5-2019

The Crusader and the Dictator: An Exploration of Ideology and Neurodivergence in Contemporary Technology Practice

David J. Williams
The Graduate Center, City University of New York

THE CRUSADER AND THE DICTATOR: AN EXPLORATION OF IDEOLOGY AND NEURODIVERGENCE IN CONTEMPORARY TECHNOLOGY PRACTICE

by

DAVID J. WILLIAMS

A master's thesis submitted to the Graduate Faculty in Liberal Studies in partial fulfillment of the requirements for the degree of Master of Arts, The City University of New York

2019

The Crusader and the Dictator: An Exploration of Ideology and Neurodivergence in Contemporary Technology Practice

by

David J. Williams

This manuscript has been read and accepted for the Graduate Faculty in Liberal Studies in satisfaction of the thesis requirement for the degree of Master of Arts.

Date	Christopher Schmidt Thesis Advisor
Date	Elizabeth Macaulay-Lewis Executive Officer

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

The Crusader and the Dictator: An Exploration of Ideology and Neurodivergence in Contemporary Technology Practice

by

David J. Williams

Advisor: Christopher Schmidt

A common theme in public discourse is the recognition that technology in general, and digital technology specifically, has an enormous impact on the everyday lives of people from all walks of modern life, in almost every corner of the globe. This thesis interrogates the connection between neurodivergence—the presence of neurological variations considered outside the cognitive norm— and individualistic ideology within the information technology industries. Through the biographies, substantial record of activities, public statements, and writings surrounding two influential figures in the contemporary practice of computer science, Richard Stallman and Linus Torvalds, it conducts an investigation into this convergence and its resulting impact on the surrounding culture.

ACKNOWLEDGMENTS

I wish to thank my advisor, **Christopher Schmidt**, whose thoughtful insights and steady guidance ensured that my research, initially a sprawling mass of semi-formed, disconnected impressions, evolved into a (mostly) focused and coherent whole.

My thanks also to **George Fragopoulos** and **Justin Rogers-Cooper**, who went above and beyond the call of duty, offering essential assistance while opening my eyes to a wide range of research possibilities.

James Mellone has been both a friend and consistent source of encouragement and sage counsel over the years, without whom I would undoubtedly not have arrived at this point in my academic career.

Finally, I want to thank my partner, **Laura Donovan**, whose tireless support and excellent editorial skills are invaluable resources in all of my endeavors, personal and professional. I can only aspire to match her generosity of spirit, and dedicate this thesis to her with love and gratitude.

Contents

Introduction

Nerd. Boffin, as they say in the United Kingdom. Geek—arguably a different category altogether, according to some.[1] Signifiers emerged in popular culture, shorthand terms for a kind of scientific and technological genius. Brilliant within a specific domain, but otherwise inept or discomfiting in social environments, if not actively unpleasant, this figure, predominantly (but not exclusively) male, occurred often enough to suggest not just a stereotype but some form of archetype, constantly recurring throughout history but seemingly apparent in a more concentrated form in recent times.

Examples of the trope include the absent-minded professor, or, inversely, the mad scientist. Jerry Lewis' "Julius Kelp" in *The Nutty Professor* (the inspiration for *The Simpsons*' "Professor Frink"), or the eccentric "Sheldon Cooper," from the television comedy *The Big Bang Theory*. Although extreme manifestations of genius are no more common in the medical industry than in other walks of life, an unusual concentration of doctors matching this description exist in the popular imagination, possibly as an expression of collective wish-fulfillment. With mysterious symptoms and your life on the line, who wouldn't tolerate the challenging presence of Gregory House from the eponymous medical drama *House*? Recent additions to the canon include the high-functioning autist Shaun Murphy from *The Good Doctor*, along with his Korean antecedent Park Shi-on. Temperance "Bones" Brennan from *Bones*, Peter Gregory from the Mike Judge comedy *Silicon Valley* (modeled directly after technology investor and PayPal co-founder Peter Thiel), and of course *Star Trek*'s Mister Spock (representing an entire species of emotionally limited geniuses),

[1] The two terms are often used interchangeably, but there is some debate over which is more appropriate when applied to technology professionals. The label "geek" was originally given to circus performers engaged in outrageous and disturbing physical stunts, including biting the heads off of live chickens, but became synonymous with an extreme degree of enthusiasm for any subject or activity. "Nerd," on the other hand, is widely attributed to Theodor Geisel, a.k.a Dr. Suess, first appearing his book *If I Ran the Zoo*, and specifically includes an interest in or aptitude for technology, according to the Oxford English Dictionary, "Nerd, n."

all follow a consistent pattern. The figurative grandfather of them all, Arthur Conan Doyle's Sherlock Holmes, is a source of awe and admiration to both his fellow literary personae and to readers, but the discomfort and alienation surrounding Holmes and each of these characters is a narrative constant.

The eccentric, socially-challenged genius makes for a popular fictional device, but do parallels exist among historical figures? Speculation abounds on the personal characteristics of several prominent examples, among them Sir Isaac Newton. Visionary inventor Nikola Tesla is considered the direct inspiration for the "mad scientist" antagonist in Max Fleisher's very first Superman animation.[2] Could we include pioneering mathematician Alan Turing in this group? Certainly the fictional character of "Alan Turing," as portrayed by Benedict Cumberbatch in the film *The Imitation Game*, fits the pattern.[3] X-ray crystallographer Rosalind Franklin, whose work is widely regarded as the unacknowledged key to Watson and Crick's unlocking of the molecular structure of DNA, is a candidate. Animal behaviorist Temple Grandin, on the other hand, is a living, self-professed example of the relationship between developmental disability and academic genius.

A common thread appears to link these individuals, both fictional and actual, connecting intellectual brilliance with characteristic patterns of thought, inclination, and behavior. My investigation into this perceived correspondence is intended as an expression of the intersection between cultural studies and the history of science, disciplines fundamentally interdisciplinary in nature and well-suited to placing cultural practices within an expansive context. Intended for

[2] Laqueur, *The New Terrorism: Fanaticism and the Arms of Mass Destruction*, 265.

[3] A recent critical analysis of Turing's biography, mathematician Andrew Hodges' *Alan Turing: The Enigma* (which also served as a source for the biographical film), applied the six diagnostic criteria developed by neuropsychiatrist Lars Christopher Gillberg to anecdotes and statements describing Turing's behavior, and concluded that he satisfied all of them, in O'Connell and Michael Fitzgerald, "Did Alan Turing Have Asperger's Syndrome?"

scholars of culture, history, politics, and philosophy, this thesis applies this interdisciplinary interpretation to a series of recent, defining moments in the history of technology, and the minds responsible for effecting them. The resulting snapshot, focused on the personal and professional accomplishments of two prominent software creators, offers an alternative view of the current techno-political landscape revealing a path to approaching future developments.

Dawn of the Different

A popular axiom widely attributed to Winston Churchill suggests that history is written by the victors. Society, it can be argued, is in turn defined by the sociable, and the reputation of the so-called nerd was, until relatively recently, historically unenviable. Often limited to the society of their peers (a not altogether undesirable fate in some eyes), the nerd's contributions to the structural and economic well-being of a community resulted in acceptance, if not always appreciation. But as societies rely more and more upon technological and scientific achievements to both improve their standards of living and preserve themselves in the face of external competition, the nerd's social standing and corresponding economic power improves greatly. An argument could even be made that in the contemporary world the nerd is enviable, with many ranking among our most influential public figures. It is most likely not an accident that an American documentary television series on the history of the personal computer was titled *Triumph of the Nerds*. Bill Gates, co-founder of Microsoft, is apocryphally credited with saying, "Be nice to nerds. Chances are you'll end up working for one."[4]

The implication of these trends is that few disciplines have been as effective at easing those

[4] Gates was popularly but falsely credited with this statement, presented as one of a series of rules for achieving success in life offered in a High School graduation speech. The most likely source is an editorial published in the September 19, 1996 issue of the *San Diego Union-Tribune* by journalist Charles Jay Sykes, reproduced in Sykes, "Some Rules Kids Won't Learn in School."

society deems nerd-like into social and economic prominence as the engineering and information sciences. In the process, new proper nouns were introduced into the dictionary, following a pattern that suggests a gradually changing status: phreaker, hacker, cracker, and, more recently, maker.

Historians of science are dedicated to illuminating those gradual series of social and political changes that enabled progress toward understanding the physical world, but the development of a shared language for describing, testing, and revealing these achievements is perhaps the most significant advance of all. From the Babylonians, through Pythagoras, to Archimedes, Newton, Leibniz, and the contemporary sciences, that language is mathematics. Galileo Galilei famously explained to his critics:

> The universe cannot be read until we have learned the language and become familiar with the characters in which it is written. It is written in mathematical language[.]"[5]

In this age of information, mathematics reaches new heights of complexity and abstraction, transcending questions of earthly and celestial mechanics as it touches upon philosophy. One recent manifestation of these conceptual investigations, evolving in parallel with advances in physics and materials engineering, is computer science.

Studying the values and characteristics associated with those described as both genius and nerd, I hope to gain a meaningful insight into their impact on society and the forces that shape them. To properly contextualize their ascent, some historical background is in order. The ubiquitous presence of digital technology in contemporary society suggests some corresponding familiarity with its origins and nature. But, not unlike the observation that many people drive automobiles but far fewer can repair them, the nature of computer science is often misunderstood and confused. Commonly regarded as a vocational field designed to provide instruction and

[5] Galilei, *Il Saggiatore*, 171.

training in software programming, in actual practice the computing and information sciences study theories that form the basis for the design and use of computers. More specifically, computer scientists conduct research in the scientific and practical approaches to computation, and its application to the systematic study of the feasibility, structure, expression, and mechanization of those methodical procedures, or *algorithms*, that underlie the acquisition, representation, storage, communication of, and access to, information.[6] This overarching description covers everything from the implementation of logical procedures in hardware to designing special purpose languages for solving specific problem domains. Many research areas in the field are sufficiently theoretical as to obviate the need for physically touching a computer keyboard when pursuing them, while others, such as Artificial Intelligence, involve orders of complexity so great they can surpass expert description and classification.[7]

Machines for calculating numerical tasks, with their unambiguous problem solving specifications and recipes for performing calculations, have existed since antiquity. Distinguishing between mechanical calculating devices and general-purpose, "infinitely programmable" machines, capable of applying random algorithms to their operations, defines the historical emergence of the computer age. Nineteenth century English mathematician Charles Babbage is credited as the "father" of the computer for his design of an analytical engine in 1837. Soon after, in 1843, Augusta Ada King, daughter of the poet George Gordon Lord Byron and Countess Lovelace, designed an algorithm for calculating the Bernoulli numbers—the sequence of digits required for determining the sum of all the powers of a positive integer—specifically for application on Babbage's engine, making her widely regarded as the first practitioner, or programmer, of what

[6] A concise, if somewhat general definition of Computer Science can be found in Butterfield and Ngondi, "Computer Science."

[7] Hutson, "AI Researchers Allege That Machine Learning Is Alchemy."

would become known as applied computer science. It was not until a century later that the next significant development arrived, through the theoretical work of Alan Turing, who, along with American Alonzo Church, proved the limits of what problems could actually be "decided" using algorithms operating within a hypothetical "universal" computing machine. In the process he outlined what would become the foundation for both the design of modern computers and the programs they could process. Following the Second World War, the term "computer science," or CS, was formally proposed to describe this new discipline, with Purdue University creating its first academic department in 1962.[8]

Having origins in mathematics, engineering, and the philosophical foundations of logic, researchers in the field of computer science often find themselves favoring one component element over another. Noting its relative novelty and interdisciplinary qualities, the question of what CS actually "is" has been the subject of much critical examination. Could it be considered a "natural" science, like chemistry and physics, and do its seemingly unrelated research areas suggest a single, unified discipline? Reflecting on these origins, pioneering computer scientists Edsger Dijkstra and Donald Knuth argued that, although not a physical science, computer science is built on a different and distinct conceptual foundation, modeling finite, dynamic processes using information-centered algorithms. The incorporation of empirical and experimental research qualifies it as a fundamentally scientific pursuit, distinct from the non-systematic approach characterized as "hacking."[9] But for those who hold a "big tent" view of the field, these non-academic computing professionals qualify equally as computer scientists. For others, including programmer and venture capitalist Paul Graham, the distinction is culturally important: "In the software business there is an ongoing struggle between the pointy-headed academics, and another equally formidable force, the

[8] Knuth, "George Forsythe and the Development of Computer Science," 721.
[9] Tedre, "Computing as a Science: A Survey of Competing Viewpoints," 364.

pointy-haired bosses," he writes, referencing the caricature made famous by Scott Adams' *Dilbert* comics.[10] The disparaging description "pointy-headed" predates computer science, but for an entrepreneur like Graham its application implies a newer, different form of disapproval. His distinction between a researcher and a practitioner, between theory and praxis, is not unlike Karl Marx's conclusion on the true value of philosophy: not to interpret the world, but to change it.[11] To some, the true value of programming is not to create a hack, but to capitalize on it.

In Rerum Materia

Practical or otherwise, what can philosophy contribute to a discourse on technology? The researcher may be engaged in pursuits bordering on the philosophical, pondering the nature of the infinite from the perspective of mathematical set theory, but how do the philosophers regard the scientist generally, and the computer scientist specifically?

In the Western tradition the earliest recorded reflections on technology began in ancient Greece, with Aristotle suggesting that *technè*, roughly translated as the practical application of craft, was capable of moving beyond imitation of the natural world and into the realization of the novel, advancing nature into directions it may logically, structurally imply through the intervention of human thought and will.[12] But it required another two millennia before the German philosopher Martin Heidegger revisited the concept. Witnessing the rapid advance and previously unimaginable power of technology, Heidegger began evaluating it through the medium of individuals and societies. His goal was to create a framework suitable for broadly examining technology in order to reveal its true nature, its *essence*. "Only the true brings us into a free relationship with that which

[10] Graham, "Revenge of the Nerds."
[11] Marx, "Theses on Feuerbach."
[12] Parry, "Episteme and Techne."

concerns us from out of its essence,"[13] he writes, and from this he concludes that only within this free relationship will the application of technology permit an experience of the real.

Assuming it can be achieved, why is such an experience important? Heidegger argues that the preexisting consensus assumed technology to be value-neutral, neither good nor bad beyond the cause for which it is applied. In his view this assumption of instrumentalism is unavoidably reductive, because the tool, the technology itself, is just as significant as the changes it introduces, the new state of being resulting from its use. Not simply a means to an end, both the tool and its application work together to either reveal or conceal its truth, and this new reality distinguishes contemporary technology from the technè of Aristotle. The ancient practices revealed, taking a feature of the natural world and refining it. Contemporary, science-based technologies are more likely to conceal, according to Heidegger. Modern technology is neither progressively beneficial nor neutral: Its existence has social, economic, and political consequences, both known and unknown.

Heidegger and his adherents believed that our failure to consider this technological reality resulted in an ultimately negative impact upon societies and cultures. Albert Borgmann extended Heidegger's phenomenological approach by contextualizing the primary expression of technology as the *device*, a means for fulfilling an individual's potentially unexamined goals and desires without consideration of side-effects and unintended consequences, necessitating a reconsideration of social practice in order to restore a healthier relationship between technology, individuals, and society.[14] Some, including American postphenomenologist Don Ihde, argued that philosophers should address this absence by actively engaging computer scientists and engineers, in the hopes of

[13] Heidegger, *The Question Concerning Technology, and Other Essays*, 4.
[14] Borgmann, *Technology and the Character of Contemporary Life: A Philosophical Inquiry*, 42.

positively influencing and informing their practices.[15] In response to the suggestion that philosophers have little to contribute to the development of technology, ethicist Carl Mitcham extends Heidegger's observation that the various and related technical fields produce evident first-order questions—namely, the problem to be solved—along with distinctly philosophical second-order questions on the true nature and identity of a technology and its corresponding meaning. These questions have the capacity to extend beyond the social, economic, and political. Instead of adopting a traditional "humanities philosophy of technology," which considers the material a "black box," subject to social and cultural interpretation, Mitcham recommends a deeper understanding in order to generate a more active engagement, and a meaningful ethical philosophy surrounding both the development and application of advances in the sciences.[16]

As technologies and related disciplines evolve, so too do the concerns of philosophy, with new lines of inquiry emerging. The generation and transmission of previously unfathomable volumes data, structured or unstructured, spurred a new philosophy of information as a means for reconsidering the concept of identity and the underlying nature of information itself.[17] Sociological considerations, such as the assumed, shared norms among practitioners in both scientific and applied professional communities, are increasingly subject to philosophical evaluation.[18] Regardless of the approach taken, it becomes clear that the value of philosophy comes from its examination of technology through the lens of theory, contemplating the potential consequences of technological development beyond their professed beneficial or Utopian intentions. Instead of assuming that we control technology, philosophy challenges us to contemplate how technology controls, influences, and changes us. When critically considering the positivistic concepts of

[15] Ihde, "Technics and Praxis."
[16] Mitcham and Robert Mackey, *Philosophy and Technology*, 7.
[17] Floridi, "Information," 42.
[18] de Vries, Sven Ove Hansson, and Anthony W.M. Meijers, *Norms in Technology*, 9:2.

science and the faith in technical solutions characterizing the scientist, computer or otherwise, along with a corresponding tendency to generate and reinforce a belief in quantifiable meritocracy that these concepts insinuate, such a perspective becomes increasingly necessary.

Practice Makes Perfect

As the influence of digital technologies expanded, the influence of the academy waned in favor of the waxing profiles of practitioners. Students still learned to recreate the discrete computational structures described by Turing, but consumers saw only commodities, concrete plastic and metal boxes branded by IBM and rated according to mysterious components well-suited for commodity fetishism, such as processor speed, available memory, or graphics capabilities.[19] The form factor of a computer soon evolved from "enterprise" to "personal," from filling a climate-controlled room to an object so small it could fit into the back pocket of a pair of trousers (in the process, adding the term "butt dial" to the popular lexicon). The cultural and economic impact of the so-called information age was less obviously the product of the researcher and became widely attributed instead to the practitioner. It was this new development, a development of "developers," which brought an established yet subliminal phenomenon into sharp focus.

