

Bản quyền và giấy phép Phần mềm Tự do Nguồn mở¹

(NHQ – Dịch và giới thiệu – Đã đăng tải trên tạp chí Tin học & Đời sống số tháng 9-10 / 2009)

Những nguyên tắc cơ bản

Các chương trình nguồn mở không phải là những chương trình *không giấy phép*. Ngược lại, chính giấy phép của chúng đã làm chúng thành *nguồn mở*. Chúng cũng không phải nằm trong *miền công cộng* (*public domain*), nghĩa là không phải của riêng ai, hoặc chỉ ít được miễn trừ quyền di sản.

Khi một nhà phát triển viết một chương trình, anh ta giữ quyền tác giả, hay *bản quyền* (*copyright*). Trong một số trường hợp, có thể hãng làm việc của anh ta nắm giữ các quyền đó. Và cái bản quyền này có thể được bán, như một tài sản phi vật chất, từ hãng này qua hãng khác.

Người giữ bản quyền được quyền tự do định ra chương trình của anh ta có thể được sử dụng như thế nào:

- Anh ta có thể giữ riêng cho mình, cấm bất kỳ ai sử dụng nó.
- Anh ta có thể bán những quyền của mình cho một người khác, người thật hay danh nghĩa.
- Anh ta có thể dùng quyền tác giả qui định những điều kiện áp đặt cho việc sử dụng chương trình của mình. Anh ta viết ra các điều kiện này trong những điều khoản của *giấy phép sử dụng*.

Cần chú ý rằng, theo luật của Pháp, không phải dễ dàng *từ bỏ quyền tác giả* của mình và đặt chương trình vào miền công cộng một cách không thể đảo ngược được.

Cũng cần giải thích rõ ràng rằng không phải cứ truyền bá mã nguồn là làm cho một chương trình trở thành nguồn mở, mà phải là quyền, được ghi rõ trong giấy phép, tự do sử dụng, sửa đổi và phân phối lại chúng.

Vì thế cần thấm nhuần sâu sắc lô-gíc sau đây: ở cơ sở của nguồn mở có *giấy phép*, và giấy phép chỉ tồn tại bắt nguồn từ *quyền tác giả*.

Vì thế, tất cả các phần mềm nguồn mở có một *chủ sở hữu*, chúng không phải là « *chẳng thuộc về ai* ». Trong một số trường hợp, chủ sở hữu này có thể là một quỹ phi lợi nhuận, hoặc cũng có thể là một hãng thương mại thông thường. Cũng có thể là sở hữu của *nhiều đồng tác giả*, đặc biệt trong trường hợp hệ quả của những đóng góp về sau.

Người giữ bản quyền được tự do ấn định các điều kiện của các giấy phép, anh ta cũng được quyền tự do thay đổi chúng, tự do bố trí dàn xếp lại nội dung, thêm bớt các ngoại lệ, hoặc truyền bá một số phần mềm này theo giấy phép này, một số khác theo một giấy phép khác.

Ngược lại, người nhận chương trình thì không được tự do. Anh ta bị ràng buộc bởi các điều khoản của giấy phép. Dĩ nhiên anh ta đã không ký vào một hợp đồng nào, nhưng giấy phép đã được thông báo rõ ràng cho anh ta, và nó qui định rằng anh ta chỉ có quyền sử dụng chương trình dưới những điều kiện này hay điều kiện nọ. Nếu anh ta từ chối các điều kiện đó, anh ta không được quyền sử dụng chương trình.

Những điều khoản sơ đẳng của các giấy phép

Tất cả các giấy phép nguồn mở đều có chung một số điều khoản tích cực sau đây:

- Xác định rõ ràng chủ sở hữu của bản quyền, bao gồm cả đến các bản sao và các bản dẫn xuất.
- Sự bắt buộc bảo toàn ghi chú giấy phép như ban đầu, trên chương trình và các phiên bản dẫn xuất.
Tất nhiên đây là một sự cần thiết kỹ thuật: mất công định nghĩa các điều khoản giấy phép làm gì

¹ Nguồn: Sách trắng « *Introduction à l'open source et au logiciel libre* »
(http://ftp.smile.fr/client/Livres_blancs_Smile_2/LB_Smile_Open_source.pdf)

nếu chúng bị dỡ bỏ ngay từ bản sao đầu tiên.

