm đến khách hàng. Do vậy, nếu bạn là một người sử dụng phần mềm, bạn có quyền góp ý, nhờ trợ giúp, nhưng bạn không có quyền ra lệnh người phát triển phải giúp bạn hoặc làm theo ý bạn. Giữ lối suy nghĩ "nhà cung cấp/khách hàng" có thể gây tác dụng ngược.

Ngoài ra, do môi trường liên lạc là Internet, mọi người đến từ các quốc gia khác nhau, có ngôn ngữ riêng, văn hoá đặc thù, những đặc điểm này có thể dẫn đến hiểu lầm tai hại trong giao tiếp. Cố gắng làm rõ quan điểm của mình, luôn lịch sự, nếu cảm thấy

bị xúc phạm, hãy liên lạc lại để làm rõ vì đôi khi chỉ là hiểu lầm do khác biệt văn hoá/ngôn ngữ.

Mỗi dự án phần mềm tự do có văn hoá ứng xử riêng. Bạn nên theo dõi những trao đổi của các thành viên cũ một thời gian để làm quen với môi trường phát triển phần mềm trước khi lên tiếng.

Có nhiều kênh giao tiếp khác nhau. Mỗi dự án thường ưu tiên một số cách giao tiếp hơn những cách khác. Với đa số dự án phần mềm tự do, hộp thư chung (mailing list) vẫn là phương tiện chính để trao đổi về phát triển phần mềm. Hộp thư chung cũng dùng để liên lạc giữa người dùng với nhau, nhưng diễn đàn Web đang dần chiếm ưu thế hơn. Một số dự án lại chọn IRC[12] làm cách liên lạc chính. Với một số dự án, phần lớn trao đổi diễn ra trên các hệ thống quản lý lỗi như bugzilla. Thông tin về cách liên lạc luôn có trên trang Web của dự án, hoặc trong các tài liệu đi kèm với mã nguồn phần mềm.

6.1.1 Tìm trợ giúp

Tài liệu "How To Ask Questions The Smart Way"[13] (bản dịch tiếng Việt ở `forum.vnoss.org/sq.php`) thường được đề cập, và thật sự rất nên đọc ít nhất một lần trong đời. Tóm gọn như sau:

- Chọn đúng nơi để hỏi.
- Đặt mình vào vị trí người trả lời trước khi hỏi.
- Tuân thủ các quy tắc trao đổi ở nơi hỏi (cụ thể là không top-post[14]).
- Viết tiêu đề rõ ràng, đi thẳng vào nội dung cần hỏi.
- Mô tả chính xác, ngắn gọn, đầy đủ thông tin.
- Lịch sự khi trả lời.

[12] Internet Relay Chat – một hình thức hội thoại trực tuyến, như Yahoo Messenger!

[13] Bản gốc tiếng Anh ở http://catb.org/ esr/faqs/smart-questions.html.

[14] Top-post là hình thức trả lời ở đầu thư trong khi vẫn để câu hỏi ở phía sau. Với top-post, thư bạn sẽ có dạng: "trả lời – hỏi", gây khó khăn cho người đọc. Cách trả lời đúng là trích dẫn câu hỏi cần trả lời và ghi câu trả lời của mình ngay sau câu hỏi. Nếu bạn nhận được nhiều câu hỏi trong một thư, bạn có thể ngắt thư gốc ra nhiều phần và trả lời riêng từng câu hỏi, theo dạng: "câu hỏi 1 – câu trả lời 1 – câu hỏi 2 – câu trả lời 2..." như minh hoạ ở http://en.wikipedia.org/wiki/Posting_style#Bottom-posting

6.2 Giúp đỡ người sử dụng

Giúp đỡ những người sử dụng mới là một trong những cách đóng góp đơn giản nhưng hiệu quả. Bạn trở thành nguồn tài liệu sống, giúp khắc phục điểm yếu về tài liệu của phần mềm tự do.

6.3 Báo lỗi

Báo lỗi là trách nhiệm cao cả (và nặng nề) của người sử dụng trong thế giới phần mềm tự do. Ở đây, người sử dụng là một phần của quy trình phát triển phần mềm. Những phản hồi của người dùng sẽ giúp tăng chất lượng và độ hữu dụng phần mềm. "Báo lỗi" còn được dùng để gợi ý những tính năng mới. Để báo lỗi về bản dịch, xem phần "Dịch thuật và viết tài liệu".