For our purposes, who are the practitioners, and why should we consider them? Ask the average so-called "man on the street," and responses might include Bill Gates, or Steve Jobs. But were they, in fact, practitioners? Gates could certainly write software code—if nothing else,

[19] "There it is a definite social relation between men,that assumes, in their eyes, the fantastic form of a relation between things. In order, therefore, to find an analogy, we must have recourse to the mist-enveloped regions of the religious world. In that world the productions of the human brain appear as independent beings endowed with life, and entering into relation both with one another and the human race. So it is in the world of commodities with the products of men's hands. This I call the Fetishism which attaches itself to the products of labour, so soon as they are produced as commodities, and which is therefore inseparable from the production of commodities." From Marx and Engels, *Capital, A Critique of Political Economy*.

donkey.bas provides evidence of that[20]—and former coworkers and employees agree that he possessed a sharp, critical intelligence capable of interpreting complex software-based solutions in real time, quickly identifying structural flaws.[21] His genius and legacy, however, were not revealed in the proprietary designs of Microsoft's Windows or Excel, but in the business decisions resulting in their becoming a de jure corporate monopoly. Likewise, Jobs publicly associated with an early generation of practitioners, participating in their culture, but elected to cherry-pick from among their public contributions (in some cases lifting designs and concepts directly, such as those created at Xerox's Palo Alto Research Center) in order to market products to the home consumer.[22] These are certainly examples of genius, but the genius of business is not the focus of this investigation. A critical examination of commercial success could potentially cast too wide and varied a net to reel in statistically significant patterns within the scope of this inquiry. In the end it wasn't Jobs, but his friend and partner Steve Wozniak who inherited the traditions of practical computer science, and these traditions incorporate ideological leanings that go beyond the marketplace—even when they never stray far from it.

Why not include more contemporary examples, such as Larry Page and Sergei Brin of Google, or Facebook's Mark Zuckerberg? Lesser-known luminaries such as Sir Tim Berners Lee, the above-mentioned Paul Graham, or his fellow programmer-turned-investor Marc Andreessen might all be worthy of consideration. Even non-programmers, such as Jimmy "Jimbo" Wales of the Wikimedia Foundation, entrepreneur Elon Musk, or venture capitalist Peter Thiel might qualify based on certain criteria. The list of influential non-academics, or of former academics with origins in computer science, is long, but to give this phenomenon of computing, neurodevelopment, and

[20] Atwood, "Bill Gates and DONKEY.BAS."
[21] Spolsky, "My First Bill G Review."
[22] Kort, "The Story of Steve Jobs, Xerox, and Who Really Invented the Personal Computer."

ideology a fair hearing we can arguably restrict our focus to two examples: two programmers with an extensive body of public writings and statements that lend themselves to close analyses, and who exist within an historical continuum illuminating our current conditions. We can consider the examples of Richard Matthew Stallman and Linus Benedict Torvalds.

Through the biographies, substantial record of activities, public statements, and writings surrounding these programmers, I intend to highlight common traits and shared characteristics, along with the manifestations of their labors and their transformative qualities. Although largely unfamiliar to the general public, both are enormously influential, with Stallman an acclaimed genius by virtue of receiving the John T. and Catherine D. MacArthur Foundation's Fellows award[23], and Torvalds' self-declared pastime resulting in the creation of a non-profit foundation supported by AT&T, IBM, Intel, Microsoft, and numerous other major technology companies for the sole purpose of supporting his work.[24] Both claim not to be politically motivated, although they often express political views, sometimes with the caveat that they bear little direct relationship to the product of their labors. Neither are overt, potentially obvious examples of the observed pattern either, when compared to examples like self-professed libertarians Eric S. Raymond and Peter Thiel (who, coincidentally, could both also be plausibly regarded as neurodivergent). Curiously, Stallman and Torvalds form near-opposite poles of specific contemporary practices in contemporary technology, namely the distinct but frequently conflated Free Software and Open Source Software movements. Their examples will aid us in historicizing these cultural and ideological traditions, contributing toward an interrogation of the connections between identity, ideology, and impact.

[23] MacArthur Fellows Program, "Richard M. Stallman."
[24] Lohr, "Group Formed to Support Linux as a Rival to Windows."

A Crossroads

The new, increased public profile and social prominence of the programmer reveals additional commonalities beyond the neurological. Certain ideological proclivities appear to be manifesting concurrently, uncovering a framework of shared values, beliefs, and biases that in turn reveal eccentric political elements. Through the examples of Stallman, Torvalds, and their contemporaries, my goal is to examine these patterns through the lens of systematic inquiry: does the contemporary practice of computer science lend itself to adherents of libertarian-styled meritocracy, and does a relationship exist between this phenomenon and a high incidence of neurodivergence among practitioners?

I begin with Stallman, the so-called "crusader" for freedom, and a consideration of the "hacker" culture, developed around post-war technical advances, which made his work possible. Described by legal scholar Lawrence Lessig as a man who "built his career on a stage of public life, as a programmer and an architect founding a movement for freedom in a world increasingly defined by 'code,'" Stallman enshrined a shared community practice into a legal doctrine for individual rights, in the process revolutionizing the technology marketplace.[25] These accomplishments came at a great personal cost, but his singular nature suggests an inexorable opportunity to fulfill the promise of greatness beyond the walls of the academy that he could not refuse.

I move on to Torvalds, a self-proclaimed "benevolent dictator." The child of political radicals and communists, Torvalds explicitly rejected the values inherited from Stallman while making the single greatest contribution to his project: the Linux kernel, a core component of

[25] Lawrence Lessig, "Forward to the First Edition", in Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman*, vii.

Stallman's free computer operating system, without which it would most likely not have become the primary basis for over 80% of all smartphones and the system underlying 96% of the Internet.[26] Valuing pragmatism and personal merit over social benefit, Linus' approach offered enthusiasts a far more comfortable, less demanding point of entry into participatory technology that resonated with their other predilections, in the process providing as much of a contribution to the project's success as any technical advance, if not more. We are, however, only now beginning to realize the social and political cost of the trade-off between ethics and accessibility: With commercial success and mainstream acceptance, the path from an unexamined application of meritocracy to a belief in the superiority of unfettered individualism appears remarkably short.

This thesis is not an attempt at diagnosis, but at exploring an apparent phenomenon, an intersection revealed in the Venn diagram composed of computer science, behavioral neurology, and political philosophy. The expression of a series of beliefs and traits accompanying technical brilliance offers a means for interrogating how the ubiquitous information systems and devices we rely upon came into being. In the process we stand to gain a real—in the phenomenological sense—understanding of our individual relationship with technology, its influence on our culture, and its impact on global society. For better or worse, our descriptive terminology continues to evolve: Techno-utopian? Possibly. Disruptor? Innovator, preferably. Genius, according to some. Superman?

[26] StatCounter, "Mobile Operating System Market Share Worldwide."

Chapter 2

The Crusader

> If I am not for myself, who will be for me? If I am only for myself, what am I? If not now, when?
>
> —Hillel the Elder, *Pirkei Avot*, 1:14[27]

In 1980 the Massachusetts Institute of Technology's Artificial Intelligence Laboratory received a new, state-of-the-art laser printer from the Xerox Corporation. An expensive rarity at the time, when operating properly it produced beautiful, typographically formatted documents. When attached to the lab's computer network it became available to all members of their research community.

There was one observable flaw in this arrangement, however: the device's high operating speed resulted in frequent paper jams. And, unless someone was within sight of the machine, it was virtually impossible to determine that a jam occurred. The designers and engineers at Xerox, a company known primarily for photocopiers, were in the habit of assuming that, wherever their equipment could be found, a human operator would always be nearby. To a young AI Lab programmer named Richard M. Stallman this was a familiar problem, one that was not so much solved as cleverly "hacked." He could modify the printer's software instructions to check periodically for a jam, and send a warning message notifying all the users on the network. Someone nearby could then fix the issue, eliminating hours of lost productivity.

But Stallman soon discovered another, hidden flaw: Xerox didn't supply the source code to the printer's operating system.

No matter. Stallman (or RMS, as he preferred to be called) was part of a larger hacker

[27] The subject of this chapter, Richard Stallman, describes drawing inspiration from this quote when launching his project to develop software based on user freedom, in Stallman, "The GNU Project."

culture. To a hacker, sharing software was like sharing an idea. It's a practice often described by hackers using Thomas Jefferson's analogy of sharing a candle's flame: the resulting new flame does not diminish the original candle's brightness, and the world becomes even brighter for it.[28] Stallman assumed that another hacker at another university laboratory had access to the code and would share it, as he had personally done when approached in the past with similar requests. He soon learned that an engineer at Carnegie Mellon University could satisfy his requirement. This time, however, when he made the request it was flatly denied. The reason? His fellow programmer had signed an NDA, or nondisclosure agreement, with Xerox. An NDA may be something we take for granted today, but it was a novel practice at the time, and essentially unknown in the computer programming community. "It was my first encounter with a nondisclosure agreement, and it immediately taught me that nondisclosure agreements have victims," said Stallman. "In this case, I was the victim."[29]

Strong words from someone attempting to modify a piece of office equipment. Who was this RMS, and why would he identify as a victim when deprived of access to software code?

Progress and Regress

A gifted student from Manhattan, Stallman first arrived at MIT in 1970 while completing a degree in physics at Harvard. Once there he discovered two things: a love for the nascent science of computer algorithms, and a home. The MIT Artificial Intelligence Laboratory offered an environment built on intellectual curiosity, nurtured by an ethic he personally shared. An offspring of DARPA, the United States Department of Defense Advanced Research Projects Agency, it formally began in 1970 under the leadership of cognitive scientist Marvin Minsky, widely

[28] Jefferson, "No Patents on Ideas (Letter to Isaac McPherson)."
[29] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 9.

considered the father of Artificial Intelligence research, and quickly developed a culture following his values of rigorous critical intelligence, inventive imagination, and unfettered curiosity. But in the decade following the Vietnam War, economic support for the lab was changing: government defense spending decreased, compelling universities and laboratories to compensate for their losses with funding from private industry. An institution as prestigious as MIT had little difficulty attracting investors, but in doing so they sacrificed their culture of sharing. Factions quickly developed in favor of capitalizing on internal research projects, and Stallman soon learned that improving collective resources through unrestricted, open collaboration was no longer an option.

Only four years before the lab's new printer arrived, Congress authorized the Copyright Act of 1976.[30] For the first time in history, protections provided to authors were now extended to software developers, and the literal text-based expression of a program became subject to copyright. As with written works, even programs without explicit licenses were now subject to the law, so programmers began attaching notices to their code in response, stating the terms under which it could be copied. Assuming, of course, their code was even available for distribution in the first place.

There was at least one problem with this new, monetized software regime: few programs existed that did not already borrow—often directly—from earlier programs. With this new legislation, Congress empowered companies and organizations to assert individual authorship over works that were very likely to have been developed communally.

The incident with the printer became Stallman's proverbial "origin story." Given the choice of earning a lot of money while losing the right of ownership, or preserving a culture of sharing and liberty, he chose the latter. As though figurative scales fell from his eyes, he suddenly saw a larger

[30] McClellan, The Copyright Act of 1976.

trend that opposed his personal beliefs, and that he was determined in turn to oppose. "From that day forward I decided this was something I could never participate in," he said in a 2002 interview. "I decided never to make other people victims just like I had been a victim."[31]

The Marketplace of Ideas

Software quickly became regarded as a high-value asset, and Stallman grew determined to both provide alternatives to "productized" programs, and to preserve a culture with roots going back to the Second World War.[6] He began by developing a new version of Emacs, a text editing program, and creating a license intended to establish a software commons. With its infinite reproducibility, his software couldn't suffer from William Forster Lloyd's "tragedy of the commons," so long as any artificial impediments to distribution were removed.[32] This was software as social contract, an early example of licensing intended to build communities of collaborative contributors.

Unfortunately, when confronted with this new regime, not all of Stallman's colleagues retained their previously shared ethic, and the AI Lab's administration soon moved to replace every component of their computing infrastructure with a new, proprietary system. In response, Stallman worked day and night to recreate this new environment's features entirely from scratch, not by copying its code but through implementing from observation, carefully parsing and reconstructing the new system's every component feature and output. In the process Stallman became something of a legend in the hacker community, a "John Henry of computer code" as one rival developer described him.[33] Working alone, he produced an entire alternative system, successfully competing

---

[31] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 11.

[32] Lloyd, *Two Lectures on the Checks to Population*, 31; *The Wealth of Networks: How Social Production Transforms Markets and Freedom*.

[33] Levy, *Hackers: Heroes of the Computer Revolution*, 351.

with a project created by teams of programmers over the course of years. Although regarded by some as one of the major human accomplishments of the early information age, it came at the cost of social alienation and great personal loss. Stallman's actions set him squarely against the new culture of programmers owning and concealing their code while profiting from the compiled results. He established himself as the living embodiment of a political philosophy applied to an entire industry, a line in the sand which offered no ambiguity or compromise.

The Ultimate Hack

> As the 1970s gave way to the 1980s, selling software became more than a way to recoup costs; it became a political statement. At a time when the Reagan Administration was rushing to dismantle many of the federal regulations and spending programs that had been built up during the half century following the Great Depression, more than a few software programmers saw the hacker ethic as anticompetitive and, by extension, un-American. At best, it was a throwback to the anticorporate attitudes of the late 1960s and early 1970s. Like a Wall Street banker discovering an old tie-dyed shirt hiding between French-cuffed shirts and double-breasted suits, many computer programmers treated the hacker ethic as an embarrassing reminder of an idealistic age.[34]

Stallman realized that being true to his beliefs required devoting his life to the creation of software that would afford others the opportunity to follow the same path. He pledged to build a completely free operating system from the ground up, and by free he meant free as in freedom, not free as in "free beer." An adherent of Wittgenstein's proposition that words, with their ability to confuse, must be used with precision, he adopted this expression as compensation for an ambiguity traceable to the Germanic origins of the English language, occasionally preferring the Latin-based variants *libre* and *gratis*.[35] Having arrived at the denomination "Free Software," his next move

---

[34] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 100.

[35] Early in his career, philosopher and logician Ludwig Wittgenstein proposed, in his *Tractatus Logico-Philosophicus*, that statements are meaningful only inasmuch as they correspond to the world around us. Although he subsequently reconsidered this theory, following the Second World War it proved very influential among a generation of researchers in the field of cognitive science, and by extension computer and information science. The Center for Cognitive Science included Wittgenstein along with MIT's Noam Chomsky and Marvin Minsky in their list of influential authors, archived at "The Center for Cognitive Science's Millenium Project."

became clear: Stallman resigned from the MIT staff in January of 1984 to start the GNU Project, dedicated to creating a new version of the popular Unix operating system and its associated tools, and applying his newly created GNU General Public License, or GPL, to the results.[36]

The GPL is often regarded as Stallman's finest hack, hacking a legal system instead of an operating system, and one that preserves communal software ownership while working within the constraints of proprietary copyright law. He described it as "copyleft," and expressed it in a form intelligible to both lawyers and programmers alike. Stallman's genius was in recognizing the similarities between legal and software code; although not identical, their systemic correspondences were more than sufficient for a skilled hacker to effectively hack the law.

The GPL permits free use and distribution of covered programs, with the added requirement that if any changes or improvements are introduced, these modifications become equally subject to the license. Put simply: if you release your new version to the public, for sale or otherwise, you are required to include the source code containing your changes. It's not unlike a manifestation of Isaiah Berlin's positive freedom, in that participation is an act of agency, free of coercion. The restriction introduced by the license limits negative freedom (freedom from interference) by offering the individual a positive choice.[37] You are always free to adopt a different approach to licensing, of course, but doing so removes your work from the commons.

In a broader sense, software is information in one of its most basic forms, simultaneously a series of sequential instructions and a source of knowledge to those who would become

[36] "GNU is an operating system that is free software—that is, it respects users' freedom." From Stallman, "The GNU Project"; The GNU General Public License, version 3, can be read in its entirety online at "GNU General Public License."

[37] Berlin, "Two Concepts of Liberty." In this 1958 University of Oxford inaugural lecture, political theorist Isaiah Berlin outlined the distinction between "negative" freedom, or freedom from external constraint, and "positive" freedom, freedom of action and self-direction. Berlin argued that both are necessary for a free society, however jurisdiction based exclusively on negative freedom is commonly regarded as a tenet of contemporary libertarianism.

practitioners. As Michel Foucault suggested in *Discipline and Punish*, the goals of power and knowledge cannot be separated.[38] By retaining and sharing knowledge we retain and share control, and with that control we gain continuing knowledge. The GPL perpetuates a dialog that was often referred to as "viral" and "cancerous" by its economic opponents, and characterized as a false or incomplete form of freedom by ideological opponents. Yet in spite of these claims and their arguable accuracy (or perhaps because of them), the GPL accomplished Stallman's goals, demonstrating the elegance of a programming hack made socially manifest.

Six years later, businesses began lobbying the courts to consider applying patent law to software, which would explicitly prevent copyrighting code as a form of speech and give exclusive rights to abstract, conceptual expressions. Discussing the hypothetical situation of patented software, Stallman justified his ethical approach. "It's not because we don't have the talent to make better software," he said, "It's because we don't have the right. Somebody has prohibited us from serving the public."[39] Often regarded as extreme, even his greatest critics began to view him as politically necessary, alternately a pole star, lighting a clear path for those approaching technology as an instrument of freedom, or a prophet, warning people of the dangerous consequences of thoughtless technology choices.

Speaking to an audience at the Maui High Performance Computing Center's 2000 conference, Stallman reflected on his decision to give up everything and dedicate his life to a cause. "I had just the right skills. Nobody was there but me, so I felt like, 'I'm elected. I have to

[38] Foucault, "The Birth of the Prison." Investigating the theoretical foundations of the contemporary penal system, Foucault argues that prisons form part of a larger, multi-institutional network of social control, with restrictions on access to knowledge and education a key component of the resulting political inequities. "[T]here is no power relation without the correlative constitution of a field of knowledge, nor any knowledge that does not presuppose and constitute at the same time, power relations."