- Sự bảo vệ tác giả đối với người sử dụng chương trình của anh ta vì những sai sót có thể và hậu quả do các sai sót đó: « *chương trình này được cung cấp 'trong hiện trạng' (« as is ») ...* ». Đây đúng là điều tối thiểu có thể được áp đặt: tác giả đã cho bạn tự do sử dụng thành quả của anh ta, vậy nên dù thế nào bạn cũng sẽ không truy cứu anh ta về những thiệt hại và lợi nhuận.

Cũng phải lưu ý rằng ở một số nước, sự phân phối có thu tiền một chương trình kéo theo những quyền không thể từ chối. Nhưng nói chung, giấy phép không thể trái với luật pháp quốc gia. Vì thế giấy phép ghi rõ « *Nếu bạn không thể phân phối chương trình sao cho thỏa mãn đồng thời các nghĩa vụ đi kèm giấy phép và những nghĩa vụ áp dụng được khác, thì bạn nhất thiết không thể phân phối chương trình* ».

Có nghĩa là, hoặc ta có thể đồng thời tôn trọng luật pháp quốc gia và giấy phép, hoặc là ta bị cấm phân phối chương trình dưới giấy phép này.

Định nghĩa phần mềm tự do

Như đã đề cập ở trên, phần mềm tự do được định nghĩa bằng sự tôn trọng bốn quyền tự do nền tảng sau đây:

- chạy chương trình,
- nghiên cứu chương trình và thích ứng nó theo yêu cầu của bạn (điều này kéo theo tất yếu quyền truy cập mã nguồn),
- phân phối lại chương trình để giúp đỡ người tiếp theo,
- và sau cùng, cải tiến chương trình và phân phối lại những cải tiến này cho mọi người (điều này cũng kéo theo quyền tự do truy cập mã nguồn).

Như chúng ta đã đề cập, đích tối thượng là quyền tự do, quyền được truy cập mã nguồn chỉ là điều kiện cần để tôn trọng quyền tự do này.

Định nghĩa của một giấy phép nguồn mở

Tổ chức Sáng kiến Nguồn mở (OSI, *Open Source Initiative*) đã ban bố một định nghĩa chính xác về nguồn mở (*open source*) nghĩa là gì, một định nghĩa ngày nay được thừa nhận một cách gần như ở mọi nơi.

Có một định nghĩa chính thức chính xác là rất quan trọng, một giấy phép không thể *ít nhiều nguồn mở*: nó là nguồn mở hoặc là không, mọi sự phải được rạch ròi.

Trên trang web của OSI, *opensource.org*, cũng chỉ ra những giấy phép chủ yếu nào tương thích với định nghĩa này. Lẽ đương nhiên, người ta phải tìm thấy ở đây những giấy phép quen thuộc, bắt đầu bằng giấy phép GPL.

Định nghĩa gồm 10 điểm, trong đó 3 điểm đầu tiên là chủ yếu:

1. Tự do phân phối lại: giấy phép không thể cấm bất kỳ ai bán hoặc cho chương trình.
2. Mã nguồn: giấy phép phải cho phép sự phân phối dưới dạng mã nguồn, và nếu mã nguồn không đi kèm chương trình, nó phải truy cập được một cách dễ dàng và miễn phí trên thực tế.
3. Phiên bản dẫn xuất: giấy phép phải cho phép các thay đổi và các phiên bản dẫn xuất, và phải cho phép rằng những phiên bản này được phân phối dưới cùng các điều khoản giấy phép.

Hãy quay lại điểm 3 này: giấy phép phải cho phép tối thiểu phân phối lại các phiên bản dẫn xuất dưới cùng một giấy phép và giấy phép không cần phải bắt buộc điều này. Chúng ta sẽ thấy sau đây rằng chính sự mập mờ này sẽ là cơ sở để phân biệt họ giấy phép BSD với họ GNU.

Trong số các điều khoản khác của định nghĩa này có những điều khoản khác nhau về sự *không phân biệt đối xử (non-discrimination)*: giấy phép không thể loại trừ nhóm NSD này, cũng như miền ứng dụng kia. Thí dụ, tác giả của chương trình không thể, với tư cách một người ủng hộ hòa bình, nói cụ thể rằng chương trình của anh ta không thể được dùng để dẫn dắt tên lửa. Chỉ cần anh ta thêm điều khoản này vào giấy phép thì nó sẽ không còn là *open source* nữa.

