

Why Software Should Be Free

By Richard Stallman

Introduction

I asked the question of whether it was ethical to make software proprietary.

We must ask what we, as programmers together, can do for the freedom of mankind. We must ask what we ought to do, not just what is profitable.

The law should conform to ethics, not the other way around.

We should perform a cost-benefit analysis on behalf of society as a whole, taking account of individual freedom as well as production of material goods.

My conclusion is that I and other programmers have a duty to encourage others to share, redistribute, study and improve the software we write: in other words, to write *free* software. (The word “free” here refers to freedom, not to price.)

How owners try to justify their demands

I put my sweat, my heart, and my soul into this program. It comes from *me*, it's *mine*!

Consider the great artists and artisans of medieval times, who didn't even sign their names to their work. To them, the name of the artist was not important.

This view prevailed for hundreds of years.

“I want to get rich, and if you don't allow me to get rich by programming, then I won't program.”

The fallacy here is in comparing only two outcomes: proprietary software vs. no software. These are not the only alternatives.

To formulate the choice as between proprietary software vs. no software is to presuppose this decision. That is begging the question.

The argument against having owners

“Should development of software be linked with having owners to restrict the use of it?”

We have to judge the effect on society of each of those two activates *independently*.

To determine the effect of restricting use, we need to compare the value to society of a restricted (i.e., proprietary) program with that of the same program, available to everyone.

The harm done by obstructing software

Restrictions on the distribution and modification of the program cannot facilitate its use.

Three different levels of material harm come from obstruction:

Fewer people use the program.

None of the users can adapt or fix the program.

Other developers cannot learn from the program, or base new work on it.

Obstruction of use

A copy of a program has nearly zero marginal cost (and you can pay this cost by doing the work yourself), so in a free market, it would have nearly zero price.

Central copying of software is simply more work than user copying.

Damaging social cohesion

If you want to use a program and your neighbor wants to use the program, then in ethical concern for your neighbor, you should want *both* of you to have it. You shouldn't be satisfied with a solution where you get it and the neighbor does not.

Many users decide to ignore the licenses and laws, and share programs anyway.

They know that they must break the rules in order to be good neighbors.

That is also a kind of psychosocial harm, which one can escape by deciding that these licenses and laws have no moral force.

Obstruction of custom adaptation

The second level of material harm is the inability to adapt program.

Usually the source code for a proprietary program is kept secret by the owner, lest anybody else learn something from it. This means that only the owner can change the program.

Most good programmers I have met have experienced the frustration of using a program whose deficiencies they were forbidden to correct.

Imagine what it would be like if recipes were hoarded in the same fashion as software. You'd say, "How dare you insult my recipe, the child of my brain and my palate, by trying to tamper with it? You don't have the judgment to change my recipe and make it work right!"

Obstruction of further advances

The existence of owners prevents evolution, making it necessary to start from scratch when developing a program.

In any kind of intellectual field, progress is built by standing on the shoulders of others. That's no longer generally allowed in the software field – you can only stand on the shoulders of the other people *in your own company*.

WHY PEOPLE WILL DEVELOP SOFTWARE

Programming is fun

When one worker gets rich, others demand the same opportunity. Soon you will find that no one is willing to work in the field without a clear shot at getting rich.

Funding free software

How can we pay programmers?

Hardware manufacturers must support software development even if they cannot restrict its use.

It is common today for university researchers to get grants to develop a system, develop it nearly to the point of completion and call that “finished”, and then start companies where they really finish the project and make it usable.

Programmers can also make their living as I have for six years: by making customer improvements to free software.

The Free Software Foundation is a charity, and its income is spent on hiring as many programmers as possible.

Because the Foundation is a charity, programmers often work for the Foundation for half of what they could make elsewhere.

Most of all, they do it because programming is fun.

This confirms that programming is among the most fascinating of all fields, along with music and art. We won't have to fear that no one will want to program.

What do users owe to developers?

Developers of free software are contributing to the users' activities, and it is both fair and in the long-term interest of the users to give them funds to continue.

This does not apply to proprietary software, since obstructionism deserves a punishment rather than a reward.

Eventually the users will learn to support developers without coercion, just as they have learned to support public radio and television stations.

What is “software productivity”?

Free software would require far fewer programmers to satisfy the demand, because of increased software productivity as all levels:

- Wider use of each program that is developed
- The ability to adapt existing programs for customization instead of starting from scratch
- Better education of programmers
- The elimination of duplicate development effort

Overall productivity is what society would like to improve, and most straightforward way to do this is to eliminate the artificial obstacles to cooperation which reduce it.

Is competition inevitable?

Constructive competition is enough competition to motivate people to great efforts.

Competition becomes combat when the competitors begin trying to impede each other instead of advancing themselves

Proprietary software is harmful, not because it is a form of competition, but because it is a form of combat among the citizens of our society.

Our society’s resources are being squandered on economic civil war.

Conclusion

Software hoarding is one form of our general willingness to disregard the welfare of society for personal gain.