[39] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 115.

work on this. If not me, who?'"[40]

Signals & Power

> In 1971 when I joined the staff of the MIT Artificial Intelligence lab, all of us who helped develop the operating system software, we called ourselves hackers. We were not breaking any laws, at least not in doing the hacking we were paid to do. We were developing software and we were having fun. Hacking refers to the spirit of fun in which we were developing software. The hacker ethic refers to the feelings of right and wrong, to the ethical ideas this community of people had—that knowledge should be shared with other people who can benefit from it, and that important resources should be utilized rather than wasted.[41]

Placing Stallman's achievements into historical context requires some understanding of the cultural forces that shaped him. In the narrative of events anticipating his crusade, frequent reference is made to "hacks" and "hackers." Contemporary users of software-based technologies may be familiar with the terms, although, in the absence of a standard definition and the presence of many pop culture applications—not least among them the 1995 film of the same name—associations may vary. The question necessarily becomes, "What does the term hacker mean according to Stallman?" Who were the hackers, and how did he come to be associated with them?

One of the original authoritative sources on the subject is Steven Levy, author of *Hackers: Heroes of the Computer Revolution*. Although a common word of Germanic origin (still occasionally applied to agricultural laborers), on the Massachusetts Institute of Technology campus of the late twentieth century hackers were computer programmers attached to the Artificial Intelligence laboratory's Building NE43, commonly referred to as "Tech Square." Taking their name from the practice of describing a student prank as a hack, these MIT hackers expressed their cleverness and humor through the modification of computer systems, frequently without permission or authorization, according to practices that began in one of the campus recreational

[40] Ibid., 22.
[41] Stallman, MEME 2.04.

societies. Known as the "Tech Model Railroad Club," its members built an incredibly complex and large miniature town in which to operate custom-built scale model replicas of trains. This town wasn't simply a static landscape, however: embedded below the surface was an expansive and intricate set of cables, engineering switches, and programmable controls, implementing the timetables and operating the tracks while animating the landscape. The club members responsible for designing and constructing this environment were the elite members of the Signals & Power Subcommittee. Scavenging for parts, laboring throughout the nights and weekends, and setting up unauthorized telephone lines and power relays, for these young men (and they were exclusively men, if unintentionally so), in the words of Levy, "[t]echnology was their playground."[42]

In 1959, when a colleague from the engineering department introduced several Railroad Club hackers to the TX-0, a newly-acquired "minicomputer" filling a small room in Building 26, there was no returning to model trains. Most of them had been dreaming since childhood of operating a computer, and the opportunity for free access to a real one was too good to pass up. Soon their culture of clever improvisation and unimpeded experimentation transferred from the rails to the keyboard, along with a shared collection of technical idioms and a corresponding ethical creed. This "hacker ethic" consisted of five elements:

- Access to computers—and anything that might teach you something about the way the world works—should be unlimited and total.
- All information should be free.
- Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.

[42] Levy, *Hackers: Heroes of the Computer Revolution*, 9.

- You can create art and beauty on a computer.
- Computers can change your life for the better.[43]

As a belief system, the hacker ethic combines elements of classic Anglo-American liberalism, with its emphasis on personal freedoms and natural rights, along with a nod toward egalitarianism that, once again, neglects gender and reflects some unintentional irony when describing a community enrolled in an elite institution—even those who subsequently accessed and hacked the MIT computers remotely beginning in 1970, using the nascent national wide area network ARPAnet, were sufficiently privileged as to have access to a computer at a time when private individual ownership was largely unheard of. But it is the last two tenets that reveal something more: A clear passion for the practice of programming, with a corresponding belief in the creativity and artistry involved and a respect for innovation and talent. These hackers weren't designing the fundamental architecture of an entire industry as a research project or corporate assignment, but because it gave them pleasure, and created a community of shared values and rewards in the process.

In the light of the remarkable and wide impact made by MIT hackers over the course of thirty years, contemporary researchers such as philosopher Pekka Himanen regard their ethic as a fundamental shift in a society's relationship to labor resulting from the information age, not unlike the ascendancy of the protestant ethic first defined by Max Weber. "Whereas the historical precursor for controlling the free flow of information is the monastery, […] the historical precedent of the hacker ethic is the academic or scientific ethic."[44] An increasingly networked society replaces religious tradition as the primary channel for transmitting this new set of values. For a

[43] Ibid., 27–38.
[44] Himanen, *The Hacker Ethic, and the Spirit of the Information Age*, 46.

young Richard Stallman, however, it was the fulfillment of another childhood dream: A community of like-minded individuals, and one of the few environments in which he truly felt at home.

Hands On

> But also I was inspired by the attitude at the MIT AI Lab, where the hackers said: "We're not going to let the administrators tell us how to do things; we're going to work on what they need, but we will decide how; and we won't let them implement computer security to restrict us with." […]Their attitude was, yes, the administrators could fire us, but we were not going to suck up to them. They weren't going to stand being treated like ordinary employees. I wouldn't have had the strength to do this on my own, but as part of a team, I learned it. We were the best, and most of us weren't getting paid an awful lot—any of us could have got a much better-paying job someplace else if we'd wanted. We were there because we were free to improve the system and do useful things, the way we wanted to, and not be treated like people who had to obey all the time.[45]

Stallman first became directly acquainted with computers during his final year of high school, when a Manhattan-based IBM research center hired him for the summer, offering the opportunity to work on small projects. He completed his assigned work over the course of a few weeks, leaving him free to familiarize himself with their systems for the remainder of his stay. A brilliant student in the applied sciences, Stallman successfully matriculated at Harvard University as a physics major in 1970, where he enjoyed visiting campus computer labs for the opportunity to familiarize himself with their manuals and documentation. Exhausting the materials available around Harvard, he soon traveled across Cambridge to the MIT Artificial Intelligence lab, curious about their unique, internally designed ITS, or "Incompatible Timesharing System."[46] Recognizing a kindred spirit, the lab administrator hired him, "more or less straight away."[47]

His obsessive, intensely focused curiosity dovetailed perfectly with the hacker ethic's first

[45] Lucas, "Richard Stallman: Talking to the Mailman," 70.

[46] The name is an example of hacker humor, tending toward puns, parody, and wordplay. ITS was created partly in response to a previous, academically crafted platform called the CTSS, the "Compatible Time-sharing System," which, among other hacker design flaws, implemented security features they found undesirable.

[47] Lucas, "Richard Stallman: Talking to the Mailman," 69.

principle of unrestricted access to computers, the so called "hands-on imperative," but even more suitable was a lifelong disdain for what he regarded as "unqualified" authority. Even in early childhood, if Stallman didn't recognize the validity or reasoning behind an instruction he gave no thought to rejecting it. Evaluating his mother's calls to the dinner table as opposed to continued study, he chose the latter: "Essentially, what I had read about, ideas such as democracy and individual freedom, I applied to myself. I didn't see any reason to exclude children from these principles."[48] In an environment created and populated by hackers, who experienced no moral quandary picking the lock of a colleague's door to access their unused computer terminal in the middle of the night, such an attitude went largely unremarked.

What distinguishing characteristics evident in Stallman and his hacker colleagues might contribute toward explaining their shared ethic? Beyond finding themselves at an elite engineering institution, what separated them from those peers who favored careers in mathematics, or conducting research in the emerging discipline of computer science? Based on anecdotes and interviews documented by Levy, the "true hackers," Signals and Power Subcommittee members who migrated to the AI lab after graduating from (or dropping out of) MIT, appeared to share numerous characteristics, among them an affinity for systems of logic and order. Founding hacker Richard Greenblatt described his early fondness for electronics based on its identity as, "[a] world where there were no ambiguities. Logic prevailed. You could build things according to your own plan. To a nine-year-old whose intelligence might have made him uncomfortable with his chronological peers, […] electronics was the perfect escape."[49]

Captured in Greenblatt's reflection is the hacker desire for control, more specifically for

[48] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 27.
[49] Levy, *Hackers: Heroes of the Computer Revolution*, 62.

mastery over a domain offering the means for creative expression. This desire manifests prominently in the fourth principle of the ethic, in which computers can be a source of art and beauty in the right hands. But whose hands are the right ones? Underlying the premise of the supposedly egalitarian third principle, in which hackers are judged by the quality of their hacks as opposed to external or circumstantial characteristics, is an unspoken value system, suggesting that participation in this community requires talent and skill as evaluated by its members, a determination of personal merit.

It should come as little surprise that constituents of a research lab shared a belief in scientific positivism, often to the point of reducing all solutions to a testable, logical paradigm they termed "The Right Thing." According to Levy, hackers believed that any technical problem had a provable, ideal solution, and the pursuit of this ideal often resulted in heated, passionate, arguments. Founding hacker Bill Gosper explained that The Right Thing, "very specifically meant the unique, correct, elegant solution… the thing that satisfied all the constraints at the same time, which everyone seemed to believe existed for most problems."[50] Analyzed through the lens of political ideology, The Right Thing resembled the more anarchic tenets of classical liberalism, such as a belief in the natural right of the individual to personal liberty, and to freedom unrestrained by coercion. Yet the most powerful hacker ethic, in terms of its influence on Stallman, was the expression of community.

Arriving a decade after the initial hacker community first formed, Stallman worked directly with Greenblatt and Gosper, taking them on as mentors and earning the mantle "last of the 'True Hackers.'" For Stallman, the lab was the embodiment of a philosophy of anarchic yet ultimately constructive cooperation. But the lab revealed a culture created by those who, like Stallman, came

[50] Ibid., 69.

from a place of great personal alienation and isolation. "[M]any hackers were loners to begin with, socially uncomfortable. It was the predictability and controllability of a computer system—as opposed to the hopelessly random problems of a human relationship—which made hacking particularly attractive."[51] When they came together, in the confines of the AI lab and, later, across the global Internet, hackers converted this shared passion and corresponding ethic into something that addressed a very real void in their lives, one that wasn't merely the pursuit of financial gain. "The idea was to make a computer more usable, to make it more exciting to users, to make computers so interesting that people would be tempted to play with them, explore then, and eventually hack on them. When you wrote a fine program you were building a community, not churning out a product."[52]

Considering the hacker's uninhibited and challenging qualities, it's not difficult to imagine how hacker culture was eventually demonized by Bill Gates and his fellow entrepreneurs as they began capitalizing on software. The belief that access to computers should be unrestricted, and that "information wants to be free," migrated to the West Coast of the United States as the MIT hackers increasingly became employed by Bay Area engineering companies or "poached" by rival research institutions, like the new Stanford Artificial Intelligence Laboratory, SAIL. As computing became "personal," the hacker ethic directly threatened a developing new business model Gates was pioneering, leading him to misrepresent the established discourse by means of rhetorical accusations of theft and "piracy."[53] As the influence of the MIT hackers waned and technology industries evolved, the term became increasingly pejorative, frequently applied to agents responsible for network security breaches, regardless of ideology or motivation. Stallman's later

[51] Ibid., 75.
[52] Ibid., 46.
[53] In is famous "An Open Letter to Hobbyists." Gates bluntly states, "As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?"

activism, his precision with language, and his self-identification as the last "true" hacker, heir to the AI lab legacy, were highly influential at rehabilitating the term, but the hacker cultural legacy has survived in large part due to one of their many gifts: their anarchic, decentralized, non-hierarchical and elegantly engineered legacy survived in the structural foundations of the Internet and World Wide Web, reflecting the social history of computer science and demonstrating again what the phenomenologists understood.[54] Technology is not neutral, and the hacker desire for a creative medium introduced their art and ethic to the world at large in subtle but profound ways.[55]

The ethic of community, highly prized by Stallman—and initially expressed through a foundational community of hackers, researchers, and lonely, isolated would-be hackers across America, formed into a boundless, "virtual" lab—is a fundamental quality of the Internet. A decade after his arrival at MIT, however, the community of hackers at the AI lab found itself destroyed by outside forces it could not oppose. In 1973, Montana Senator Michael Mansfield introduced an amendment to the Military Authorization Act expressly limiting ARPA appropriations for defense research exclusively to projects with direct military application. The AI lab found itself without sufficient funding for the first time, and veteran hackers were urged to organize themselves into profitable business ventures. One such venture, based around building "Lisp machines," general-purpose computers designed for the lab's LISP ("List Processor") programming language, resulted in two competing factions, forcing Stallman to choose sides and support those colleagues who

[54] Rosenzweig, in reviewing competing narratives of the history of the Internet, concludes that they were all essentially correct: "Such a profound and complex development cannot be divorced from the idiosyncratic and personal visions of those scientists and bureaucrats whose sweat and dedication got the project up and running." In "Wizards, Bureaucrats, Warriors, and Hackers: Writing the History of the Internet," 1552.

[55] "This 'ARPAnet' was very much influenced by the Hacker Ethic, in that among its values was the belief that systems should be decentralized, encourage exploration, and urge a free flow of information. From a computer on any 'node' on the ARPAnet, you could work as if you were sitting at a terminal of a distant computer system. Hackers from all over the country could work on the ITS at Tech Square, and the hacker values implicit in that were spreading. […] The contact helped to normalize hackerism, so you could find hackers in Utah speaking in the peculiar jargon developed in the Tool Room next to the Tech Model Railroad Club. Levy, *Hackers: Heroes of the Computer Revolution*, 138.

preserved the hacker ethic of information freedom over the faction operating based on the perceived competitive advantage of withholding their programming code. This was the event that resulted in his "heroic" year-long one-man recreation of the hidden software's features, but in the end the damage done to the community was irreparable. Hackers migrated to private industry and other institutions, and in the absence of government funding they were not replaced.

So it happened that in 1983 Stallman found himself without a community. "I'm the last survivor of a dead culture," he wrote, "and I don't feel I belong in the world anymore."[56] Faced with this dissolution, he consider dropping out of programming entirely. "I suppose I could have stopped working on computers altogether. I had no special skills, but I'm sure I could have become a waiter."[57] Computer programming was a source of great personal pleasure for Stallman, so he dedicated himself to continuing the practice in the service of recreating the community he lost, and opposing those who would destroy its values, and, by extension, its right to exist.

> I think that hackers—dedicated, innovative, irreverent computer programmers—are the most interesting and effective body of intellectuals since the framers of the U.S. Constitution…. No other group that I know of has set out to liberate a technology and succeeded. They not only did so against the active disinterest of corporate America, their success forced corporate America to adopt their style in the end. In reorganizing the Information Age around the individual, via personal computers, the hackers may well have saved the American economy…. The quietest of all the 60s sub-subcultures has emerged as the most innovative and powerful.[58]

Considering the Elephant

In the narrative of Stallman, his genius, and the culture that emerged from his fellow hackers, we find ourselves returning to questions of difference. We read of a group, largely male and socially challenged by their environment, constructing an alternative environment more suited

[56] Ibid., 450.
[57] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 100–101.
[58] Stewart Brand, in Levy, *Hackers: Heroes of the Computer Revolution*, 455.

to their nature. That nature is expressed through an affinity for systems of order and logic in which they can exercise their creativity, systems providing evident truths and demonstrably correct answers. And, having established an environment conforming to their nature, it becomes a source of obsessive attention, to the point where some members participate to the exclusion of all other activities, including the academic pursuits that initially brought them together. This single-mindedness in pursuit of a perfect hack, or the solution to a challenging puzzle, was often expressed through days of uninterrupted programming without sleep, or indifferently breaking in to locked rooms housing coveted equipment. A pattern of shared behaviors and attitudes arises. To observers and participants in the various related fields of computer science, it is a pattern that occurs among practitioners with noticeable frequency.

In response to a fellow software developer's blog posting on the topic, Jeff Atwood pondered an issue often regarded as an "elephant in the room" of the technology industry, a prominent issue rarely addressed directly. Put bluntly, he wrote, "Many of the best software developers I've known share some of the traits associated with Asperger's Syndrome."[59] Writing over thirty years after Stallman left MIT to build a Free Software movement, Atwood described colleagues fixated on order and literal interpretation, who preferred working with systems over people and possessed near-limitless abilities to focus on details over prolonged periods. Does a connection exist between a professional affinity and what psychologists describe as a neurobiological condition? Does the presence of the latter contribute to increased abilities in the former?

To consider this possibility, it becomes necessary to understand what we mean by Asperger's Syndrome, a condition defined by psychiatrists as a "pervasive developmental disorder"

[59] Atwood, "Software Developers and Asperger's Syndrome."

that became subject to debate and disagreement—to the point where many professionals in the field are not convinced it legitimately exists. The term, alternately "Asperger Syndrome," was first applied in 1981 by English psychiatrist Lorna Wing, referencing the research conducted by Austrian pediatrician Hans Asperger in 1944 on a range of symptoms observed in children considered mildly autistic (and which echoed work previously published in 1943, unbeknownst to Asperger, by Austrian émigré Leo Kanner at Johns Hopkins Hospital).[60] According to Asperger, these children demonstrated, "a lack of empathy, little ability to form friendships, one-sided conversations, intense absorption in a special interest, and clumsy movements." He referred to them as "little professors" due to their ability to talk at great length on these subjects of special interest, as though lecturing a class.[61]

Degrees of Separation

Dr. Wing's advocacy influenced further research into autism and related childhood developmental disorders, resulting in the formal inclusion of Asperger Syndrome, or AS, in the fourth edition of American Psychiatric Association's Diagnostic and Statistical Manual of Mental Disorders (DSM-IV). Due to similarities with autism, a disorder characterized by communication and social interaction impairments and a tendency toward repetitive behavior, AS was included within the "autism spectrum," but, unlike clinical autism, those diagnosed with the syndrome were considered highly functional, capable of living without supervision and overcoming many of their initial social and motor-skills difficulties over time. Investigations into possible genetic and physiological factors required more precise diagnostic criteria, which were further developed in 1989 by Swedish psychiatrist Christopher Gillberg. He defined six categories, with severe

[60] Attwood, *The Complete Guide to Asperger's Syndrome*, 35.
[61] Ibid., 11.

impairment in reciprocal social interaction being the most prominent. Evidence of at least four of the five remaining—all-absorbing narrow interests, imposed routines, speech and language problems (including prosody, unusual syllabic stresses and tonal speech patterns), difficulty with non-verbal communication, and motor clumsiness—in children around the ages of two or three years old, were regarded as sufficient evidence of the disorder.[62] Other diagnostic criteria were developed, but among practitioners Gillberg's work was largely regarded as authoritative. With few modifications, including the requirement that no unrelated disorder symptoms be present, these criteria were included in the 2000 publication of DSM-IV, and by 2015 research published in the British medical journal The Lancet estimated that over 37 million people globally were affected by AS.[63]

As a pervasive developmental disorder, autism and conditions in the autism spectrum are assessed primarily in children and early adolescents. Adult diagnoses can be difficult, requiring slightly different criteria including difficulty understanding social situations and a tendency toward inflexible, black-and-white thinking in areas of politics and morality. A general disinterest in social pleasantries, such as small-talk and general conversation is common, along with a tendency to engage in pedantic over-explaining and steering conversations toward preferred topics. Some clinicians estimate that women are significantly less likely to be diagnosed, with women representing only 20% of all patients. Subsequent research suggests that the actual number of female "Aspies," as many of the diagnosed prefer to be called, is much higher, but girls are more likely to develop coping skills which conceal their impairment. Adult women seeking diagnosis may close the gender disparity gap to around 30%, according to some psychologists.[64] Cambridge

[62] Ibid., 37.