Giấy phép GNU và BSD

Có hai họ giấy phép nguồn mở lớn: họ BSD và họ GNU. Đôi khi người ta gọi giấy phép *copyleft* những giấy phép của họ thứ hai và *non copyleft* những cái thuộc họ đầu. « *Copyleft* » tất nhiên là một cách chơi chữ của từ « *copyright* », nhưng tuy vậy *copyleft* không phải là sự chối bỏ tác quyền.

Để không có sự nhầm lẫn, cần nói rõ rằng cho dù phong trào phần mềm tự do ưa chuộng các giấy phép *copyleft* hơn, bắt đầu bằng GPL, nhưng không có sự *tương ứng* giữa *phần mềm tự do* và *copyleft*: các giấy phép BSD cũng là của phần mềm tự do.

Họ BSD

Giấy phép BSD (*Berkeley Software Distribution*) cho phép mọi sử dụng của chương trình, mã nguồn của nó và các dẫn xuất. Đặc biệt, mã nguồn dưới giấy phép BSD có thể được sử dụng tích hợp với những phần mềm dưới giấy phép không nguồn mở. Chúng ta đều biết rằng Microsoft đã lấy lại mã nguồn TCP-IP dưới giấy phép BSD đóng gói vào trong Windows, và rằng MacOS X được xây dựng dựa trên FreeBSD.

Ràng buộc đặc trưng duy nhất là cấm lợi dụng danh nghĩa của tác giả, ở đây là Đại học Berkeley, để làm tiền hoặc tạo lợi thế cho mình.

Vì thế đây là giấy phép phóng túng nhất, kéo theo ít ràng buộc nhất: các chương trình dưới giấy phép BSD gần như nằm trong vùng công cộng. Có lẽ đó cũng là giấy phép cổ nhất, vì phải lùi lại đến năm 1980. Nó không cấm thay đổi nội dung của giấy phép, đến mức người ta gặp vô số các phiên bản dẫn xuất, có khi khác nhau chỉ vài từ. Đây là một nhược điểm cho tính sáng sủa và dễ hiểu của giấy phép.

Trong họ BSD, người ta cũng tìm thấy giấy phép MIT, và giấy phép Apache. Giấy phép sau này có tầm quan trọng lớn vì nó đã được sử dụng bởi khoảng năm chục dự án mới tính riêng của quỹ Apache. Người ta cũng có thể kể đến các giấy phép Mozilla (MPL) và SUN (CDDL). Sự khác biệt của các giấy phép khác nhau này chỉ ở mức vài chi tiết.

Giấy phép GNU GPL

Giấy phép GNU GPL được đến 70 % chương trình nguồn mở sử dụng. Nhưng con số phần trăm này không phải là điều quan trọng nhất vì có một số phần mềm nguồn mở tiêu biểu lại dùng những giấy phép khác.

Giấy phép GNU GPL « *GNU General Public License* » đặc trưng chủ yếu bởi điều 2 của nó, điều nói về quyền sửa đổi chương trình và phân phối lại các sửa đổi này, tức là tạo ra các phiên bản dẫn xuất, với điều kiện là các dẫn xuất này vẫn phải nằm dưới cùng giấy phép GPL.

Đó là điều mà một số người gọi là đặc trưng *siêu vi* của giấy phép: nó được truyền cho các công trình dẫn xuất. Nhưng đúng hơn phải nói đó là sự đối ứng, hay là sự có đi có lại.

Đương nhiên do đó tất cả vấn đề nằm ở chỗ biết chính xác thế nào là một *công trình dẫn xuất* và nói *phân phối* nghĩa là gì? Mặc dù đã có rất nhiều bài báo nói về chủ đề này, nhưng những vùng tối vẫn còn. Thậm chí một số người còn cho rằng việc giữ lại một vài nghi ngờ cũng không phải là dở.

Chúng ta hãy cùng xem những gì đã rõ ràng.

« Công trình dẫn xuất » là gì?

Chắc chắn, nếu bạn lấy một đoạn mã nguồn của chương trình A, bạn sửa đổi các dòng mã hoặc thêm một số dòng để tạo ra một chương trình B, đó là một *công trình dẫn xuất*.

