

Software Ownership: A Natural Right or Not?

The extent of ownership over computer software is often argued and has varying legislation in different nations globally. Therefore, this paper will only critically analyze the natural rights aspect of the argument at hand. Natural rights, by definition, do not depend on the laws or beliefs of any form of government; instead, they are rights tied with the concept of natural laws that are not created by society or court judges. Essentially, natural laws are based on the intrinsic nature of humans such as to deem things right or wrong. The focus will be on the natural right of property. This paper will first delve into explaining Locke's labor theory of property, the application of said theory to owning computer software, and a natural rights argument against extending property rights to software. The paper will then conclude with my personal position on the matter.

John Locke's labor theory of property pertains to the subject of natural property rights for humans. The fundamental component of his theory regards the connection between property and labor. The basis of Locke's theory is guided by the notion that God created this world and gave it to all "men in common" for the reason of using it to their "best advantage" (§26). The theory puts a big emphasis on the idea that Earth itself and Earth's creatures that are inferior to humans are common to all people, but every individual human person is owned by themselves – oneself is one's property. Locke is adamant in the fact that nobody in nature has the ownership over another person, only oneself can have that right of property. Therefore, an individual's labor is considered to be the individual's property. This leads into the core of Locke's labor theory of property which is when one changes the state that nature provided for some object through mixing one's own labor with the object, the right of others over that object is removed. So, because one mixed their labor (which is one's property) with a natural object common among all people, that object is considered to be the property of that individual.

Prior to diving into an argument of Locke's theory applying to software property rights, what computer software fundamentally is should be defined. Generally, computer software is a created collection of instructions that guides computer functionality that can also typically be distributed among different computers. Software can be developed by either one individual or many. An issue then arises over the dispute of ownership because software can be mimicked, duplicated, and/or modified by others. Therefore, the area for dispute lies in the contention for the right of ownership over the software.

First, I will consider the application of Locke's labor theory of property within the area of computer software ownership. It can be argued that because any software's original source code was produced by the labor of an individual, the resulting software is owned by that individual, and any copies of that software has the original source code created by that same individual, so the rights of ownership remain to the source code author(s). Although software can be written by a collective of individuals, for the sake of argument, assume that the collective is still one single entity – negligibly through a contractual agreement. To better illustrate Locke's theory, I will

apply it to a real-world computer software example. Consider a scenario where the web browser software Google Chrome is distributed in nature (disregarding governmental legislation). The theory's application would then be represented as follows: although there are over three billion users of the web browsing software, since the entity Google labored the software's source code, the right of ownership belongs to Google, not the users with a copy of the software (Balakumar). In this scenario, the intellectual property of uniformly creating the entire software package belongs to the original creator of the source code because the creator labored existing (or new) algorithms available to any person into the single software package. Essentially, the end-users have access to the software but cannot claim ownership of Google Chrome's intellectual property, the source code, under Locke's theory since end-users did not labor the functions and algorithms compiled together into the source code for the browser software.

The mentioned application of Lockean theory to software ownership is still problematic because some aspects of the theory are not fully clear. Some of these noteworthy flaws within Locke's theory of property were highlighted by Robert Nozick. His main counterargument is in the form of a question regarding the extent to which the premise of mixing labor to natural objects applies. He asks if mixing a juice he labored (therefore owned) into the sea would yield him acquiring ownership of the entire sea or yield him naively dissipating the juice away. This question he raises exemplifies a grey area of acquisition of property within Locke's theory. Yet, an attempt to clarify this grey area would be to reference a point Locke makes that the object which labor has been applied to is one's property "where there is enough, and as good, left in common for others" (Locke §27). Generally meaning that others should not be left "worse off" by one acquiring ownership of an object. Basically, this acts as a condition (proviso) of the labor theory of property that vaguely answers the question Nozick raises. Nozick's scenario regarding the juice would likely fail this condition since one taking ownership of the entire sea through arbitrary labor being applied to the entire sea would imply that there would be not enough sea left in common for others – so, the juice ownership would be naively dissipated. However, Nozick does indeed reference this proviso from Locke, but he outlines that it is not clear if Locke meant for another to not be left "worse off" to be a stringent or weaker requirement of the proviso (Nozick). Using the previous Google Chrome example, if an individual claims ownership over source code similar to the existing source code, quantifying if he/she may or may not leave the original creator "worse off" is impossible without an authoritative body judging – which is a glaring issue. So, if Locke's theory is to be properly applied to software property rights, the provisos of his theory need more clarification for the requirements surrounding acquisition.