[63] GBD 2015 Disease and Injury Incidence and Prevalence Collaborators, "Global, Regional, and National Incidence, Prevalence, and Years Lived with Disability for 310 Diseases and Injuries, 1990-2015: A Systematic Analysis for the Global Burden of Disease Study 2015."

[64] Attwood, *The Complete Guide to Asperger's Syndrome*, 46.

University's Simon Baron-Cohen, one of the leading clinical researchers into autism, maintains that the spectrum disorders are linked to the X chromosome, and are manifestations of what he refers to as extreme forms of the "male brain," a theory scientist and prominent autism spokesperson Temple Grandin supports.[65]

Theories aside, the general consensus on AS appeared reasonably established. But in 2013 the subsequent edition of the Diagnostic and Statistical Manual, DSM-V, removed Asperger Syndrome from its list of diagnoses. Variations of the condition that significantly impaired day-to-day functioning were retained, but folded into the more general category of autism spectrum disorders. Severe autism in children was well understood, but the relationships between AS, so-called "high functioning" autism, and other unspecified pervasive developmental disorders were difficult to establish conclusively. Researchers speculated that what few patterns were discernible could be due to individual differences in intelligence.[66] The World Health Organization still maintains the diagnosis in its 1994 classification guidelines, but the upcoming 11th edition echoes the practice of combining it into the broader spectrum disorders category, with the distinction that it manifests "without disorder of intellectual development and with mild or no impairment of functional language," a recommendation the U.S. Center for Disease Control follows.[67]

Whatever Hans Asperger's initial conclusions may have been, the contemporary resolution appears to be that the syndrome bearing his name is synonymous with autistic symptoms that are challenging but otherwise not disabling, crippling, or overly disruptive, in individuals with otherwise high intelligence. The developmental aspects may fade over time, including poor language skills and physical clumsiness, but the social deficits, narrow range of interests, and

[65] Baron-Cohen, "The Essential Difference: Men, Women and the Extreme Male Brain."
[66] Witwer and Lecavalier, "Examining the Validity of Autism Spectrum Disorder Subtypes."
[67] World Health Organization, "Autism Spectrum Disorder without Disorder of Intellectual Development and with Mild or No Impairment of Functional Language."

intense concentration persist. According to clinician James McPartland, Aspies at all ages tend to remain "active but odd," their symptoms, as Dr. Wing observed, "shading into eccentric normality."[68]

Disorder and Disability

> I do think there is a benefit in trying to help people with autism-spectrum conditions with areas of difficulty such as emotion recognition. Nobody would dispute the place for interventions that alleviate areas of difficulty, while leaving the areas of strength untouched. But to talk about a "cure for autism" is a sledge-hammer approach and the fear would be that in the process of alleviating the areas of difficulty, the qualities that are special—such as the remarkable attention to detail, and the ability to concentrate for long periods on a small topic in depth—would be lost. Autism is both a disability and a difference. We need to find ways of alleviating the disability while respecting and valuing the difference.[69]

Medical issues are hardly immune from politics, and for some diagnosed with AS the category became regarded as less a disability than a distinction. They chose to view themselves as neurologically different, or "neurodiverse," a term coined by Australian Judy Singer to place Aspies in contrast with the ordinary, "neurotypical" population. Following the social model of disability, a condition is only as disabling as society allows, so the question became one of disability rights, not clinical treatment.[70] Faced with this perception, psychiatric professionals began reconsidering their approach. "The neurodiversity movement challenges the medical model's interest in causation and cure, celebrating autism as an inseparable aspect of identity."[71]

The Neurodiversity movement advocates on behalf of those diagnosed on the autism spectrum, arguing that their neurological conditions aren't medical disorders, deficits, or dysfunctions, but naturally occurring variations in the human genome. For Baron-Cohen, these

---

[68] McPartland and Ami Klin, "Asperger's Syndrome."
[69] Baron-Cohen, in Saner, "It Is Not a Disease, It Is a Way of Life."
[70] Muzikar, "An Interview with Steve Silberman Author of Neurotribes."
[71] Kapp et al., "Deficit, Difference, or Both? Autism and Neurodiversity."

variations may have always existed out of necessity, resulting in people capable of improving the conditions of their societies by virtue of their neurological "gifts." Neurologist Kirk Wilhelmsen, regarding both himself and his son as classifiable within the autistic spectrum, sees the possibility of a cure for autism as potentially tragic, as their differences could prove invaluable in the ongoing evolution of the human race.[72]

One prominent advocate of the neurodiversity cause is Steve Silberman, author of the 2015 historical review *NeuroTribes: The Legacy of Autism and the Future of Neurodiversity*. Although not an Aspie himself, Silberman took up the cause after immersing himself in the technology industries of Silicon Valley, where his experiences with the parents of diagnosed children resulted in his Wired Magazine article "The Geek Syndrome." He became convinced that the offspring of adults potentially within the high-functioning categories (if themselves undiagnosed) were exhibiting similar or more pronounced diagnostic criteria at a significant rate, suggesting a clear genetic component. And, like Wilhelmsen, he began to consider the possibility of medically eliminating autism as potentially harmful, removing "the very abilities that have made them dreamers and architects of our technological future" echoing the position of the Autism Rights Movement in re-conceptualizing the condition as an identity, not a disease.[73] Applied to AS, viewing the diagnosis as a social or cultural construction, bordering on eccentricity, is understandable:

> The philosopher of science Ian Hacking has wonderfully described the phenomenon of diagnostic categories "making up people." A large set of social factors contribute to the popularity of a diagnosis, which in turn creates the lenses through which we see ourselves and others. The growth of autism diagnosis has been set in motion by patient activism, statistical conflation, websites, blogs, and chat rooms, art exhibitions, provision of services specifically for that diagnosis, media portraits, Web and medico/political campaigns, books such as Andrew Solomon's prize-winning *Far*

[72] Silberman, "The Geek Syndrome."
[73] Ibid.

> *From the Tree*, and the attention of talented writer-doctors such as Oliver Sacks. All these combine to create a particular kind of being who, for better or worse, bears an identity that is a medical diagnosis, most often a psychiatric one that at once stigmatizes and destigmatizes.[74]

For parents of children with "low-functioning," extremely debilitating forms of autism, disregarding the possibility of a cure in favor of uncovering their offspring's potential talents and avenues of social contribution, as Silberman suggests, does not represent their perspectives.[75] His application of speculative retrospective diagnoses (such as the Irish Journal of Psychological Medicine's study of Alan Turing, referenced in the introduction) as a means of portraying autism positively, is an approach that troubles parents of severely disabled autistic children seeking a cure or remediation instead of allowances based on identity politics. Autism may be a natural human variation, but it is not of a single variety, and in response to the Neurodiversity movement some medical professionals concluded that it is "still reasonable to include other categories of autism in the psychiatric diagnostics."[76]

By Any Other Name

Although Silberman's ideology may be arguable, his investigation shined a light on a largely unspoken consensus, the proverbial elephant in the room: a large proportion of the technology industry was composed of people who could potentially be diagnosed as falling within the autism spectrum. Bill Gates, though still alive, was a popular target for retrospective diagnosis not only due to his single-minded ability to focus, but also because of the flat tone of his speaking voice, his fits of caustic, intolerant rage in the face of avoidable mistakes, and his tendency to violently rock back and forth while seated at his desk concentrating on a problem. Douglas

[74] Appignanesi, "'Dr. Death,' a Review of Edith Scheffer's Asperger's Children: The Origins of Autism in Nazi Vienna," 34.

[75] Khazan, "Autism's Hidden Gifts."

[76] Jaarsma and Welin, "Autism as a Natural Human Variation: Reflections on the Claims of the Neurodiversity Movement."

Coupland, author of the novel *Microserfs*, was not the first to conjecture that, "all tech people are slightly autistic." As Silberman observed, "Flattened workplace hierarchies are more comfortable for those who find it hard to read social cues. A WYSIWYG [what-you-see-is-what-you-get] world, where respect and rewards are based strictly on merit, is an Asperger's dream.[77]

As for the work itself, Baron-Cohen observed that, "In the social world, there is no great benefit to a precise eye for detail, but in the worlds of maths, computing, cataloging, music, linguistics, engineering, and science, such an eye for detail can lead to success rather than failure."[78] Musician Gary Numan would be among the first to agree; interviewed at an event for the music licensing agency PRS for Music, he enthused that, "My Asperger's is an absolute advantage." When it came to disconnecting emotionally and not taking criticism to heart, as well as enabling obsessive focus, AS "gives more than it takes."[79] Grandin addressed the topic directly in her essay, "Choosing the Right Job for People with Autism or Asperger's Syndrome," stating:

> It is important that high functioning autistics and Asperger's syndrome people pick a college major in an area where they can get jobs. Computer science is a good choice because it is very likely that many of the best programmers have either Asperger's syndrome or some of its traits.[80]

Grandin, when asked about the connection between AS and information technology, went so far as to say, "We wouldn't even have any computers if we didn't have Asperger's!"[81] Software developer John Slegers agrees, and believes technology companies outside of Silicon Valley need to do a better job of recruiting and accommodating Aspies: "Even though it is well-known that Autistic people are often better programmers and engineers than their 'Neurotypical' peers and how companies can nourish this potential, this knowledge has barely permeated the corporate

[77] Silberman, "The Geek Syndrome."
[78] Baron-Cohen, "Is Asperger Syndrome Necessarily Viewed as a Disability?"
[79] PRS for Music, *Gary Numan: "My Asperger's Is an Absolute Advantage."*
[80] Grandin, "Choosing the Right Job for People with Autism or Asperger's Syndrome."
[81] Mayor, "Asperger's and IT: Dark Secret or Open Secret?"

cultures of tech companies worldwide."[82] Others appear to agree, including Redmond, Washington-based Microsoft, where they implemented a Global Diversity and Inclusion Autism Hiring Program, a "multi-day program specific to people on the autism spectrum," that specifically includes programmers, or (as they call them) "Software Engineers."[83] There is even a software testing company in Chicago, Aspiritech, that hires people with Asperger's exclusively.[84]

The trends suggest that, for the high-functioning population, more businesses will begin treating their disorders as routine, above and beyond the technology sector. But employment isn't the only measure of success. The deficit in social skills, in many instances, creates a virtual career ceiling above which many Aspies cannot rise, leaving them stuck in lower, less remunerative positions. Opinions on the professional consequences of AS vary widely. Peter Thiel believes that the lack of attachment to social conventions demonstrated by those diagnosed in the autism spectrum provides the strength necessary to create an innovative business and become a successful venture capitalist. "Many of the more successful entrepreneurs seem to be suffering from a mild form of Asperger's where it's like you're missing the imitation, socialization gene."[85] Programmer, Aspie and AS advocate Roger Meyer, however, believes that even the more dynamic among the diagnosed are challenged to apply their skills broadly. Minimal human interaction and few sensory distractions create a positive environment where Aspies can apply their encyclopedic knowledge and solve problems using efficient if unorthodox methods, but, "As those jobs increasingly become automated and/or outsourced, Aspies' chances for employment are diminished as well."[86]

[82] Slegers, "Why the Tech Industry Needs More Autism."
[83] Microsoft, "Microsoft Global Diversity and Inclusion Autism Hiring Program."
[84] Hazard, "Does Asperger's Make You a Better Software Coder?"
[85] McFarland, "Why Shades of Asperger's Syndrome Are the Secret to Building a Great Tech Company."
[86] Mayor, "Asperger's and IT: Dark Secret or Open Secret?"

Looking Back in Anger

Would an understanding of the neurobiological characteristics of Levy's MIT hackers provide us with greater insight into the culture they developed, and by extension the ethic that inspired Stallman to oppose the direction of a developing software-based industry single-handed? To Silberman, Grandin, and others ideologically aligned with neurodiversity movement, speculative retrospective diagnoses are a means for portraying the autism spectrum in a positive light. Yet characterizing Aspies as exceptional still risks accusations of "ableism," defining people by their disabilities. The characterization may be intended as positive, but in the process people are still assigned or denied abilities, skills, and dimensions. In the absence of expert intervention, care must be taken not to reduce a historical figure, living or dead, into a collection of symptoms, no matter how convenient it may prove when interpreting historical events.

Where the first generation of MIT hackers is concerned, barring adult diagnoses among their survivors, speculation is all that remains. Most were born around the time that Asperger and Kanner first published their observations, and in the early 1960s, before Stallman's tenth birthday, it still wasn't a widely known diagnosis. Children exhibiting behaviors associated with AS were generally considered gifted, precocious, and odd at that time. From the interviews conducted by Levy, there are numerous passages that arouse suspicion when examined through the lens of developmental disorders. Greenblatt's description of his childhood attraction to electronics certainly qualifies. Faced with the possibility of losing complete control over the AI lab's computer, Gosper threatened to physically demolish the machine.[87] The polarized, categorical perspective revealed by the hacker concept of The Right Thing is a hallmark trait of AS, and Levy relates that arguments frequently descended into shouting matches when opinions on specific

[87] Levy, *Hackers: Heroes of the Computer Revolution*, 117.

instances of The Right Thing differed.[88]

Meeting Greenblatt and Gosper in the AI lab, Stallman discovered his place in the world. "For the geeky outcast who rarely associated with his high-school peers, it was a heady experience to be hanging out with people who shared the same predilection for computers, science fiction, and Chinese food."[89] It is not unreasonable to assume that their shared predilections covered more than superficial interests. For Stallman, a public figure, the collected catalog of his recorded eccentricities is impressive. Technology journalist Peter Wayner describes him as, "strident, super intelligent, highly logical, and completely honest," and fundamentally unable to mislead hypothetical business customers or investors.[90] Eben Moglen, legal scholar and personal friend to Stallman, equates his accomplishments directly to his personal traits. "If any person exemplifies the old adage 'What you see is what you get,' it's Stallman. […] All those personal eccentricities that lots of people see as obstacles to getting to know Stallman really 'are' Stallman: Richard's strong sense of personal frustration, his enormous sense of principled ethical commitment, his inability to compromise, especially on issues he considers fundamental. There are all the very reasons Richard did what he did when he did."[91]

Personal impressions aside, Williams' biography includes numerous anecdotes from Stallman's childhood that are strongly reminiscent of high-functioning developmental disorders. His mother, Alice Lippman, along with several childhood friends, describe a boy effortlessly talented in mathematics and applied sciences but socially inept, often manifesting behavioral problems. In his own words,

I did have friends, but I couldn't fit into a school. So I was sent to a private school for

[88] Ibid., 69.
[89] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 49.
[90] Wayner, *FREE FOR ALL: How Linux and the Free Software Movement Undercut the High-Tech Titans*, 50.
[91] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 21.

> people like that. But most of the people there were either insane or stupid, and I was terribly shamed to have been lumped with them. I wasn't just too smart. Some smart people can get along fine with society. I couldn't.[92]

With their fixation on logic and order, Aspie children aren't easily satisfied with decisions based on social or traditional reasoning. Baron-Cohen labels this a preference for "first principles," fundamental rules used to inform decisions. "Someone with autism or Asperger's, they kind of ask those *why* questions. They want more logical answers. Just saying 'Well we do this just because everybody else does,' that doesn't meet their test of logic."[93] Arguing with his mother over participating in family activities, Stallman recalled, "I saw no reason why […] she should be able to tell me what to do, period. Essentially, what I had read about, ideas such as democracy and individual freedom, I applied to myself, I didn't see any reason to exclude children from these principles."[94] Teaching himself calculus at age seven, Stallman "saw little need to dumb down his discourse for adults."[95] Taking advanced science classes at Columbia University while still a high school student, Stallman, "would catch some mistake in their lecture. And he was not shy about letting the professors know it immediately. It got him a lot of respect but not much popularity."[96]

The possible manifestations of developmental AS symptoms were not limited to Stallman's intellect, personality, and social behaviors. His curious verbal style, described as, "odd pauses in his speech that often confuse listeners into thinking he is done with his train of thought," coupled with an unusual tempo, are considered symptomatic.[97] Stallman's gaze is famously intense, to the point of becoming intimidating, and lingering long after his audience breaks eye contact.[98] His mother describes how, at age ten, he wanted to join a touch football game but could not keep up.

---

[92] Gross, "Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-Certified Genius."
[93] McFarland, "Why Shades of Asperger's Syndrome Are the Secret to Building a Great Tech Company."
[94] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 27.
[95] Ibid., 28.
[96] Ibid., 33.
[97] Scott, "I've Seen a Breakdown Video of RMS. Is Richard Stallman Autistic?"
[98] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 59.

"He wanted to play so badly, but he just didn't have the coordination skills," Lippman recalls. "It made him so angry."[99] She also remembers sensory reactions suggesting a heightened sensitivity to color and sound when Stallman was an infant, and, when asked about the possibility, answered "I do feel that Richard had some of the qualities of an autistic child. […] I regret that so little was known about autism back then."[100]

Stallman himself publicly agreed with this assessment, considering himself potentially "borderline autistic" or at least possessing what was sometimes referred to as a "shadow syndrome," a term coined by Harvard psychiatrist John Ratey to describe milder forms of mental disorders. Asked if he considered the traits a positive or negative influence on his life, he responded, "Both […]. It's certainly been a hindrance to me in my romantic life, but other than that, I don't think it has held me back much. At the same time, it may have given me the strength and firmness and thoughtfulness to do something important with my life."[101]

Correlation and Causation

An all-absorbing interest in programming, followed by a tireless campaign for software freedom. Impaired reciprocal social interactions. Formal, pedantic language with odd vocal characteristics. Rigid thinking, early evidence of poor motor skills, easily frustrated and enraged by inefficiency and illogical behavior, and exhibiting intense single-minded stubbornness along with the ability to focus obsessively over prolonged periods. By any application of the Gillberg criteria, it's not unreasonable to assume that Stallman is an undiagnosed high-functioning autist falling within the Asperger Syndrome end of the spectrum while possessed of a genius-level intellect. Even if such an assumption is accurate, is it significant? Is the existence of the Free Software

[99] Ibid., 28.
[100] Ibid., 30.
[101] Tennant, "Asperger's Oxymoron."

movement, an ideology capable of shaping a global society by opposing economic and legal control over an industry, ultimately the result of a poorly understand neurobiological condition? Is, as Temple Grandin suggested, an entire composite industry, built on mathematics and engineering, essentially the result of people born with this condition?