Tuy nhiên thông báo lỗi không hề đơn giản. Với vai trò người báo lỗi, bạn cần cung cấp chi tiết thông tin về môi trường phát hiện lỗi (bạn đang sử dụng những phần mềm nào, phiên bản bao nhiêu, trên những phần cứng nào...), cách tái tạo lỗi, hành vi mong muốn... Bạn đồng thời là người giúp người phát triển chẩn đoán, tìm lỗi và thử phần mềm mới để đảm bảo lỗi đã được sửa.

Việc đầu tiên khi báo lỗi là tìm nơi báo lỗi. Các dự án khác nhau có các cách thông báo lỗi khác nhau: đa số sử dụng bugzilla để theo dõi lỗi, Ubuntu sử dụng phần mềm riêng Launchpad, Debian quản lý lỗi qua thư... Một số dự án phần mềm thậm chí không có hệ thống quản lý lỗi. Thông thường, bạn sẽ tìm thấy nơi để thông báo lỗi và cách thông báo lỗi trên trang Web của phần mềm gặp lỗi. Nếu không chắc chắn, bạn có thể gửi thư đến hộp thư chung của người phát triển phần mềm, hoặc cá nhân những người phát triển để hỏi cụ thể cách làm.

Lưu ý luôn lịch sự, nói rõ mình muốn thông báo lỗi nhưng không biết phải làm thế nào và đề nghị hướng dẫn. Thông thường hộp thư chung sẽ được ưa chuộng hơn so với gửi trực tiếp hỏi từng cá nhân (một phần để bảo đảm tính mở trong phát triển, một phần vì cá nhân bạn hỏi có thể vắng mặt một thời gian, nếu hỏi hộp thư chung, bạn vẫn nhận được câu trả lời từ những người khác).

Trong nhiều trường hợp, người phát triển sẽ yêu cầu bạn thử lại trên phiên bản mới nhất của phần mềm vì nghi nghờ lỗi đã được sửa. Nếu bạn không biết phải làm sao để cài đặt phiên bản mới nhất trên hệ thống mình, bạn có thể lịch sự hỏi người phát triển nhờ giúp đỡ, nói rõ phiên bản Linux distro bạn đang sử dụng (hoặc hệ điều hành khác, nếu bạn không dùng Linux).

Thông thường, bạn sẽ báo lỗi trực tiếp cho distro bạn đang sử dụng. Distro, với tư cách là "hạ nguồn" (downstream), sẽ xem xét nếu lỗi đó là đặc thù của distro và tự sửa, hoặc lỗi chung của "thượng nguồn" (upstream) và chuyển thông báo lỗi về thượng nguồn.[15]

Nếu bạn chép phần mềm từ thượng nguồn và tự biên dịch chương trình, dĩ nhiên bạn không cần thông qua distro mà liên lạc trực tiếp với thượng nguồn. Điều này cũng áp dụng nếu bạn không dùng Linux distro (ví dụ như dùng OpenOffice.org trên Windows).

Trước khi mở lỗi mới trên hệ thống quản lý lỗi. Luôn tìm lỗi tương tự để bảo đảm không mở trùng một lỗi đã có. Các hệ thống quản lý lỗi (như bugzilla) thường cho phép bạn theo dõi hoạt động xử lý lỗi bằng cách gửi thư cho bạn mỗi khi có thay đổi liên quan đến lỗi, nhờ đó bạn có thể biết khi nào thì lỗi ảnh hưởng đến mình được sửa, hoặc người phát triển cần thêm thông tin gì từ mình.

Một số lỗi có thể được đóng với những "giải pháp" nghe phản cảm (ví dụ như INVALID trên bugzilla). Nếu bạn thấy khó chịu về cách đặt tên, hãy phản ảnh với người quản lý hệ thống, hoặc những người phát triển.

Một số trường hợp, người phát triển không nhất thiết đồng ý với bạn (như cho rằng đó không phải là lỗi, hoặc không đúng đường lối phát triển của phần mềm...). Bạn nên kiên trì (và lịch sự) giải thích quan điểm của mình nếu cho rằng mình đúng.