Cũng chắc chắn như vậy, nếu bạn gọi các hàm của chương trình A từ một chương trình B, qua việc liên kết hai chương trình (« *link* »), thì ở đây cũng thế, chương trình B là một *công trình dẫn xuất*. Sự liên kết giữa các chương trình có thể là tĩnh hoặc động, tức là liên kết chỉ diễn ra khi chạy. Đã có cuộc tranh cãi về việc liệu có thể xem một liên kết động là một công trình dẫn xuất hay không.

Thật vậy, trong các môi trường kỹ thuật hiện đại, tồn tại nhiều cách khác nhau để gọi các dịch vụ của một chương trình khác lời gọi hàm. Sự gọi các dịch vụ của một chương trình A thông qua các giao thức chuẩn trên mạng không kéo theo B là một *công trình dẫn xuất*. Nếu quả vậy thì một trình duyệt web gửi một truy vấn đến một máy chủ mà các chương trình CGI của nó dưới GPL, thì bản thân nó cũng bị buộc phải là GPL.

Trong thực tế, mọi người thường chấp nhận rằng một chương trình B được coi là *dẫn xuất* của chương trình A, nếu B *không thể vận hành một cách hữu ích mà không có A*, điều đó độc lập với các phương thức kỹ thuật của sự liên kết.

Vậy một chương trình sử dụng thí dụ một cơ sở dữ liệu MySQL, dưới giấy phép GPL, sẽ được coi thế nào? Nếu chương trình này, để gọi đến cơ sở dữ liệu, nó không sử dụng các thư viện dưới giấy phép GPL, mà chỉ gọi các dịch vụ của MySQL thông qua các giao thức chuẩn, thì nó sẽ không bị kéo theo phải là GPL. Nhưng nếu chương trình chỉ có thể chạy với một cơ sở dữ liệu MySQL, thì người ta sẽ có thể coi nó là một công trình dẫn xuất, bất chấp mọi thứ. Cần ghi chú rằng tài liệu FAQ (trả lời các câu hỏi thường gặp) của MySQL về chủ đề giấy phép đã bị phê phán vì đã đề hiểu lầm rằng mọi dạng sử dụng vào mục đích thương mại cần phải được đặt dưới giấy phép thương mại. Điều đó là sai.

« Phân phối » là gì?

Ở đây cũng vậy, có một số điều rõ ràng. Chắc chắn, nếu bạn thương mại hóa chương trình của bạn như là một phần mềm hoàn chỉnh, điều đó gọi là phân phối.

Ngược lại, sử dụng và triển khai một chương trình *trong nội bộ một cơ quan*, đó không phải là phân phối. Điều đó có nghĩa là một hãng có thể xây dựng một công trình dẫn xuất, và sử dụng nó trong nội bộ trên tất cả các máy trạm và máy chủ mà hãng thấy cần, không bắt buộc phải truyền bá mã nguồn của công trình. Đó là một điểm mấu chốt trong lĩnh vực kinh tế.

Một câu hỏi quan trọng khác là về quan hệ khách hàng – nhà cung cấp trong các nghề của tin học. Khi một nhà cung cấp dịch vụ như Smile xây dựng một ứng dụng sử dụng các thành phần dưới giấy phép GPL và chuyển giao ứng dụng này cho khách hàng của mình, nhà cung cấp phải cung cấp đầy đủ toàn bộ tập hợp mã nguồn, cả những phần mới thêm vào. Nghĩa vụ phân phối mã nguồn chỉ liên quan đến những người nhận chương trình, ở đây là khách hàng. Không bắt buộc phải công bố trên một nơi công cộng.

Ngoài ra, khách hàng có thể, hoặc giữ chương trình cho riêng mình (nghĩa là trong nội bộ tổ chức), hoặc là phân phối nó, nhưng khi đó bắt buộc phải dưới giấy phép GPL.

Chúng ta cũng cần lưu ý rằng sử dụng công trình dẫn xuất dưới dạng dịch vụ trực tuyến (*Software as a Service*), kể cả dịch vụ thương mại, không phải là phân phối. Điều Google đang làm là một ví dụ. Về điểm này, xin xem phần đề cập giấy phép AGPL dưới đây.

Tư tưởng của GPL

Vượt lên trên các từ ngữ, *tư tưởng* của giấy phép GPL là, với tư cách là tác giả hay chủ sở hữu của một chương trình, tôi cho bạn quyền sử dụng nó và sử dụng mã nguồn của nó, *với điều kiện bạn*

cũng sẽ làm như vậy. Tóm lại, đó là có đi - có lại.