Stallman's rise to prominence occurred at a time when the nature of autism spectrum disorders became more widely known, and, when combined with his singular nature, placed the possibility in high relief. Speculative diagnoses, even of the not-so-retrospective variety, such as those applied not only to Bill Gates but also Elon Musk (who accepts the possibility), Peter Thiel (who denies it, although he holds the neurodivergent in high esteem), and Mark Zuckerberg (due in large part to his curiously restricted affect), are a popular pastime among technology enthusiasts, but there is no evidence to support the idea that only Aspies can become truly great software coders. After all, many neurotypical people manage that accomplishment every day. Still, Jeff Atwood's observation that, when compared to other professions, programmers are far more likely to find sufficient pleasure in their work as to continue the practice obsessively after working hours, is provocative:

> For better or worse, I spend nearly every waking hour (almost literally) in front of a computer. I have a hard time imagining accountants going home and cracking open their accounting ledger and sliding on their green visor, you know, "for fun". I think other professional fields can be asperger-y, but few as much as computer science.[102]

Considering the state of current medical knowledge, one can only observe the pattern, speculate on its nature, and imagine where it might lead. And sometimes the smallest detail of a pattern catches your eye, like the X-Ray diffraction pattern in Rosalind Franklin's Photo 51 that suggested the base pairs of the DNA molecule had to be internally angled. Stallman's mother,

[102] Atwood, "Software Developers and Asperger's Syndrome."

Alice, described how he overcame his discomfort with loud noises at age seven through his fascination with subway trains, standing at the front window, mentally mapping the system.[103] The MIT hackers all shared a similar fascination, first expressed through their development of complex timetables and mechanical features as part of the Tech Model Railroad Club's Signals and Power Subcommittee. To the staff of transit museums, this fascination is familiar. Their most enthusiastic visitors are boys on the autism spectrum. Perhaps most interesting is that there is no single explanation for their interest. Some may enjoy the systematized nature of schedules and fixed routes, but one parent offered that his son is, "not interested in maps and timetables. What they love about it isn't necessarily the same thing." The autism spectrum is populated with great variety, and the patterns displayed may never be fully understood.[104]

In the end, whatever his other motives and whatever the results, Stallman was initially motivated to reach out, connect, and build a community. As computer and network access became more common in the late 1980s, and the practice of hacking evolved and proliferated, new generations of alienated, socially challenged kids found themselves participating, and a young security hacker going by the pseudonym "The Mentor" expressed his drives. The computer does what he wants it to do. "If it makes a mistake, it's because I screwed up. Not because it doesn't like me… or feels threatened by me…"[105] When the AI lab provided unrestricted access to Stallman's computer over the ARPAnet, he recognized an opportunity:

> [A] lot of those people felt outcast by society—they were geeks; their families and their fellow students didn't understand them; they had nobody. And we welcomed them into the community and invited them to learn and start to do some useful work. It was amazing for them not to be treated as trash.[106]

---

[103] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 30.
[104] Leland, "For Autistic Boys, the Subway Is Actually Soothing."
[105] Blankenship, "The Conscience of a Hacker."
[106] Lucas, "Richard Stallman: Talking to the Mailman," 72.

Expanding the hacker community was something he could support. If not Richard Stallman, the last true hacker, then who?

Chapter 3

The Dictator

> It is the fate of manufactured goods to slowly and gently depreciate as they get old and have to compete against more modern products. But it is the fate of operating systems to become free.
>
> —Neal Stephenson[107]

Richard Stallman's crusade for software freedom, and his corresponding hack of the legal system facilitating this project, depended on the participation of his fellow hackers, the quality of the resulting software, and its adoption by both hackers and general users alike. None of these conditions were guaranteed, and Stallman understood that the GNU Project would not be successful unless the most fundamental component, the kernel, or core set of routines and functions required by the system, was provided with these applications. GNU was always intended to be a complete operating system, compatible with any computer hardware available and supporting any potential application.[108] Underlying this goal is a basic assumption rarely questioned these days: a computer requires an operating system.

This condition seems commonplace today, almost to the point of indifference. In the smartphone marketplace, consumers distinguish themselves by their preference for what a device "runs," alternately a version of Google's Android platform or Apple's iOS ("OS" being the abbreviation for operating system, a detail that may escape a user's notice). As recently as the final decade of the twentieth century software publishers retained the services of so-called "evangelists," enlisted to "spread the gospel" of their technology platform and its virtues. Accepting their operating system and associated hardware was a key tenet of the conversion experience.[109]

---

[107] Stephenson, *In the Beginning Was the Command Line*, 6.

[108] Stallman's project goals were outlined in his original public announcement, and in "The GNU Manifesto."

[109] Former Apple Computer marketing director Guy Kawasaki credits his Apple colleague Mike Murray with inventing the term "software evangelist" in his book *The Macintosh Way: The Art of Guerilla Management*, 2.

Even among the non-technical, it became widely understood that an operating system was a necessary requirement for using software applications, an intermediary between software and hardware that manages memory and storage, and is ultimately responsible for the interface between the user and the machine. Certain features, such as time-sharing and multitasking, were sufficiently revolutionary as to provide a competitive advantage.[110] As recently as 1999, science fiction author Neal Stephenson could still remember a time when the success and pervasiveness of the concept would be met with astonishment:

> A computer at least had some sort of physical reality to it. It came in a box, you could open it up and plug it in and watch lights blink. An operating system had no tangible incarnation at all. It arrived on a disk, of course, but the disk was, in effect, nothing more than the box that the OS came in. The product itself was a very long string of ones and zeroes that, when properly installed and coddled, gave you the ability to manipulate other very long strings of ones and zeroes. Even those few who actually understood what a computer operating system was were apt to think of it as a fantastically arcane engineering prodigy, like a breeder reactor or a U-2 spy plane, and not something that could ever be (in the parlance of high-tech) "productized."[111]

How does an abstraction evolve from the esoteric, to the contentious, to taken for granted? Is it fundamental to the nature of a computing device that an intermediary layer of software, an operating system, need exist, or is it just an evolutionary stage? Somewhere in the history of computing technology, a software "family" became a battleground, and, for some, a source of revolutionary heroics. A new regime appeared, destined to fulfill the bell-curved promise: from freedom, to empire, to freedom again. In the process, the effects rippled outward into the surrounding non-technical culture, through the rhetoric of a new development model, and the selection of an unlikely "dictator."

[110] The introduction of interactive time-sharing in the 1960s, a means for dividing computing resources among multiple users, was a significant advance from the earlier "batch processing" model, and boosted the popularity of early versions of AT&T Bell Labs Unix among programmers according to co-creator Dennis M. Ritchie, "The Evolution of the Unix Time-Sharing System."

[111] Stephenson, *In the Beginning Was the Command Line*, 1.

The Imitation Machines

The Industrial Age gave rise to many technological breakthroughs, including improvements in materials engineering and high-tolerance standardized manufacturing. It was only matter of time before the concepts behind Charles Babbage's proposed analytical engine, a mechanical general-purpose calculating device capable of applying custom instructions toward solving diverse problems, were pursued in earnest. An increasingly complex world, actively engaged in military conflict at a previously unheard of scale, provided strong incentives for gaining a competitive advantage in the processing and manipulation of information. But, in the absence of an existent paradigm, mechanical, electrical, or otherwise, mathematicians and engineers had only a limited understanding of the real potential of these systems.

One approach to establishing a computing device's potential entailed developing theoretical, abstract machines. These abstractions were attempts at understanding just exactly what could and couldn't be "computed," independent of any mechanical implementation. Still used as teaching tools, they embody mathematical attempts at answering the question of just what questions could in fact be answered using mathematical logic. The virtual machines, or automata, are described as "thought exercises," demonstrating the series of instructions (the algorithms) required to produce a result (the output, or answer) based on a given set of inputs (the question). A common computer science classroom exercise is to ask, "Can a given automaton identify a string of symbols as a palindrome, equivalent when read in reverse order?" At each stage the machine must answer "yes" or "no" in response to the input, in effect deciding if that input is provably true based on the rules of logic inherent in its design. Although interesting as an exercise in logic, the foremost goal was to identify which problems are *decidable*, a question initially raised by Gottfried Leibniz, the co-discoverer of calculus, when he imagined creating a machine that operated by

manipulating symbols in the service of evaluating mathematical statements.[112] He wished to simplify the process of determining if such a statement was true via automation, a goal that was reframed by German mathematician David Hilbert in 1928 as the *Entscheidungsproblem*, or "decision problem."[113]

This hypothetical machine technique was the approach adopted by Alan Turing in 1936.[114] To determine if a machine could be created capable of solving any potential mathematical problem, Turing worked his way up from simplicity to greater complexity, from designing simple, single-purpose automata to conceptualizing something he called the "α-machine," or automatic machine, capable of reading a symbol on an infinitely long tape of sequential symbols (be they numbers, letters, or self-defined glyphs), writing a new symbol (if desired), and either advancing the strip backwards or forwards in response to the symbol based on the instructions provided, a.k.a. the algorithm determining the machine's operation. Under some conditions it might also halt in response to its input, and simply stop working. Other inputs might result in an infinite loop of reading and advancing.

Turing's genius was in expressing the decision problem in a form that was relatively easy to visualize: Is there a machine that can be programmed to determine when it will not be able to answer a question, or more specifically make a decision? Beyond this, what can a computing machine actually decide? Turing's automaton, later referred to as the "universal Turing machine," offered proof that there are in fact problems that computers simply cannot solve, first among them the ability to identify which problems they are incapable of solving! But as for the second part, Turing demonstrated that any problem that can be expressed using an algorithm can be computed.

---

[112] Davis, *Engines of Logic: Mathematicians and the Origin of the Computer*, 3.

[113] Hilbert and Ackermann, *Grundzüge Der Theoretischen Logik*.

[114] Turing published his paper on the subject, "On Computable Numbers, with an Application to the Entscheidungsproblem," while a fellow at King's College, Cambridge.

In fact, the definition of an algorithm evolved from a recipe, a sequential set of rules used to calculate a problem's solution, into anything that can be accomplished using a Turing machine. Conversely, a Turing machine is any automaton that can perform computations using any arbitrary algorithm.

Turing wasn't the only person to provide a formal proof of the decision problem. Alonzo Church, working in Princeton, New Jersey, reached the same conclusion using his lambda calculus, a system for writing abstract logical expressions when evaluating functions and their effects on atomic logical values, an alternative and more traditional approach than automata.[115] In 1936 they compared notes during Turing's studies at Princeton, and agreed that their work supported each other's research – one consequence of which is the commonly accepted contemporary practice of referring to it as the Church-Turing thesis.[116] Church's formal, mathematical approach proved applicable to a variety of inquiries, including type theory studies in symbolic logic, and provided inspiration for one of the few programming languages considered beautiful, John McCarthy's LISP.[117] Turing's automata are closely related to the formal grammars defined by linguist Noam Chomsky, modeling the valid syntax of a language and giving rise to the "regular expression" pattern-matching technique.[118] Turing's approach, however, also resulted in those unambiguous, essential concepts central to the development of the modern computer.[119]

[115] Church, "An Unsolvable Problem of Elementary Number Theory."

[116] Mathematician Andrew Hodges describes Turing's infrequent yet collegial interactions with Church during his studies at Princeton, in his 1983 biography *Alan Turing: The Enigma*.

[117] In 1960 MIT Artificial Intelligence researcher McCarthy introduced LISP, a "programming system" based on lambda calculus, in his article "Recursive Functions of Symbolic Expressions and Their Computation by Machine."

[118] Chomsky, "Three Models for the Description of Language."

[119] Mathematician John von Neumann met Turing while attached to Princeton's Institute for Advanced Study, and subsequently described Turing's role in the development of the modern computer as second only to Babbage. Copeland, *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life: Plus the Secrets of Enigma*.

The theory of a universal stored-program computing machine soon gave way to its reality, but the operation of these devices was far from standardized. To be reprogrammable, affordances suitable for reprogramming needed to exist, and these typically took the form of cables and mechanical switches. The first general-purpose computer, John Mauchly and J. Presper Eckert's Electronic Numerical Integrator and Computer, or ENIAC, required weeks' worth of planning and the physical reorganization of components when applied to a different, distinct programming task.[120] Likewise, the original operating system was a collection of human beings—six female human beings, specifically. A team of women was recruited, as part of the war effort, to map questions to hardware, and rearrange the components into suitable sequential arrangements. Debugging likewise required physical intervention, manually locating broken connections and faulty components inside the refrigerator-sized machines.[121] Although the association would soon become overwhelmingly male, in the decades following World War II the word "computer" was synonymous with "woman" in the public imagination.[122]

For a computer to then become both useful and manageable, developing a means for supporting the "stored-program" model was necessary, in which both the data and the program itself are placed in a computing device's memory, obviating the need to modify its physical configuration. Disk drives and nonvolatile RAM replaced punch cards and magnetic drums, while mechanical switches, electromagnetic relays, and vacuum tubes gave way to solid state electronics following William Shockley's advances in semiconductor transistors at Bell Labs.[123] Operations in mathematical logic could now be expressed through pulses of electricity applied to sequentially

[120] Copeland, "A Brief History of Computing: ENIAC and EDVAC."
[121] Fritz, "The Women of ENIAC."
[122] Light, "When Computers Were Women," 458.
[123] Riordan and Hoddeson, *Crystal Fire: The Birth of the Information Age*, 143.

arranged crystals of silicon. By the 1960s the second generation of computers began.

Swords into Plowshares

Having advanced to the point where machines of great intricacy, capable of processing millions of instructions at record speeds and rearranging themselves into new configurations, were storing, printing, and displaying the results of complex calculations, the men behind the curtain stepped into the spotlight. As part of the war effort, first-generation machines were funded by military agencies, solving logistical and intelligence problems in partnership with academic researchers.[124] With a new international order in place following the war, technology became a primary line of defense, and the military-industrial complex President Dwight Eisenhower openly cautioned against began driving research initiatives.[125] Eisenhower hoped that new developments in science and technology could provide benefits beyond military application, and ARPA, the Advanced Research Projects Agency, soon took on that mission. In 1963, working in partnership with private industry, ARPA (now DARPA, the Defense Advanced Research Projects Agency) funded "Project MAC," soon to become the Massachusetts Institute of Technology's Artificial Intelligence Laboratory (and future home of Richard Stallman). No longer limited to missile defense, DARPA branched into information processing.[126]

As a relative novelty, the operating system was particular to the machine and supplied by the vendor: commercial manufacturer IBM's 1964 "mainframe" business computer line, the 360, was a design concept that came with OS/360, the operating system for any machine built according to IBM's 360 specification (although owners of these systems occasionally designed and ran

[124] ENIAC was specifically commissioned by the United States Army's Ballistics Research Laboratory, as documented by historian William T. Moye in "ENIAC: The Army-Sponsored Revolution."

[125] Eisenhower, "Farewell Address to the Nation."

[126] Chiou et al., "A Marriage of Convenience: The Founding of the MIT Artificial Intelligence Laboratory."

alternatives).[127] One of DARPA's early projects was a call for improved features and hardware-independent standards. Under the guidance of ARPA director J. C. R. Licklider, a psychologist by training and the visionary behind the global Internet and the graphical user interface, the Artificial Intelligence Laboratory, General Electric, and AT&T's Bell Labs came together to develop a new operating system concept, the Multiplexed Information and Computing Service, or MULTICS.[128] Instead of large machines dedicated to a limited set of tasks, MULTICS was intended to be a next-generation, portable, multi-user system capable of time-sharing and "utility computing," simultaneously processing requests for a large, remotely scattered user community.

It was here that two computing researchers from Bell Lab's Murray Hill, New Jersey office, Ken Thompson and Dennis Ritchie, found both frustration and inspiration. Inspiration from the MULTICS project's many exciting and promising ideas, but frustration due to their employer's decision to withdraw from the project in 1968 based on suspicions that its ambitious nature would not result in a usable product.[129] The telephone company connected calls using hardware, specifically physical switches, but the possibility of a computer-based approach was considered worth investigating. With the relationship severed, Thompson began implementing many of the MULTICS concepts from scratch, using the machine-level assembly language instructions of the processor in their laboratory's Digital Equipment Corporation PDP-7 "minicomputer." Writing processor instructions was a tedious, painstaking process, so Thompson and Ritchie began developing higher-level programming languages to relieve them from specifying low-level details, and when it came time to migrate their work to a new machine, the DEC PDP-11, Ritchie completed the C programming language, ensuring their new system became portable.

---

[127] Computer History Museum, "IBM System/360."

[128] Licklider's vision of networked, interactive computers was encapsulated in his article, "The Computer as Communications Device," 21.

[129] Warford, *Computer Systems*, 460.