[15] Do trong Linux, đa số các phần mềm được phát triển độc lập. Các distro sẽ gom các phần mềm này lại với nhau, điều chỉnh một số cho phù hợp với distro, sau đó phân phối cho người dùng. Mỗi phần mềm có thể được đóng gói bởi nhiều distro khác nhau, mỗi gói của một distro có thể khác nhau một chút. Mô hình này giống như cùng một thượng nguồn của một con sông được tẻ ra làm nhiều nhánh (hạ nguồn) khác nhau.

Một số lập trình viên khá thô lỗ khi trả lời. Trong hầu hết trường hợp họ không có ác ý. Tuy nhiên nếu bạn không chấp nhận thái độ như thế, bạn có thể từ chối sử dụng phần mềm đó (và nói điều đó với những người phát triển).

6.3.1 Tự biên dịch và chạy thử phần mềm mới

Hèm... khúc này khó à.

6.4 Dịch thuật và viết tài liệu

Đây là một trong những lĩnh vực tốn nhiều công sức nhưng lại dễ thực hiện (dù thực hiện với chất lượng cao thì không dễ).

Trước khi dịch, bạn nên liên lạc với những nhóm chịu trách nhiệm dịch phần mềm bạn định dịch. Các nhóm này sẽ đưa ra các chỉ dẫn về cách dịch, kiểm tra bản dịch và đưa bản dịch về nguồn.

Để thông báo lỗi bản dịch, bạn cần lưu lại chính xác chuỗi dịch sai, tên và phiên bản ứng dụng bạn đang dùng. Dễ dàng nhất là bạn chụp hình phần dịch sai và gửi đến các nhóm có trách nhiệm.

6.5 Sửa lỗi, phát triển phần mềm

Phần mềm tự do được xây dựng chung quanh, dễ đoán, mã nguồn tự do. Mã nguồn là cốt lõi. Thay đổi mã nguồn cũng giống như đụng chạm vào tài sản quý nhất, do đó thái độ của người phát triển với bạn có thể không được nhã nhặn như khi bạn liên lạc với tư cách người sử dụng.

Cách tốt nhất để bắt đầu đóng góp vào mã nguồn một phần mềm nào đó là sửa lỗi. Bạn sẽ học được nhiều thứ. Bạn sẽ nắm vững mã nguồn hơn trước khi thật sự thêm những tính năng mới.

Cách đối xử giữa những người phát triển với nhau (và cả những người đóng góp mới) khác nhau một trời một vực giữa các dự án phần mềm tự do. Đó có thể là một cộng đồng cực kỳ thân thiện. Nếu bạn gửi một bản patch lên hộp thư chung của dự án Cairo[16],

[16] http://cairographics.org – Thư viện vẽ đồ hoạ nền tảng cho GTK+ và GNOME

Carl Worth, quản lý dự án[17] Cairo, sẽ cám ơn bạn rối rít, sau đó mới chỉ ra bạn cần phải làm thế này, thế này... và hi vọng bạn sẽ gửi bản cập nhật khác.

Đối nghịch với dự án Cairo là Linux kernel. Nếu bạn gửi patch lần đầu lên hộp thư chung linux-kernel mà không bị chê bai và từ chối, bạn là một trong số ít những lập trình viên ưu tú. Linus Torvalds, cha đẻ Linux, rất hay công kích, và đã công kích thì dùng lời lẽ cực kỳ thiếu khiêm tốn. Mọi người trên linux-kernel không phiền, đơn giản đó là cách họ trao đổi và làm việc.

Để tham gia phát triển phần mềm, bạn phải làm quen với môi trường phát triển mà phần mềm đó sử dụng, vốn cực kỳ đa dạng trong thế giới phần mềm tự do. Đại đa số các ứng dụng trên Linux hiện nay vẫn được viết bằng C/C++, với hệ thống biên dịch dự trên autoconf/automake hoặc cmake, sử dụng hệ thống quản lý phiên bản như Subversion, Git, Mercurial... Tuy nhiên không có gì là chuẩn. Tham gia nghĩa là "nhập gia tuỳ tục". Bạn phải sử dụng cùng bộ công cụ phát triển những người khác đang dùng. Cụ thể, nếu bạn thích dùng Subversion, nhưng dự án bạn định tham gia dùng Git, bạn nên xem xét sử dụng Git, dù chỉ cho dự án này.