Giấy phép GPL có hệ quả là nó chia thế giới thành hai « phe »: phe GPL và phần còn lại của thế giới. Nếu bạn ở phía GPL, thì bạn có thể tiếp cận không hạn chế mọi di sản nguồn mở dưới GPL. Còn nếu bạn ở phe kia, tức là bạn không muốn phân phối mã nguồn của bạn cho những người khác với cùng độ tự do mà bạn đã được nhận, thì bạn không thể tận dụng kho tàng đó.

Vì thế, người có thể gọi đó là nguyên tắc *có đi-có lại*, điều mà những phê phán giấy phép này gọi nó là khía cạnh siêu vi.

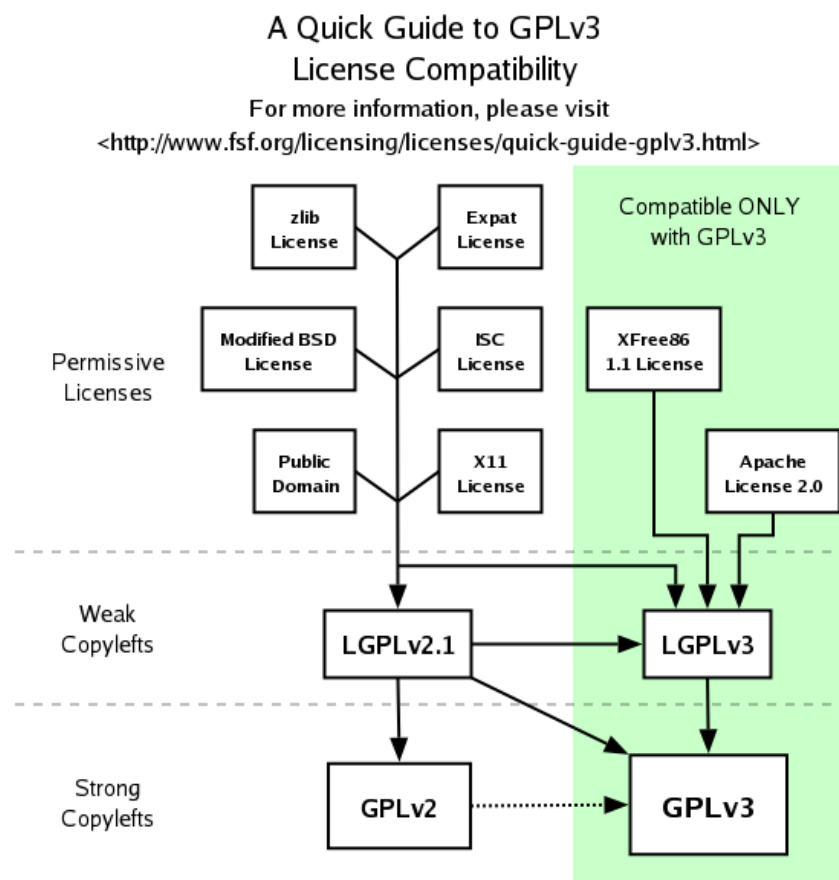
Sự tương thích của các giấy phép

Vấn đề tương thích của các giấy phép là một vấn đề hàng đầu. Nếu một chương trình A đặt dưới giấy phép L_A và một chương trình B dưới giấy phép L_B , thì có thể xây dựng một chương trình C sử dụng đồng thời A và B không? Chương trình C sẽ thừa kế các yêu cầu của L_A và của L_B , và nếu chẳng may có các mâu thuẫn giữa các yêu cầu này, nếu không thể tôn trọng cả hai đồng thời, thì sẽ phải từ bỏ việc sử dụng A và B.

Vì vai trò thống lĩnh của giấy phép GPL trong nguồn mở, câu hỏi chính ở đây là sự tương thích với giấy phép GPL. Một chương trình nguồn mở, nếu có một giấy phép không tương thích GPL, thì việc sử dụng nó sẽ bị hạn chế hơn.

Trong số các giấy phép tương thích, ta có thể kể đến các giấy phép BSD, MIT, hay là giấy phép Apache (tương thích GPL-v3). Trong số các giấy phép không tương thích, có thể kể đến SUN CDDL, Eclipse, Mozilla.

Hình vẽ sau đây được lấy từ website gnu.org. Các mũi tên chỉ thị sự tương thích của các giấy phép.