Reimplementing their new operating system's kernel in C made it possible to target and compile the instructions into other computer architecture "dialects," and as new features and tools were added, a new system was born.[130] In 1983 Thompson and Ritchie won the Association for Computing Machinery's Turing Award for their contributions to generic operating systems theory, based on the development of their pet project, named "Unix" in playful homage to its predecessor.[131]

Born to be Wild

> On the political front, the Department of Defense relaxed its notion that only government, academic and research sites could connect. This was partially the result of recognizing the expansion of private networks (like IBM's VNet and Prodigy) and of networks distributing news and mail (UUNET, Bitnet, Fidonet), as well as the recognition that the network of networks was already vastly larger than the US and its "allies." On the "engineering" side, the advent of the desktop machine and the commercial modem meant that individuals could have a computer at home and plug in to their telephone lines. All for under $3000![132]

Unix provided both system-wide supervisory control functions and a common platform manifesting an elegant design philosophy, one that enabled programmers to focus on their work and forget about the operating system, more or less, while using it as a free utility. Somehow, ten years later, Unix escaped the corporate research lab in New Jersey and become widespread enough that a young girl in Stephen Spielberg's 1993 blockbuster film *Jurassic Park* could look at a computer terminal and say, "It's a Unix system! I know this!"[133]

As a result of a federal antitrust suit, AT&T and the Bell operating companies were bound by a 1956 consent decree, preventing them from leveraging their monopoly status to compete in

---

[130] Ritchie, "The Development of the C Language."
[131] van Vleck, "Kenneth Lane Thompson - A.M. Turing Award Laureate."
[132] Salus, *The Daemon, the Gnu & the Penguin*, 196.
[133] In the film, the character Lex Murphy discovers a terminal running the "fsn" graphical file manager, developed for Silicon Graphics' Irix version of Unix, https://www.linux.com/news/my-workstation-os-irix.

related technology markets.[134] Having published a description of their system, Ritchie and Thompson delivered a paper on their work at a 1973 ACM symposium, and whenever the source code was requested, Thompson would distribute tapes for the cost of materials and shipping. This proved very popular among researchers at his alma mater, the University of California, Berkeley. While on sabbatical there in 1975 he installed a copy on the Computer Science department's PDP-11, and his colleagues soon began developing new features and porting the system to new machines, sharing their work with the rapidly expanding community of Unix users. Another decade passed, and the Department of Justice broke up the Bell telephone system completely, releasing the individual component businesses from their previous collective commitments. The newly independent AT&T was now free to capitalize on Unix. Berkeley's unlicensed, communally-developed variants were legally required to remove any copyrighted code, and a series of competing businesses fractured the Unix market into incompatible and expensive variations.

The genie, of course, wouldn't return to the bottle, and the hacker community naturally embraced free Unix. This was perfectly acceptable to its authors, as the communal hacker ethos was present in its beginnings:

> What we wanted to preserve was not just a good environment in which to do programming, but a system around which a fellowship could form. We knew from experience that the essence of communal computing, as supplied by remote-access, time-shared machines, is not just to type programs into a terminal instead of a keypunch, but to encourage close communication.[135]

It should come as no surprise that Richard Stallman was familiar with Unix, and that he would base his new Free Software project on reimplementing it that same year.[136]

---

[134] Pinheiro, "AT&T Divestiture & the Telecommunications Market," 305.

[135] Ritchie, "The Evolution of the Unix Time-Sharing System," 1577.

[136] "I chose that design to follow because it was portable and seemed fairly clean. I was never a fan of Unix; I had some criticisms of it too. But it was ok overall as a model." Stallman, "How I Do My Computing."

Our Man in Helsinki

These Unix variants soon forked—split into different, somewhat incompatible versions—and those released under the liberal Berkeley Software Distribution licenses further evolved into different niches, following different trajectories.[137] One in particular, dubbed "MINIX" by its creator, proved a very effective teaching tool. In 1987 Andrew Tanenbaum, an American lecturing in the Netherlands on operating system design, provided the source code to MINIX along with purchased copies of his textbook, under a custom license that prohibited redistribution.[138] What made this Unix particularly attractive to students and hobbyists at the time was compatibility: MINIX was designed for the recently released IBM personal computer and its Intel microprocessor. Although such machines were not inexpensive, they were considerably more affordable than the Apple line of home computers, and available for purchase from a wide range of manufacturers. Bill Gates' business model of leveraging his operating system against IBM's hardware standards not only made Microsoft one of the wealthiest companies in the world, it unwittingly sowed the seeds of their eventual downfall. Stephenson's prediction was on track: Gates had somehow convinced people to purchase something they couldn't touch, but a proprietary, closed operating system was a contradiction in terms, destined to be marginalized.

In the spring of 1990 a twenty-one year-old man named Linus sat in the darkened bedroom of his mother's apartment in Helsinki, windows covered in blackout curtains, admiring his new, generic gray Intel-based personal computer. A university student, he had recently discovered Unix, the "platonic ideal" of operating systems, and was preparing to embark on an exciting adventure.

> Now everybody has a book that has changed his or her life. The Holy Bible. *Das Kapital*. *Tuesdays With Maury*. *Everything I Needed to Know I Learned in Kindergarten*. […] The book that launched me to new heights was *Operating*

[137] Anonymous, "BSD License Definition."
[138] Tanenbaum, "Some Notes on the 'Who Wrote Linux' Kerfuffle."

Systems: Design and Implementation, by Andrew S. Tanenbaum.[139]

Self-professed nerd Linus Benedict Torvalds, although seventeen years younger, had many things in common with Richard Stallman, including parents that divorced when he was a child, being raised primarily by his mother (while actively disliking his father), and a natural facility with mathematics and physics.[140] Torvalds shared much in common with hackers in general, such as an obvious and obsessive fascination with computer programming. His maternal grandfather, a mathematician, noticed Torvalds' immersion when using a calculator and correctly guessed that he would enjoy entering simple programming instructions into the Commodore VIC-20 he used for graphing arithmetic expressions.[141] Also like Stallman, Torvalds requested and enjoyed reading books on systems programming, which is how he learned of the existence of machine languages, instructions specific to computer hardware that operate between the higher-level languages, like Dennis Ritchie's C or the Commodore's installed version of BASIC, and the shunting of electrical pulses between the processing chip and its adjacent memory register chips.[142] This discovery, and the pleasure Torvalds experienced when exploring and modifying the computer, was the modest beginning of a passion that revolutionized the technology industry and propelled Unix, by way of another evolutionary leap, into becoming the de facto operating system of the twentyfirst century.

Throughout his autobiography there is reasonable evidence to suggest Torvalds shares other attributes in common with Stallman and his hacker colleagues, and, having become a prominent figure in technology circles, he is no stranger to speculative diagnoses. He was an unathletic child, largely due to being born late in the year and thus smaller than most of his schoolmates, but he also had no interest in competition or social activities. The pure logic and sense of absolute control

[139] Torvalds and David Diamond, *Just for Fun: The Story of an Accidental Revolutionary*, 51.
[140] Ibid., 4.
[141] Ibid., 7.
[142] Ibid., 10.

provided by computer systems was a source of pleasure, and, when programming, he exhibited the ability to concentrate and focus for extended periods of time, exhibiting great anger when interrupted. As a child he lacked sentimentality or empathy, and little memory for the people he interacted with, going so far as to suggest that he was "face blind," unable to recognize people he already knew by sight. Torvalds would focus instead on his narrow range of interests, to the point of becoming oblivious to his surroundings.[143] It wasn't long after his tenth birthday that he began spending most of his time alone in his room, working on his computer. Describing how she came to understand her son's nature, Torvalds' mother Anna wrote:

> When you see a person whose eyes glaze over when a problem presents itself or continues to bug him or her, who then does not hear you talking, who fails to answer any simple question, who becomes totally engrossed in the activity at hand, who is ready to forego food and sleep in the process of working out a solution, and who does not give up. Ever. He—or she, of course—may be interrupted, and in the course of daily life often is, but blithely carries on later, single-mindedly. Then you know.[144]

Like Stallman, Torvalds' sense of alienation from his peers persisted well into his college years, and, also like Stallman, he remained primarily occupied with hacking when not enrolled in classes. Until he discovered his own local community of peers—not so much hackers as applied science students in a university-sponsored club named Spektrum—he found social interactions challenging and upsetting.[145] Whatever their similarities, Torvalds was sufficiently different as to develop a less rigid worldview and cultivate interests beyond the technical. These differences and their consequences may have been just as critical to the success of the Free Software project as Stallman's unwavering idealism, if not more so.

[143] Ibid., 69.
[144] Ibid., 70.
[145] Ibid., 111.

Baptism of Flame

Having encountered Unix and the C programming language at the University of Helsinki, Torvalds discovered a world of new possibilities. As Steven Levy recounted, the hacker revolution in software sharing migrated across North America to the San Francisco Bay Area, inspiring a generation of hardware hackers into collaboratively developing and opening up the architecture of computers. Advances in semiconductor technology reduced the required form factor specification requirements, from the size of a refrigerator to something that could fit on a single desk, and the age of the personal computer dawned.[146] Ironically, the popularity of these "micro" computers gave rise to the Microsoft Corporation, and a profound but temporary reversal to the culture of software sharing. By the time Torvalds was a young college student the combination of (relatively) affordable microcomputer hardware and the global Internet enabled him to have a Unix of his own, in his darkened room at home.

With his mail-ordered copy of Tanenbaum's MINIX installed, Torvalds soon realized that this operating system, intended as a teaching aid, was deliberately limited—most prominently in its ability to emulate a command-line terminal interface for connecting to another machine over the telephone network. So he begin to design his own emulator, separate from MINIX and written in assembly code, and found the results gratifying enough to continue adding features and improvements, such as the ability to transfer and save files. He soon realized that he was effectively developing a new operating system kernel, providing an interface between the computer and any software routines requiring hardware access. And he began uploading the source code of his new operating system kernel to the university's shared file server, making it available to anyone

[146] Levy describes how Berkeley engineer Lee Felsenstein saw himself bringing the Hacker Ethic to the people, sharing the designs for electronic systems and founding the Homebrew Computer Club, in *Hackers: Heroes of the Computer Revolution*, 153.

interested by means of the Internet. Encouraged by his science club friends, in August of 1991 he posted a message to the international MINIX users' online newsgroup:

> I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).[147]

Torvalds, like many of his hacker peers, was familiar with Stallman and the GNU project at that time, and as his kernel (named "Linux," in keeping with the Unix naming tradition) became more comprehensive he began building a complete system by compiling and incorporating it into the collection of GNU software tools. Much to his surprise, the work proved popular, with fellow hackers from around the globe requesting features and offering to contribute code and support. This increased popularity highlighted the need for a formal licensing arrangement, and here again Stallman's influence proved judicious: his hack of the legal system, the GNU General Public License, matched Torvalds' desire to share his source code under terms that required anyone wishing to distribute it for commercial purposes share their modifications in turn.[148]

Another result of the new Linux kernel's popularity was the attention it attracted among MINIX enthusiasts, including Tanenbaum himself—in no small part because communication was taking place over an Internet newsgroup dedicated to MINIX users. A professional researcher, Tanenbaum created his kernel as an experiment in the "microkernel" design concept, a modular, minimalist architectural approach in which the kernel is limited to passing messages between higher-level software processes and managing virtual memory addresses. Elegant in theory, a successful microkernel design needs to address problems of latency (delays in message transmission and reception) and resulting resource contention (requests introducing changes to the

[147] Torvalds and David Diamond, *Just for Fun: The Story of an Accidental Revolutionary*, 85.
[148] Ibid., 96.

state of the system that other components have not registered). Intended to provide simplicity and increased portability, microkernels of the time suffered performance issues and introduced what Torvalds regarded as "false simplicity."[149] Instead, he created Linux using a "monolithic" approach, in which the components required to manage hardware were compiled as structures within the kernel. Brought to his attention, Tanenbaum found it necessary to publicly outline what he perceived as the failings in Torvalds' work.

The resulting debate over The Right Thing became a classic Internet "flame war," an overreaction based on hacker pride that soon became heated and personal, branching into tangents unrelated to software design and involving language bordering on abuse. Although Tanenbaum and Torvalds eventually agreed to disagree and remain civil, their argument, and the philosophical differences it revealed, provided Torvalds with no small amount of notoriety among the international hacker community of the early 90s. The debate concluded with Tanenbaum asking if Torvalds had no objections to unknown people selling his work, a practice that gained popularity with the rise of Free Software "distributions" built using GNU and the Linux kernel, complete with source code. "I just sent [Tanenbaum] an email back saying Yes, and I haven't heard from him since," Torvalds mused.[150]

Given Stallman's goal of creating a complete free-as-in-freedom operating system, and Torvalds' stated intention of developing his system as a hobby, how did their efforts coincide? Releasing the Linux kernel under the GPL was a significant factor, but a miscalculation on Stallman's part also contributed. Tanenbaum was not the only developer researching microkernels: "Mach," a microkernel project at Carnegie Mellon University, was considered the leading

[149] Ibid., 99.
[150] Ibid., 112.

candidate to replace the established BSD Unix kernel design at that time, and the GNU project elected to pursue a similar approach, a microkernel named GNU Mach. The resulting system suffered delays and performance issues similar to MINIX, leaving the GNU system without a vital component until Linux became available. According to Stallman, "We'd started developing a kernel in 1990, but the design I chose turned into a research project—it took six years to get a test version. So Linux is the kernel that we actually use with GNU, for the most part."[151] The Mach architecture eventually developed to the point of practical application, and now forms the foundation of Apple's XNU kernel, powering their desktop and mobile operating systems. In his poem *La Bégueule* Voltaire cautioned that "the best is the enemy of good," and in the early 90s, just as the GNU project supplied the missing piece of Linux—the share-and-share-alike license—Linux supplied the missing piece of the GNU system, a kernel that was good enough, and destined to get even better.

Will Success Spoil Free Software?

Creating a critical software component is an impressive accomplishment, but if that were Torvalds' only contribution the current technology landscape would be very different. Some additional quality was necessary to advance the GNU project from an idealistic hacker proposal to a platform capable of widespread adoption, supporting entire industries. "It would take the Linux kernel project to transform Free Software from a 'mad man's dream' into a large-scale movement with thousands of contributors, many of whom would eventually commit to—and alter—the legal and ethical principles of Free Software."[152] The Linux kernel project, in turn, could distinguish itself from GNU in two distinct areas: a talent for identifying the most effective solution to a given

[151] Lucas, "Richard Stallman: Talking to the Mailman," 77.

[152] Coleman, "Code Is Speech: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers," 424.

problem, and a distaste for ideological idealism.[153]

Sharing software informally was a common practice before the Copyright Act of 1976, and, with the fragmentation of Unix systems into commercial and free variants, new software copyright licenses began to proliferate. The free BSD Unix versions adopted a permissive "4-clause" license (later reduced to as few as two clauses), that differed from the GPL in that it did not impose any conditions or obligations beyond preserving the original copyrights.[154] The beneficiary of their source code was free to modify and conceal the changes as they wished, a condition of complete freedom—including a negative freedom, freedom from coercion—that reflected a libertarian political philosophy.[155] Torvalds, it should be noted, experienced no coercion when deciding to use software released by the GNU project, nor was he coerced into applying the GPL to his kernel. He wanted to share his work, under the condition that any contributions or modifications would be shared with him in turn, and he realized that the General Public License was the best method for achieving his aim. Ironically, prominent libertarian John Gilmore, cofounder of the Electronic Freedom Foundation and one of Levy's second generation cryptography, or "crypto," hackers, initially adopted a BSD license for his software projects only to discard it in favor of the GPL when was unable to incorporate external improvements.[156]

Torvalds' kernel, and the ease of integrating it, along with the GNU tools, into a complete, free, Unix-like system powering a personal computer, was a revelation to a new generation of hackers around the world, attracting attention and participation in record numbers. One such

---

[153] Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, 9.

[154] Text of the original 1990 BSD license, Regents of the University of California, "License: BSD-4-Clause."

[155] Coleman and Alex Golub, "Hacker Practice: Moral Genres and the Cultural Articulation of Liberalism," 269.

[156] Gilmore's FreeSWAN project, in Wayner, *FREE FOR ALL: How Linux and the Free Software Movement Undercut the High-Tech Titans*, 61.

admirer was Eric S. Raymond, a programmer living in Pennsylvania who studied Torvalds' methods and applied them to his own projects. In the process he formulated "Linus' Law," an aphorism describing the rapid advances in quality experienced when using a collaborative development method: "Given enough eyes, all bugs are shallow." Raymond also recognized that Linux represented a new phenomenon, a software development project distributed globally over the Internet that managed to coordinate and incorporate contributions successfully and effectively, due in large part to the discerning judgment and personal style of the project's founder. Torvalds was hardly the first hacker to open his software to external participation: Stallman, among others, adopted a similar approach, paving the way with the Emacs text editor at MIT.[157] "I think Linus's cleverest and most consequential hack was not the construction of the Linux kernel itself, but rather his invention of the Linux development model," Raymond wrote, characterizing it as a "bottom-up" model reminiscent of an open marketplace, in contrast to the rigid "top-down" approach of commercial software engineering.[158] His collected observations formed the basis of *The Cathedral and the Bazaar*, intended in part as an expansion on the concepts proposed by Frederick Brooks in his 1995 book *The Mythical Man Month: Essays on Software Engineering*.

Raymond, another outspoken libertarian, considered the Linux hackers a collection of self-interested agents motivated by personal gain and ego satisfaction. He regarded the absence of a dedicated managerial class as proof of his theory, and the project's success reflected the libertarian economic principle crediting self-organization as the optimal behavior of the marketplace. In contrast to the earlier, compact hacker communities of MIT, Stanford, Murray Hill, and Berkeley, the distinguishing novel element contributing to Linux's revolutionary growth was the absence of

[157] Hal Abelson, in Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 86.
[158] Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, 6.

geographic limitations. "Harnessing the power of the Internet results in great hacks."[159]

Raymond's pragmatic analysis of Torvalds' success was well received, particularly among the software and technology professionals of the San Francisco Bay Area. By the late 1990s, this was a group that evolved in the shadow of Bill Gates, influenced by the earlier generations of DIY hardware hackers and aggressive Stanford University engineering entrepreneurs. Dubbed "Silicon Valley" after the established semiconductor industries built by William Shockley, William Hewlett, and David Packard, this later generation was further influenced by the original MIT hacker ethic, imported directly to Stanford in the person of the AI lab's John McCarthy. One entrepreneur, Marc Andreessen, built a company around a software application for accessing the World Wide Web, Tim Berners-Lee's research project into exchanging documents over the Internet. His Netscape web browser was successful enough to attract Microsoft's notice, resulting in its gradual marginalization as a commercial product, and a corresponding anti-trust lawsuit filed against Microsoft by the federal government. Andreessen sold the remains of his company to AOL, and Netscape's former hackers were inspired by Raymond to publicly release the source code of their browser under a permissive license.[160]

Another admirer of Raymond's work was Tim O'Reilly, a publisher of computer manuals who recognized, in the face of the Internet's Schumpeter-style "creative destruction" of the publishing industry, the opportunity to expand his business to the Web.[161] O'Reilly, a fellow libertarian, published Raymond's book, and joined him and several other like-minded colleagues in

[159] Ibid., 22.