The clarity flaws within Locke's theory, which were just outlined, lead to arguments that natural rights should not extend into the realm of software property rights. Assume that when applying natural property rights to software acting as intellectual property, its copies being intellectual property of the source owner is implied, because the copies depend on the original software's source code. The consequence is that a possible argument may arise where some natural rights *may actually be lost* when applying natural property rights to software in this manner. Two possible cases in which natural rights are lost when extending property rights to software will be highlighted. The first case indicates the loss of labor; the second case indicates

the infringement of free thought. Note: for both cases, assume that the natural rights of an individual include private property of software and labor, similar to Lockean theory; this via the logical reasoning that labor on any universally available physical object or any universally available intellectual property yields private acquisition.

For the first case, recall the scenario of the web browser software Google Chrome in a natural law sense. Assume that an individual applies labor to form software basically identical to Chrome. Google can claim that because the copies depend on the source code, the entire software and its copies are entirely owned by Google. Meaning, one could argue under the applied natural rights described above that the new software created would wrong the creators of Chrome since they hold the private property rights of the software. Then, the individual who naively created the ‘copy’ would then have his/her labor be completely nullified by the extension of property rights to the software when confronted by Google. This is a striking issue because labor could essentially become void due to the aforementioned condition when there is not enough intellectual property universally available after Google acquires ownership over the source code. Generally, the extension of natural rights would cause individuals to lose the right to their own labor when laboring with specific intellectual property that is already owned. To reiterate, the key issue is that the bounds to which there is enough universally available intellectual property are very difficult to quantify – especially in regard to software. Since the subject is based on natural rights, no governing body or authority can decide the limits of these bounds. Therefore, the right to property over one’s own labor itself can potentially be lost. The second case of possibly losing the natural right of free thought bounces off the same example scenario. So additionally, when we assume that Google *does indeed* get property rights over the browser and its copies, then no individual would be permitted to produce software that has similar functionality to the software based on the source code originally created – losing the ability to transform thought into reality. Hence, extending natural property rights to software ownership can interfere with the natural right of one’s own labor and freedom of thought.

Briefly, a possible counterargument to the cases above will be explained. Some might argue that a distinction between source code and the software’s functionality as a whole is necessary to be made for both cases. It can be said that if natural rights are extended only to source code, rather than the entire software package, then the natural rights of self-labor and free thought would not be infringed upon. The issue with this counterargument, though, is that natural rights exist without authoritative bodies that can determine the differentiation between software and source code in nature. Additionally, determining the scope for differentiating between algorithms and software, or basically any other intermediary software component, would follow the same issue. Also, note that in the example, Google claimed ownership of the software’s copies since the copies depend on the original labored source code. Then, through chain reasoning, Google could also claim that owning the source code implies ownership of every intermediary component of the software.

Finally, my overall stance on the subject is a utilitarian approach where natural rights should not be extended to software as a whole when mixing labor. Consequently, this would mean that software’s subcomponents such as source code, assembly code, etc. would not be

protected by natural rights to property because it has already been noted that in nature, these cannot be differentiated consistently. I will return to Locke's labor theory of property and say that natural rights are inapplicable to software. The problem surrounds the idea mentioned in case one of the described argument against natural rights' application. The ability for one individual to privately possess cloneable intellectual property such as software is not usually beneficial to the entire populace. This is because one could claim that they then own every copy that depends on the source code written. Furthermore, Locke's theory is imprecise because it can form scenarios where individuals can unknowingly violate those already owning software. Together, this can incentivize the idea of rushing to claim ownership of software and prevent individuals from improving upon the existing software which halts beneficial technological growth.

However, counterexamples to my stance exist, as Locke himself included that the prime goal of natural ownership is to "better" humanity. Plus, Locke mentions that letting property "rot" is a disservice to natural intent (§37). Yet, I refute by saying that Locke is being too vague once again. I respond by raising the question: in nature, who is to decide if the ownership of specific software betters the life of oneself and others? I must then conclude by affirming that the range to which humanity is "better[ed]" varies depending on the natural viewpoint that is different among most individuals. Therefore, collective ownership and betterment of software might resolve any discrepancies over natural means.

Additional Source:

Balakumar, K. “Google Chrome Browser now has More than 3 billion Users.”

TechRadar, 26 May 2021, <https://www.techradar.com/news/google-chrome-browser-now-has-more-than-3-billion-users>. Accessed 12 Dec. 2021.

[Source for number of chrome users globally.]