Copyright 2007 Free Software Foundation, Inc. Verbatim copying and distribution of this entire image are permitted worldwide, without royalty, in any medium, provided this notice is preserved.

Cần lưu ý rằng một giấy phép L_A thuộc loại copyleft, và hầu như giống hệt GPL, nhưng lại mang

một tên khác, sẽ không tương thích GPL, vì một công trình dẫn xuất không thể đồng thời GPL và L_A. Đó là trường hợp, ví dụ, của giấy phép « tương hỗ » được Microsoft đưa ra gần đây, có tên gọi là MsRL.

LGPL

Giấy phép LGPL rất gần với GPL, nhưng cho phép gọi các hàm từ một chương trình khác, không cần các chương trình sử dụng chương trình dưới LGPL này phải là nguồn mở. Giấy phép này vì thế đặc biệt thích hợp với các thư viện hàm, bản chất là để cho các chương trình khác nhau gọi tới, không đặt ra các điều kiện quá mạnh cho các chương trình này.

Ban đầu LGPL có nghĩa là *Library* GPL, nhưng tên gọi sau đã được đổi thành *Lesser* GPL (yếu hơn GPL), bởi vì Richard Stallman muốn hạn chế tối đa sự tương ứng « thư viện (*Library*) = LGPL » và cho phép trừ tính có cả các thư viện dưới GPL. Giấy phép LGPL là một sự thỏa hiệp giữa quyết tâm mạnh mẽ cổ động cho nguồn mở, và tránh sự thu hồi nó để phục vụ cho các phần mềm tư hữu, và mặt khác ý muốn đem lại dịch vụ lớn nhất thông qua sự sử dụng rộng rãi nhất.

Cần lưu ý rằng đằng sau sự cho phép liên kết (*linker*), giấy phép LGPL còn đưa vào một số điều kiện khá tế nhị trên chương trình được liên kết. Ví thế, một số lập trình viên đã thích phân phối dưới giấy phép GPL kèm theo một phụ lục *special linking exception*, cho phép gọi hàm.

Giấy phép GPL *with special linking exception* dễ đọc hơn và cho phép nhiều hơn, là giấy phép mà SUN đã chọn cho phần mềm JDK của mình.

GPL-v3

Phiên bản 3 của giấy phép GPL đã được hoàn thành trong năm 2007, và đang được triển khai dần.

Nó nhằm cải thiện phiên bản v2, và thích ứng nó vào một bối cảnh đã vận động, trên các điểm sau đây:

- Một sự định nghĩa pháp lý chặt chẽ hơn các từ ngữ, làm cho ít hơn các thuyết minh sai;
- Cấm ngăn cản, nhờ các thiết bị vật lý, hoặc bằng cách không cung cấp thông tin cần thiết, sự thiết đặt của phần mềm được sửa đổi trên phần cứng đích của nó. Đó là điều đã được gọi là sự tivo hóa, bắt nguồn từ tên của hãng Tivo, nhà sản xuất máy ghi âm đã nhờ vào thủ thuật này.
- Trường hợp đã có lệnh cấm, tại nhiều nước, vượt qua các cơ chế quản lý bản quyền số (DRM – *Digital Right Management*). Một công trình dẫn xuất một phần mềm GPL-v3 không thể đề cập sự cấm đoán này. Nghĩa là, không cấm viết một chương trình làm DRM sử dụng các thành phần GPL-v3, nhưng mà cấm sự cấm đoán vượt qua rào cản đó.
- Một sự bảo vệ chống lại các yêu sách của các văn bằng sáng chế phần mềm: người phát tán mã nguồn của mình dưới giấy phép GPL-v3 cho tất cả các quyền sử dụng do giấy phép cho phép và cấm sự truy xét người sử dụng về chuyện văn bằng sáng chế phần mềm.
- Khả năng cho phép thêm một số hạn chế riêng cho giấy phép, trong một số khả năng giới hạn, điều này thêm một chút co dãn trong các vấn đề tương thích các giấy phép.

AGPL (Affero)

Như chúng ta đã nói ở trên, không ai cấm một người lấy một chương trình dưới giấy phép GPL, xây dựng trên nền này một chương trình dẫn xuất, và sau đó sử dụng chương trình này cho nhu cầu riêng của anh ta, bao gồm cả việc triển khai trong nội bộ tổ chức của anh ta, và không nhất thiết phải phát tán mã nguồn. Cũng như vậy, không hề cấm việc đưa ra một dịch vụ truy cập được trên Internet, cũng được xây dựng bằng chương trình dẫn xuất này, không cần phải phát tán mã nguồn, vì sự sử dụng này không phải là sự phân phối.