[160] Cohen, *The Know-It-Alls: The Rise of Silicon Valley as a Political Powerhouse and Social Wrecking Ball*, 95.

[161] Austrian economist and Harvard professor Joseph Schumpeter is widely remembered among Silicon Valley technologists for his 1942 book *Capitalism, Socialism, and Democracy*, in which he described the Marxist concept of the internal destruction and recreation of wealth as an evolutionary process of innovation, reinventing industries dynamically and, in the process, contributing to the continued survival of capitalism as a viable economic system.

creating a new advocacy organization following the release of the Netscape code. They labeled their cause "open source software," to distinguish it from Stallman's Free Software, and formed the not-for-profit Open Source Initiative (OSI) to promote software licenses corresponding with their values.[162] To the delight of Raymond and his colleagues, Torvalds expressed his approval of the term "open source," preferring it to Free Software, and publicly supported their organization. In many respects, this casual decision had an incredible impact on the evolution of computer technology. Just as Bill Gates' refusal to participate in the culture of software sharing put Microsoft on the path toward becoming a multi-billion dollar industry, Torvalds' disinterest in (and eventual opposition to) the idealistic ethic of Stallman and his Free Software Foundation enabled a wave of entrepreneurs and venture capitalists to reshape the technology landscape in ways that would eventually eclipse Microsoft.

WWLD (What Would Linus Do?)

The emerging popularity of the Linux and GNU systems inspired the creation of ancillary projects, some based around packaging these and other applications into working "distributions" (with source code available on demand) and others dedicated to specific resources, such as web servers or graphical desktop environments. "As these virtual organizations got off the ground in the mid- to late 1990s, Eric Raymond […] sought to refashion the public persona presentation of Free Software to attract business investors. To do so, he replaced the term 'Free Software' with the ostensibly nonideological terminology of 'Open Source Software.'"[163] O'Reilly, sensitive to the value of public relations and the rhetorical utility of selective terminology, was interested primarily in the freedom of software *developers*, not users, emphasizing their freedom to distribute software

[162] Tiemann, "History of the OSI."

[163] Coleman, "Code Is Speech: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers," 425.

on whatever terms they preferred. "Once a term like 'open source' entered our vocabulary, one could recast the whole public policy calculus in very different terms, so that instead of discussing the public interest, we are discussing the interests of individual software developers while claiming that this is a discussion about 'innovation' and 'progress,' not 'accountability' or 'security.'"[164]

If one defines ideology as a framework for making the world intelligible, received or otherwise, at best Raymond could present Open Source as a concept without a *conspicuous* ideology. Stallman himself favored representing Free Software as a form of idealism, and otherwise avoided commingling his cause with other political positions. "The free software movement doesn't require you to have any particular political stand on other issues," he explained.

> Basically, free software combines capitalist, socialist, and anarchist ideas. The capitalist part is: free software is something businesses can use and develop and sell. The socialist part is: we develop this knowledge, which becomes available to everyone and improves life for everyone. And the anarchist part: you can do what you like with it.[165]

For Raymond and O'Reilly, the second, "socialist" component was disagreeable, both to their personal political philosophies and to the corporate audience they wished to reach. Their success was almost immediate: in a five-year period the term "open source" eclipsed Free Software, and the entire operating system became commonly referred to as Linux—the GNU components becoming effectively marginalized in the popular imagination. In 2017 corporate sponsorship of OSI expanded to include Microsoft, a company famously opposed to Free Software and the GPL.[166]

To Stallman, abdicating the core value of his project was never an option. Having

---

[164] Morozov, "The Meme Hustler: Tim O'Reilly's Crazy Talk."

[165] Lucas, "Richard Stallman: Talking to the Mailman," 83.

[166] In an interview with the Chicago Sun Times, then-Microsoft CEO Steve Ballmer said, "Linux is not in the public domain. Linux is a cancer that attaches itself in an intellectual property sense to everything it touches. That's the way that the license works." "Microsoft CEO Takes Launch Break with the Sun-Times."

experienced a communal society built on freedom and experiencing the perceived injustice of its loss, preserving software freedom became his life's mission. Torvalds appreciated the benefits of Stallman's achievements and shared many of the same values, but believing in software freedom as an inalienable right for both users and creators, Stallman's central tenet, was not one of them. At a 1996 Free Software conference Torvalds openly disparaged some of Stallman's technical decisions, to the surprise of many, including Raymond, but the amusement of others:

> For Raymond, the warm reception other hackers gave to Torvalds' comments confirmed a suspicion: the dividing line separating Linux developers from GNU developers was largely generational. Many Linux hackers, like Torvalds, had grown up in a world of proprietary software. They had begun contributing to free software without perceiving any injustice in nonfree software. For most of them, nothing was at stake beyond convenience. Unless a program was technically inferior, they saw little reason to reject it on licensing issues alone.[167]

The new generation of hackers appeared less interested in freedom, preferring to avoid any form of technological self-denial if it interfered with the successful completion of their work. Still, when considering someone as influential as Torvalds there had to be some motivation beyond mere convenience. Outside of MINIX, the BSD Unix variants may not have been available for Intel-based personal computers at that time, but Torvalds was free to participate in their communities knowing it was only a matter of time before additional, unrestricted alternatives were available. What made the GPL attractive, but not the concept of free (as in freedom) software? If not user freedom, what did Torvalds value instead?

A curious detail of Torvalds' childhood is his identity as the Finnish equivalent of a "red diaper baby," a term applied to John McCarthy by journalist Noam Cohen and to hardware hacker Lee Felsenstein by historian Roy Rosenzweig. Torvalds' parents Nils and Anna were both registered members of the Communist Party, and his father, a journalist, was outspoken in his

[167] Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, 161–62.

views, even holding public office. As a boy Torvalds suffered some teasing and discrimination due to this affiliation. According to his Nils:

> [H]is son made a constant effort to distance himself from the left-wing rhetoric that was the backdrop of his childhood. "He wouldn't let me discuss it, he would leave the room. […] Or else he always made a point of having an opposing view. I know Linus was teased in school for having the wrong father. The message to me was 'Don't put me in this awkward situation.'"[168]

One can only speculate on how greatly this influenced Torvalds' views, but in his autobiography he plainly states that attempting to impose one's "world views" on others is unacceptable.[169] Torvalds resents Stallman's "black and white" thinking, or any attempt at imposing morality on software. It becomes apparent that he regards software, and technology in general, as politically neutral, echoing the instrumentalist view, and considers his personal freedom as the leading consideration. Removing his ability to choose through the process of "institutionalizing morals" is a great offense according to Torvalds, suggesting that the negative freedom of freedom from coercion is a point of common ground he occupies with his fellow open source advocates. He admit to admiring Stallman and his followers, respecting their strong convictions, but would prefer "they keep these opinions to themselves."[170]

In the absence of a system of moral or ethical beliefs, Torvalds sees enlightened self-interest as the motivating factor: allowing people to pursue their interests will generate optimal self-organization, and all conflicts can be resolved by the quality of the results. Quality, as an indication and measure of merit, is the final arbiter.[171] For the self-selected participants, the highest reward is the esteem of one's peers, social reputation.[172] In a voluntary, non-compulsive

[168] Torvalds and David Diamond, *Just for Fun: The Story of an Accidental Revolutionary*, 66.
[169] Ibid., 193.
[170] Ibid., 196.
[171] Ibid., 121.
[172] Ibid., 122.

environment such esteem generates trust, which is the fuel powering a successful project's continued momentum. In the absence of software quality, earned merit, social esteem, and corresponding trust, a collaborative development project could end, fork into different projects, or assume new leadership.[173] All very pragmatic, no ideology required.

Torvalds expresses sympathy for the "creative destruction" view of market evolution, blaming greed, and the corresponding desire to maintain revenue streams from obsolete industries, as harmful. In his view, pursuing quality instead of profit is the path of wisdom, as the results will provide a competitive advantage once the markets are free of artificial distortions, impediments which are inevitably unsustainable and destined to fail in his view.[174] Behind these economic theories is an overarching belief in a single fundamental force motivating human existence. As the title of his autobiography suggests, Torvalds believes the best reason to engage in any activity is, "just for fun."

More specifically, Torvalds argues that survival is the original motivation, but that once survival is reasonably assured people, being social by nature, pursue status, sometimes at the expense of safety and survival. The final objective is entertainment, and these three motivating factors are addressed in sequence, each one following the achievement of the other, with entertainment, or "fun," the most advanced. "Survive. Socialize. Have fun. That's the progression."[175]

An obvious comparison to Torvalds' philosophy is psychologist Abraham Maslow's five-level hierarchy of needs, ranging from physical preservation to the fourth need, self-actualization,

[173] Ibid., 190.
[174] Ibid., 215.
[175] Ibid., 246.

but stopping short of the final need for spiritual transcendence.[176] From an external perspective, Torvalds' hostility to ideology, in favor of projecting his personal motivations on humanity as a whole, suggests an insular, unexamined sense of privilege. Assessing his actions and choices, he appears to share the traditional liberal political values common to most hackers: free speech, freedom from coercion, meritocracy, and the power of the individual, all "reworked in the context of interaction with technical systems to create a diverse but related set of expressions concerning selfhood, property, privacy, labor, and creativity."[177]

The rhetoric adopted by open source advocates, invoking enlightened self-interest as a path to the greater good, suggests the liberal economic tradition of Adam Smith, whereas utilitarian openness is simply an ethical necessity to those who support the freedom of discourse embodied in software code. "The moral and semiotic load of free software is a commitment to prevent limiting the freedom of others," a fundamentally positive freedom for enabling, not preventing liberty.[178] Open source can also be interpreted as descending less from Smith's market theories and more from the pragmatism of John Dewey: "By cultivating pragmatic principles, rather than being inscribed by dogmatic principles, through the development of tools for problematic situations, individuals increase their potential to create knowledge that serves society."[179] The larger goal of Torvalds' "pragmatism as play" is experimentation through the equivalent of an enhanced form of peer review, although the argument that such an approach remains bound to the existence of democratic freedom still stands. By comparison, Stallman's idealism follows Immanuel Kant's categorical imperative to behave according to morals and ethics, regardless of material or practical

[176] Himanen, *The Hacker Ethic, and the Spirit of the Information Age*, 48.

[177] Coleman and Alex Golub, "Hacker Practice: Moral Genres and the Cultural Articulation of Liberalism," 267.

[178] Coleman, "The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast," 509.

[179] Maher, "Software Evangelism and the Rhetoric of Morality: Coding Justice in a Digital Democracy," 123.

effect.

Curiously, ideological denial is another commonly observed component of hacker ideology, one that may facilitate and enhance its broader cultural impact by inviting adoption and participation across the political spectrum.[180] Some, including cultural anthropologist Gabriella Coleman, trace the origins of this aversion to "an overt aesthetic dislike for politics and a culturally embodied experience of freedom that conceptually shuns politics."[181] Politics, unlike programming, is considered chaotic, sloppy, and ineffective, producing ambiguous results, a belief directly contradicted by Stallman's creation of the legal doctrine of Copyleft and Lawrence Lessig's Creative Commons movement, which deliberately mirrors free and open source software practices.

An Army of One

It quickly becomes evident that Torvalds represents a later generation of hacker, albeit a generation remarkably consistent in shared values and traits. Torvalds' generational distinctions, however, helped move the landscape of computer and information technology in a new direction. But they did not occur in a cultural vacuum: his "non-ideological" approach enjoyed an enthusiastic reception among his contemporaries in Silicon Valley and other communities around the San Francisco Bay Area, in large part because it mirrored and amplified their established convictions. A belief in individualistic self-sufficiency and non-hierarchical meritocracy, freedom from restraint, interference, and coercion, and the power of decentralized self-organizing markets all reflected a particular strand of Western liberalism embedded in North American cultural mythology, forming an unusual synthesis of Anglo-American liberal traditions and technocratic utopianism occasionally referred to as "cyber-libertarianism" or, more recently, "techno-

[180] Coleman, "The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast," 508.

[181] Ibid., 512.

libertarianism." This American libertarianism, filtered through the lens of technology, became "a more unified theory of Meritocracy."[182] To understand this ideology, strangely popular among those wishing to *avoid* ideology, some context is in order.

Among the hackers—widely considered outsiders, and often lacking in social skills—technical merit was always a favored value. From their shared perspective computer programming was a manifestation of rational positivism, and many, including Torvalds, argued in favor of their ethic based on its supposedly objective merits: increased efficiency, higher quality, and greater utility. Computers were expressions of mechanical rationality, conceptually intolerant of error, and the ability to express one's self creatively by means of a computer influenced practitioners into believing that hacking could be a true measure of individual merit.

Although the concept of placing the most deserving into positions of authority predates recorded history, formalized systems for gaining status and material reward based on merit are generally believed to have originated in ancient China, when followers of the philosopher Confucius advocated merit as a more suitable approach to governing an empire than hereditary rights and privileges. The practice took hold in the Western world after Britain became a global empire in the 18th century, subsequently recognizing a similar structural need. Enlightenment philosophers, including Voltaire and Mill, admired the concept, and in the United States a federal system of examinations was implemented in 1883 to end the practice of patronage-based corruption through appointments to powerful government positions.[183] As a political system, however, the concept of meritocracy was originally intended negatively. The first credited application of the term was provided by British sociologist Michael Dunlop Young, who portrayed meritocracy as a

[182] Rogers, "Libertarianism, Meritocracy and Equality."
[183] Kazin, Rebecca Edwards, and Adam Rothman, "The History of Meritocracy."

fantasy, unable to address the underlying realities of inequality of opportunity or the tendency to introduce measurement bias.[184]

Many hackers were first introduced to the concept of meritocracy in the novels of Ayn Rand, including *Atlas Shrugged* and *The Fountainhead*. Through her descriptions of brilliant and talented loners who refuse to compromise or suffer injustice, Rand constructed a theory of "objectivism," a self-described philosophical system based around the idea that objective reality exists, and pursuing self-interest was the highest expression of moral good.[185] Rand's "selfish supermen" were not simply dissatisfied with a society they viewed with detachment, they were heroic individuals, actively influencing and changing the world. Often taken very seriously, in spite of their fantastic, unrealistic qualities, the impact of Rand's works on impressionable minds became the source of a widely circulated witticism:

> There are two novels that can change a bookish fourteen-year old's life: *The Lord of the Rings* and *Atlas Shrugged*. One is a childish fantasy that often engenders a lifelong obsession with its unbelievable heroes, leading to an emotionally stunted, socially crippled adulthood, unable to deal with the real world. The other, of course, involves orcs.[186]

It should come as no surprise that when expressing moral outrage and intransigence over modifications to the MIT computer, hacker Bill Gosper reminded AI lab supervisor Edward Fredkin of "Rourke," the architect protagonist of Ayn Rand's *The Fountainhead*, who believed it preferable to destroy his work than allow diminishing through compromise."[187]

The primary vector for dispersing hacker values to Stanford appeared in the person of John McCarthy, recruited from MIT to create the Stanford Artificial Intelligence Laboratory. A widely

[184] Young, "Down With Meritocracy."
[185] Duignan, "Ayn Rand."
[186] Rogers, "Ephemera 2009 (7)."
[187] Levy, *Hackers: Heroes of the Computer Revolution*, 117.

respected figure, due both to his credentials and his earned merit as the creator of the LISP programming language, McCarthy ensured that hacker individualism and anti-authoritarianism would take root and prosper in Palo Alto.[188] The university boasted a well-established reputation for being aggressively entrepreneurial: founded by railroad tycoon Leland Stanford to compete with prestigious east coast research institutions, its engineering program was built by Frederick Terman, a major advocate of meritocracy following in the footsteps of his father, Lewis Terman, the man responsible for the Stanford Binet intelligence test.[189] Frederick launched the careers of Williams Hewlett and David Packard, the technology entrepreneurs and developers of semiconductor-based electronic devices that contributed to the region's unofficial title, "Silicon Valley."

As recently as five years ago the technology writer Evgeny Morozov referred to Silicon Valley as "a burgeoning enclave of Randian thought," including publisher O'Reilly among Rand's adherents. In a 2012 blog post describing his target audience, O'Reilly wrote: "There's a way in which the O'Reilly brand essence is ultimately a story about the hacker as hero, the kid who is playing with technology because he loves it, but one day falls into a situation where he or she is called on to go forth and change the world."[190] The connection between enthusiastic hacker and objectivist superman in his rhetoric is difficult to overlook.

Another Elephant

From a belief in technology as an indicator of merit, to an affinity for the capitalist heroics of Rand's protagonists, the dominant political ideology of Silicon Valley is frequently described as

[188] Cohen, *The Know-It-Alls: The Rise of Silicon Valley as a Political Powerhouse and Social Wrecking Ball*, 45.

[189] Ibid., 63.