Thường sau khi sửa đổi mã nguồn, bạn không gửi nguyên mã nguồn mới cho người khác mà gửi dưới dạng "patch". Một tập tin patch sẽ mô tả chính xác những sửa đổi của bạn, và có thể được dùng để tái tạo lại phiên bản mã nguồn bạn đã sửa, bằng chương trình `patch`. Xem lại tập tin patch của mình trước khi gửi đi là một thói quen tốt. Loại bỏ những thay đổi không cần thiết (như xoá khoảng trắng, đổi tên...). Bình thường những bản patch "không sạch" rất khó được chấp nhận.

Cách gửi patch cho dự án cũng khác nhau. Một số dự án như những ứng dụng thuộc GNOME đa phần chỉ chấp nhận patch trên hệ thống quản lý lỗi (cụ thể là bugzilla.gnome.org với GNOME). Gửi patch lên hộp thư chung, đa phần bạn sẽ được yêu cầu mở "bug" và đính kèm patch vào. Ngược lại, Git chuộng gửi patch qua hộp thư chung, thay đổi được chia nhỏ ra một tập patch, mỗi patch được mô tả rõ ràng trong phần thông tin commit. Nếu không rõ những nguyên tắc phát triển, bạn nên hỏi trước.

Những đóng góp của bạn có thể bị từ chối do chưa đạt chất lượng. Bạn có thể ghi nhận các góp ý, làm lại và gửi lên lại. Việc này tuy tốn thời gian, nhưng góp phần nâng cao khả năng của chính bạn. Nếu đóng góp của bạn bị từ chối vì "không phù hợp với mục đích của dự án", đừng quá thất vọng. Bạn có thể cố gắng thuyết phục những người quản lý dự án, hoặc chuyển sang dùng phần mềm khác (và đóng góp vào đó), hoặc tách nhánh phần mềm (sẽ được bàn cụ thể hơn ở phần sau).

6.6 Phát triển phần mềm mới, tách nhánh phần mềm

Vậy, bạn đã quyết định phần mềm hiện có không phù hợp với mình. Bạn muốn đi con đường riêng. Theo cách gọi của phần mềm tự do, bạn "tách nhánh" (fork) phần mềm.

Trong lịch sử phát triển của phần mềm tự do, có rất nhiều ví dụ tách nhánh thành công. Hai ví dụ nổi bật nhất là EGCS (một nhánh của GCC, sau này thay thế chính GCC) và Xorg (tách nhánh từ XFree86 và là bản X Window được dùng trên hầu hết các distro hiện nay). Tuy nhiên tách nhánh đòi hỏi phải đầu tư nhiều công sức và gây xung đột trong cộng đồng phát triển. Nếu tránh được, nên tránh. Xem thêm bài "The Freedom of Fork"[18] của LWN.NET.

Có một cách khác nữa để có được phần mềm vừa ý mình là viết mới. Trước khi viết, bạn nên tự hỏi mình, liệu có ai đã viết một phần mềm tương tự như thế chưa? Câu trả lời trong 99% trường hợp là "có".

Có rất nhiều phần mềm tự do khác nhau cùng thực hiện một công việc. Đa số khác nhau về cách tiếp cận vấn đề, cách thực hiện công việc. Một số khác nhau ở ngôn ngữ lập trình, theo ngôn ngữ ưa thích của người phát triển. Nếu chịu khó tìm kiếm, bạn sẽ tìm ra một phần mềm phù hợp với nhu cầu của mình, đồng thời phù hợp với khả năng tham gia của mình.

Viết phần mềm mới là quyền của bạn. Không ai có quyền cấm bạn viết ra và công bố. Tuy nhiên, những phần mềm với một người phát triển thường chỉ dừng ở những ứng dụng nhỏ, khó có thể phát triển hoàn chỉnh, đầy đủ tính năng. Nếu viết mới đơn giản là

[17] Chữ gốc là maintainer. Nếu dịch sát nghĩa thì là người bảo trì phần mềm.

[18] http://lwn.net/Articles/282261/

niềm vui của bạn, bởi vì bạn *thích viết*, nhớ sử dụng giấy phép phần mềm tự do! Nếu bạn viết bởi vì bạn cần một phần mềm với tính năng như thế, bạn nên bỏ thời gian tìm kiếm trước.