Tuy nhiên, với sự phát triển mạnh mẽ các chào hàng dịch vụ thuê máy chủ, dạng *Software as a Service* (SaaS), kiểu sử dụng này có nguy cơ mở rộng. Nhưng hãy nghĩ kỹ mà xem, cho chương trình truy cập trực tiếp được từ những người sử dụng cuối qua Internet, rõ ràng là một cách cấm truy cập mã nguồn, mà lại đang làm một sự khai thác thường là thương mại, có hơi hướng của một sự phân phối.

Để chống lại nguy cơ trốn tránh trách nhiệm này, công ty Affero đã tạo ra, hợp tác cùng với FSF, giấy phép AGPL hoặc Affero GPL. Nó giống hệt GPL, nhưng được thêm một điều khoản rằng nếu chương trình ban đầu cho phép một truy cập từ mạng và đã phát tán mã nguồn của nó qua mạng, thì các chương trình dẫn xuất cũng phải làm như vậy.

Đây là một biện pháp nền tảng, và dường như nó có xu hướng sẽ được phổ biến rộng rãi trong tương lai.

Sở hữu trí tuệ và văn bằng sáng chế

Thuật ngữ Sở hữu trí tuệ theo nghĩa chung qui chiếu đến tất cả các khía cạnh pháp lý liên quan đến sự sở hữu của các tài sản phi vật chất và do trí thức tạo ra.

Bản quyền trên một chương trình là một khái niệm khá rõ ràng. Ngay cả nếu chúng ta đã dẫn ra trước đây những sự thiếu chính xác có thể trong khái niệm công trình dẫn xuất, thì có một điều chắc chắn: nếu một nhà lập trình ngồi trước bàn phím và viết ra mã nguồn mà bộ óc anh ta nghĩ ra, anh ta không đang vi phạm bất kỳ bản quyền nào. Chính anh ta, hoặc ông chủ lao động của anh ta, là những người giữ quyền tác giả của mã nguồn này.

Nói một cách khác: người ta biết khi nào người ta vi phạm một quyền tác giả.

Ngược lại, các văn bằng sáng chế phần mềm có thể bị vi phạm mà không biết. Một văn bằng sáng chế phần mềm có thể đề cập đến sự sử dụng trong một chương trình một thuật toán hay một thủ pháp. Ở Hoa Kỳ, nơi được phép đăng ký các văn bằng sáng chế phần mềm, các hãng lớn đăng ký hàng ngàn văn bằng, trong đó rất nhiều chỉ đơn thuần là một thủ pháp thông thường, hoặc ai cũng đều biết. Một số trong số đó có thể không được chấp thuận, nhưng người ta sẽ chỉ biết điều đó vào cuối cùng nếu như có sự tranh chấp, tức là sau một phiên tòa cực kỳ tốn kém.

Vì thế, chúng ta hiểu rằng các văn bằng sáng chế phần mềm là một mối nguy hiểm lớn cho ngành công nghiệp tin học nói chung. Nó có tác dụng làm xơ cứng thị trường, để bảo trì độc quyền của các hãng phần mềm lớn, những kẻ duy nhất có thể chịu được các chi phí pháp lý theo kiểu Mỹ. Và giữa họ, dù điều gì xảy ra, cũng không tấn công lẫn nhau.

Đôi khi người ra cho rằng các phần mềm nguồn mở sẽ phải lo ngại nhiều về văn bằng sáng chế, đơn giản vì mã nguồn của chúng là công khai. Nếu chúng sử dụng những thuật toán đã đăng ký văn bằng sáng chế, thì điều đó sẽ dễ dàng bị phát hiện. Ngược lại, nếu Microsoft hay Oracle sử dụng một mã nguồn vi phạm một văn bằng, hay thậm chí một bản quyền, thì sẽ cực kỳ khó khăn để chứng minh điều đó.

May mắn thay, các văn bằng sáng chế vẫn chưa có đất sống ở châu Âu, nhưng mối hiểm họa là lớn và những kẻ bảo vệ nó chưa hạ vũ khí.

(Kỳ sau: Hỗ trợ phần mềm nguồn mở)