Trước hết, tìm trong kho phần mềm của distro bạn đang dùng. Kho phần mềm Debian/Ubuntu, Fedora hay Gentoo không hề nhỏ. Có nhiều thứ bạn chưa nhận ra. Bạn còn có thể tìm ở những kho phần mềm tự do lớn, như `sourceforge.net`, `code.google.com`, `savannah.gnu.org`... hoặc "chợ thịt tươi" `freshmeat.net`. Nếu bạn nhắm đến các thư viện hoặc các ứng dụng được viết bằng một ngôn ngữ nhất định, nên tìm đến những kho chứa phần mềm chuyên biệt (như `cpan.perl.org` cho Perl, `rubyforge.org` cho Ruby, `trapexit.org` cho Erlang, `pear.php.net` cho PHP...). Hỏi thăm trên các diễn đàn, hộp thư hỗ trợ cũng giúp tìm phần mềm phù hợp.

Sau cùng nếu vẫn quyết định viết mới. Bạn sẽ cần đưa ra một vài quyết định. Đầu tiên là một chỗ trên mạng cho phần mềm của bạn. Sourceforge.net vẫn luôn là một lựa chọn, với mười năm kinh nghiệm phục vụ cộng đồng. Ngoài ra còn nhiều dịch vụ cung cấp chỗ miễn phí cho phần mềm tự do khác, một số đã đề cập ở trên. Nếu phần mềm của bạn dùng ngôn ngữ đặc trưng, nên dùng kho phần mềm riêng cho ngôn ngữ đó, bạn sẽ dễ được tìm thấy trong cộng đồng ngôn ngữ đó hơn. Luôn luôn lưu trữ bản sao lưu, các dịch vụ miễn phí có thể chấm dứt hoạt động bất cứ lúc nào.

Việc lựa chọn phần mềm quản lý mã nguồn có tác động đến việc sao lưu. Trào lưu hiện nay thiên về dùng những hệ thống quản lý mã nguồn phân tán (ba phần mềm nổi trội là Git, Mercurial và Bazaar). Với phần mềm quản lý mã nguồn phân tán, bạn có thể chép toàn bộ lịch sử phần mềm của mình ở nhiều nơi, giảm phụ thuộc vào nơi cung cấp chỗ chứ trên mạng. Trong trường hợp bất khả kháng, bạn vẫn có thể dùng Subversion hoặc CVS.

Điều quan trọng kế tiếp là giấy phép. Thông thường, giải pháp đơn giản nhất là dùng giấy phép GPL (phiên bản 2 hoặc 3) cho ứng dụng, hoặc LGPL cho thư viện. Nếu bạn thật sự không quan tâm, "giấy phép public domain" có thể được dùng: bạn tuyên bố đây là sản phẩm công cộng, bất cứ ai cũng có thể làm bất cứ gì, thậm chí loại bỏ tên tác giả. Nếu bạn quan tâm đến kinh doanh, lưu ý một số giấy phép sẽ hạn chế đóng mã nguồn để kinh doanh, cụ thể là GPL. Nếu sự chú ý của bạn là những người phát triển và muốn thu hút họ, nên xem xét giấy phép BSD/MIT, cho phép nhiều quyền hơn với người phát triển (và hạn chế quyền của người sử dụng).

Một phần mềm cần có một "bản thông cáo" để giới thiệu. Hầu hết các dịch vụ cung cấp chỗ chứa đều hỗ trợ cách để công bố, như HTML hoặc Wiki. Thông tin quan trọng bao gồm giới thiệu tóm tắt về phần mềm và cách liên lạc. Nếu có nhiều hơn bốn người sử dụng, bạn có thể sẽ cần một hộp thư chung, hoặc một diễn đàn nhỏ để duy trì cộng đồng nhỏ của phần mềm của mình.

Hai điều cuối cùng cần làm, và cũng là hai điều khó nhất, là viết phần mềm, liên tục cải tiến phần mềm, và duy trì cộng động chung quanh phần mềm đó. Từ giờ bạn không còn là người dùng cuối nữa. Bạn trở thành "đầu nguồn". Hãy nghĩ đến những khó khăn khi vẫn còn là người dùng cuối và giúp những người dùng mới vượt qua. Chúc thành công!