[190] Morozov, "The Meme Hustler: Tim O'Reilly's Crazy Talk."

libertarian, a label, as we have observed, applied to and embraced by many hackers. The term's first recorded use traces back to 1789, expressing a philosophical opposition to determinism in favor of belief in free will.[191] Its political application evolved to define a tradition described as "classically liberal," originating in the principles embodied by the American and French revolutionary uprisings that held individual liberty and equality above all else.[192] Although progressive at the time, placed in opposition to an establishment based on hereditary rights, these principles were not the exclusive domain of the left. A key point of departure between left- and right-libertarianism centers on the nature of property rights and private ownership, and in the 1960s an American libertarian was more likely to be either a small-government collectivist or an anarcho-capitalist such as economist Murray Rothbard, espousing positions openly opposed by Ayn Rand.[193] To distinguish themselves and avoid confusion, adherents of private property rights, minimal government, and unregulated capitalism elected to us the term libertarian instead of liberal.[194]

Although objectivists opposed left-libertarians, regarding their attempts at regulating the accumulation of property as acts of aggression, contemporary right-libertarians are widely considered the intellectual descendants of Rand, her ideas informing "the libertarian commitment that capitalism and freedom were foundationally linked."[195] Libertarian economics became influential following the appointment of objectivist sympathizers to prominent positions within conservative administrations in the U.S. and U.K., among them former Chair of the Federal Reserve Alan Greenspan, a follower and personal friend of Rand.[196] Supporters of meritocracy, as a

[191] Belsham, *Essays, Philosophical, Historical, and Literary*, 11.
[192] Rothbard, "The Libertarian Heritage: The American Revolution and Classical Liberalism."
[193] Carlson, "Libertarianism," 1006.
[194] Russell, "Who Is a Libertarian?"
[195] Carlson, "Libertarianism," 1006.
[196] Boaz, "Libertarianism."

political theory, see a key distinction between their belief systems: in their analysis, merit, just like objectivism's selfishness, is a fundamentally moral concept. For the deserving to be rewarded, society must be fundamentally equal and fair. Libertarianism allows for unequal conditions and distorted outcomes, including inherited wealth and entrenched economic monopoly, and such inequalities lead inevitably to corresponding inequalities in political power. Individual liberty is held as a fundamental right, not a moral value, and fairness has no bearing on freedom.[197] The arguments against libertarianism are well understood, with perhaps the most prominent being its "atomistic," ahistorical nature. Individuals are formed in large part through the interaction of families and communities, who provide benefits in addition to obligations. For libertarians the central philosophical issue is not individuality versus community, but consent versus coercion.[198] Libertarianism, like meritocracy, is utopian. Unlike socialism, no political system completely embodies its principles—although some have argued that Somalia, lacking a functioning central government for over twenty years and still considered precarious, could be cited as an example.[199]

As far back as 1996 the cultural and social consequences of the counterculture in combination with unregulated capitalism and technology were observed critically. English media theorists Richard Barbrook and Andy Cameron dubbed it "the California Ideology," a phenomenon that "emerged from a bizarre fusion of the cultural-bohemianism of San Francisco with the hi-tech industries of Silicon Valley." It "simultaneously reflects the disciplines of market economics and the freedoms of hippie artisanship […] through a nearly universal belief in technological determinism."[200] The contemporary libertarian values of individual freedom and unlimited

[197] Mulligan, "What's Wrong with Libertarianism: A Meritocratic Diagnosis," 77.
[198] Boaz, "Libertarianism."
[199] Although such arguments are usually intended as satire (see this video, https://www.youtube.com/watch?v=8Ul-Efi1Xys, for an example), at least one economist, Peter Leeson, suggested that Somalia's lack of a central government provided an economic advantage, in "Better Off Stateless: Somalia Before and After Government Collapse."
[200] Barbrook and Andy Cameron, "The Californian Ideology."

economic opportunity were wedded to a profound faith in the emancipatory potential of the new information technologies. Coincidentally, Barbrook and Cameron label the hacker hero of their cultural narrative (popularized in the cyberpunk novels of the day) the personification of what they refer to as "autistic libertarianism."

In 2017, technology journalist Noam Cohen came to describe the California Ideology as the "Silicon Valley belief system," in which adherents advocate for "a highly individualistic society led by the smartest people, who deliver wonderful gadgets and platforms for obtaining goods, services, and information efficiently, freeing us to compete in the marketplace for our daily bread."[201] Cohen characterizes it as a fusion of the hacker ethic with the Stanford belief that ideas should be tested and disseminated through the marketplace. The Electronic Freedom Foundation, cofounded by Gilmore with fellow libertarians John Perry Barlow and Mitch Kapor, represents the classic hacker anti-authoritarian belief that no government should have the right to impose limits on the freedoms inherent in cyberspace. A decidedly different, objectivist ethic voiced by Peter Thiel is that the "best and brightest" should lead society, free from ignorant authorities who would impose coercive regulations.[202] The hacker confidence in technology's ability to improve society becomes combined with the belief that an unregulated market, composed of participants acting naturally according to their own self-interest, is the path to a moral workplace and a just society. Responding to a 2015 New York Times article describing both Amazon's corporate offices and warehouse fulfillment centers as joyless dystopian environments, company founder Jeff Bezos expressed complete disbelief. His Silicon Valley belief system required an unalloyed faith in the free market, making unhappiness logically impossible.[203]

---

[201] Cohen, *The Know-It-Alls: The Rise of Silicon Valley as a Political Powerhouse and Social Wrecking Ball*, 9.

[202] Ibid., 156.

[203] Ibid., 111.

The arrival of the open source movement altered the discourse from collectivist, positive freedoms to individualist, negative freedoms, a change that perfectly suited Silicon Valley venture capitalists and entrepreneurs. Raymond was successful, with commercial adoption of Linux skyrocketing alongside the appearance of numerous Linux-oriented businesses, although its greatest success, powering mobile devices, was yet to come. Torvalds became a celebrity among the new generation of hackers, as well as becoming a commodity in demand. He relocated to Silicon Valley in 1995, and by 2000 was employed at a foundation near Portland created specifically to facilitate his work, so that the resulting benefits could be shared among competing corporate interests including Oracle, Google, IBM, and, yes, Microsoft.[204] Among the participants of the Linux kernel distributed development project, Torvalds was crowned "Benevolent Dictator for Life," intended as a term of endearment while clearly indicating his status as the ultimate arbiter of all technical decisions.[205]

Not Entirely Fun

The Venture Capitalist class of Silicon Valley continued to "move fast and break things," as Jonathan Taplin famously warned, to predictable economic results. Google, Amazon, and Facebook creatively disrupted the market using the tech virtues of scalability and innovation, evolving into monopolies while remaining largely detached from the rampant economic inequality and environmental crises surrounding them. Today these companies are among the wealthiest, most powerful organizations on the planet, and all three built their success on a technical foundation of Linux and GNU. Torvalds' hobby lead to the fulfillment of the GNU project, significantly contributed toward ending Microsoft's software hegemony, and created the only rival to Apple in

[204] The Linux Foundation, "Our Corporate Members."
[205] Torvalds and David Diamond, *Just for Fun: The Story of an Accidental Revolutionary*, 168.

telecommunications, the Android system, compelling Apple to adopt a competing version Unix in order to remain viable as a hardware and services company.

In this environment of economic success, regarded by techno-libertarians as a positive development vindicating their beliefs, hackers continued the practice of denying political intent. With his ideals of sharing and community, critics enjoyed portraying Stallman as essentially communist, but Free Software was never anti-capitalist. Open source software, on the other hand, embodied libertarian values, offering no idealism beyond individual freedom. As apolitical as he was nonideological, Torvalds never openly adopted a philosophical alignment beyond his hierarchy of needs—when approached on the topic of politics, he argued in favor of a conciliatory approach he believed common to European parliamentary systems.[206] Stallman makes a concerted effort to keep his politics separate from the Free Software movement, but feels no ambivalence when discussing his personal political orientations:

> I'm not an anarchist—we need a state so we can have a welfare state. I'm no "libertarian" in the usual American sense, and I call them rather "antisocialists" because their main goal is a laissez-faire, laissez-mourir economy. People like me are the true libertarians.[207]

In spite of, or perhaps because of these differences, people are far more likely to know Torvalds' name, although that could be equally related to its similarity with his project's name. Linus Torvalds, accidentally or otherwise, changed the world. His genius may have arrived unintentionally, but its results are undeniable.

If the narrative of Torvalds and Linux ended here, the consensus of his legacy would be mostly positive. Not every historical figure operates from a place of selflessness and principle, and good intentions can have as many beneficial as harmful results. But the public appearance of a

[206] Ibid., 22.
[207] Lucas, "Richard Stallman: Talking to the Mailman," 83.

"fun" project concealed another hacker characteristic. The public "flaming" of MINIX creator Tanenbaum was not exceptional for Torvalds: technical arguments taking place over the Linux Kernel Mailing List, the primary channel for coordinating developer activities, were both frequent and frequently high-strung, incorporating streams of personal and abusive language. As a public, "open" project, these interactions are both visible and subject to analysis. In a 2016 study, researchers applied machine-learning algorithms to the mailing list's messages to identify the dominant characteristics of participant discourse, and the results indicated that Torvalds was significantly more likely to use commonly recognized English-language expletives and profanity than his fellow contributors.[208] When approached by project participants about this behavior, Torvalds refused to consider moderating his language or communication style while acknowledging his tendency to cause offense: "Yes. And I do it partly (mostly) because it's who I am, and partly because I honestly despise being subtle or 'nice.'" Claiming that foul language and hostility make for an effective management style, he opposed the idea of imposing any form of oversight, stating in response to such calls, "venting of frustrations and anger is actually necessary, and trying to come up with some 'code of conduct' that says that people should be 'respectful' and 'polite' is just so much crap and bullshit."[209]

Such language does not give the impression of fun Torvalds described as his primary motivation, but it is possible he enjoys confrontation. Equally possible is a previously mentioned component of hacker culture, the fondness for arguing over The Right Thing. In his autobiography Torvalds' mother describes heated arguments with his sister at the kitchen table, easily attributable to childhood irascibility. For some, however, this behavior might be reminiscent of children with AS, unable to process disagreement or countenance dissent, let alone rein in violent impulses or

[208] Schneider, "Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List," 2.
[209] Brodkin, "Linus Torvalds Defends His Right to Shame Linux Kernel Developers."

process their strong feelings constructively. Other interpretations suggest the lingering unaddressed consequences of childhood abuse, referencing a reply Torvalds publicly made to the suggestion he consider psychological counseling:

> Others have been verbally abusive to me, and I've dealt with it on my own. But the most important thing is that when I explode on the mailing list and rant, I immediately feel better. That's the solution and it works for me.[210]

Regardless, for many participants the dictatorship became toxic, not fun. Even so, it came as a surprise to many that on September 16, 2018, following a routine update to the mailing list mentioning an upcoming release, Torvalds wrote:

> This week people in our community confronted me about my lifetime of not understanding emotions. My flippant attacks in emails have been both unprofessional and uncalled for. Especially at times when I made it personal. In my quest for a better patch, this made sense to me. I know now this was not OK and I am truly sorry. […] I am going to take time off and get some assistance on how to understand people's emotions and respond appropriately.[211]

This message appeared shortly after Noam Cohen conducted a series of interviews with Torvalds and members of the Linux Foundation. According to Cohen, "Torvalds's decision to step aside came after *The New Yorker* asked him a series of questions about his conduct for a story on complaints about his abusive behavior discouraging women from working as Linux-kernel programmers."[212]

One immediate consequence of this decision was the creation of a project Code of Conduct. The Linux kernel remains actively developed under the leadership of other project members. After 27 years, the dictator was deposed, but the operating system remains free, in every sense of the word.

---

[210] Kuhn, "On Psychological Help."
[211] Torvalds, "Linux 4.19-Rc4 Released, an Apology, and a Maintainership Note."
[212] Cohen, "After Years of Abusive E-Mails, the Creator of Linux Steps Aside."

Conclusion

> *It's one of the ironies of the history of free software that its moment of greatest fame was associated with this term, open source, which you reject.*
>
> Because it's not the name of a philosophy—it refers to the software, but not to the users. You'll find lots of cautious, timid organizations that do things that are useful, but they don't dare say: users deserve freedom. Like *Creative Commons*, which does useful, practical work—namely, preparing licences [sic] that respect the freedom to share. But Creative Commons doesn't say that users are *entitled* to the freedom to share; it doesn't say that it's wrong to deny people the freedom to share. It doesn't actively uphold that principle. Of course, it's much easier to be a supporter of open source, because it doesn't commit you to anything. You could spend ten minutes a week doing things that help advance open source, or just say you're a supporter—and you're not a hypocrite, because you can't violate your principles if you haven't stated any.
>
> —Richard Stallman, interviewed by Matt Lucas in *New Left Review*[213]

Although not intended to produce a definite conclusion or diagnosis, my research suggests two patterns. First, there appears to be a strong connection between the traits associated with neurodivergence and talents in certain disciplines, specifically engineering, computer programming, and the applied sciences. In instances of high natural intelligence, those falling within the autism spectrum appear capable of making great contributions to their fields of endeavor. This is a perspective shared by many among both the diagnosed and those familiar with the condition. Some go so far as to suggest a genetic, inherited component, and conjecture that many of our cultural and social advances are directly attributable to these traits, implying some form of evolutionary determinism.[214]

Second, on the ideological side of the equation, the culture of Silicon Valley, and of similar technology centers across the United States, exhibits a political alignment best described as "right-libertarian." Although not all members of the tech community share the same ideals or affiliations,

[213] Lucas, "Richard Stallman: Talking to the Mailman," 79.
[214] Siegert and Ward, "Clinical Psychology and Evolutionary Psychology: Toward a Dialogue," 248.

those that have successfully established the dominant industries—along with the educational institutions supporting them—are often proudly and outspokenly libertarian, believing in extremely limited government and freedom from interference. Their talent for systems and belief in scientific positivism lead many to assume that the marketplace and technology will identify the "best and brightest," equally capable of solving problems in other domains of the public sphere while obviating the need for government activity beyond enforcing contracts and protecting individual (and property) rights.

Taken separately, these concepts have been explored in both scholarly and informal contexts. But the job of the scholar is to raise questions that aren't being asked; as a technology professional, having observed these circumstances first-hand, I find myself asking, "Is there a connection between these phenomena?" Discussing the essentialism of gender, Sigmund Freud once famously wrote, "Anatomy is destiny." Is neurobiology ideology?

The social alienation experienced by many outcast nerds, along with a deficit in empathy born of an inability to perceive the perspective of other, less precise and systematic individuals, could be a contributing factor. The ability to excel in a specific domain, particularly one that affords personal expression and creativity, can give rise to a sense of measurable superiority—especially when the results prove fantastically lucrative. And a tendency toward inflexible absolutism can lead many to extreme conclusions and extremist ideologies. The appearance of angry nerds using the Internet to support authoritarian "alt-right" politics, for example, enacting real-world harassment of opponents in the process, was recently popularized using the term "weaponized autism."[215] This is all unavoidably conjecture: the simple fact is that we cannot definitely measure, or even establish a firm metric for, what is or isn't (and more specifically, *who*

[215] Klee, "The Rise of 'Weaponized Autism.'"

is or isn't) sufficiently different that an individual's talents and beliefs become predictable. Which is ultimately for the best, as it can lead to the justly discredited tenets of eugenics—tenets Hans Asperger practiced when evaluating the "fitness" of children for life in Nazi society. To believe traits and ideologies are biologically determined is to ignore the great complexity and variation within humanity, let alone the fact that, among professionals, the very definition of these potential disorders is a source of debate and disagreement.

Nonetheless, psychologists, neurobiologists, and political scientists are researching the connections between structures of the brain and political orientation, between "conservatism" and "liberalism," broadly-defined, and their results do suggest a relationship. Although not specific to the autism spectrum, some researchers, including psychologist John Jost, believe such neurological differences exist and may be structural in nature.[216] Others, such as social psychologist Jonathan Haidt, go so far as to suggest that libertarian morality is directly connected to measurable psychological traits, and the "libertarian narrative" resonates strongly with those possessing attributes strongly correlated with AS.[217]

If the observations compiled in this thesis are reasonable then, from a medical perspective, treatments or interventions for childhood autism could directly affect future developments in specific academic and commercial fields. From an economic and political perspective, those who argue for a greater sense of social participation might consider how such arguments reach people inherently limited in their ability to empathize or connect with others, or who hold independence from constraint and obligation among their highest values—an issue far beyond neurobiology. With the resurgence of labor unions and democratic socialism, some companies are finding

[216] Jost and Amodio, "Political Ideology as Motivated Social Cognition: Behavioral and Neuroscientific Evidence."

[217] Iyer et al., "Understanding Libertarian Morality: The Psychological Dispositions of Self-Identified Libertarians."

themselves compelled to engage with arguments for social responsibility through the bottom-up efforts of activists and organizers, including the recently-formed Tech Workers Coalition and those responsible for a walkout staged at Google offices around the world, protesting the handling of sexual misconduct claims.[218]

From a larger cultural perspective, how do we take responsibility for our near-constant consumption of technologies that atomize and isolate us, encouraging our selfish, anti-social tendencies while representing these mediated interactions as new, idealized forms of sociability instead?

> According to a new Nielsen report, adults in the United States devote about nine hours a day to looking at a screen. Mark Zuckerberg, Larry Page, Peter Thiel, and Jeff Bezos have gotten bewilderingly rich off of this so-called attention economy because they are controlling almost four hours a day of that screen time. […] The social media tools that are sometimes touted as instruments of freedom against dictatorships turned out to be weapons capable of harming democracy. As Tim Berners-Lee recently told *Vanity Fair*, the internet 'ended up producing—with no deliberate action of the people who designed the platform—a large-scale emergent phenomenon which is anti-human.'[219]

The designers may not have deliberately set out to create such a platform, but it is increasingly apparent that the Internet, and the alternatives it provides to traditional means of engaging attention, reflects their decentralized hacker values. Contemplating Peter Thiel's praise for the hypothetical "Aspie entrepreneur," Carnegie Mellon University Vice Provost Justine Cassell observed:

> It seems odd to say that the best innovation is going to come from somebody who doesn't study people. In fact who isn't able to spend casual time with them. If really what he's saying is Silicon Valley start-ups are places where the winners are the ones who don't have any social skills, to me that's a sad statement about who is inventing the technology of tomorrow.[220]

---

[218] O'Brien, "How Silicon Valley Is Being Reshaped by Trade Unions"; Lee, "Google Staff Walk out over Women's Treatment."

[219] Taplin, "Rebirth of a Nation: Can States' Rights Save Us From a Second Civil War?," 34.

[220] McFarland, "Why Shades of Asperger's Syndrome Are the Secret to Building a Great Tech Company."

Anecdotal evidence is not data, and opinions vary widely, but a general consensus borne out by my research is that Asperger syndrome, computer science, and libertarianism are each notably compatible with the other. And in places where they concentrate, such as Silicon Valley, the cultural influence of these affinities is not entirely regarded as positive. Observing a phenomenon she described as "cyberselfishness" in 1996, journalist Paulina Borsook wrote:

> But what will result if the people who want to shape public policy know nothing about history or political science or, most importantly, how to interact with other humans? Programmers, and those who know how to make money off them, mostly find it easier to interact in e-mail than IRL (in real life), and are often not good at picking up the cues, commonplaces, and patterns of being that civilians use to communicate, connect, and operate in groups. The convergence between libertarianism and high-tech has created the true revenge of the nerds: Those whose greatest strengths have not been the comprehension of social systems, appreciation of the humanities, or acquaintance with history, politics, and economics have started shaping public policy. Armed with new money and new celebrity—juice—they can wreak vengeance on those by whom they have felt diminished.[221]

Whatever the relationship between neurobiology and ideology, the impact of our genius programmers, our crusader and dictator, extends far beyond hackers and hobbyists. Their work quietly altered the global landscape, socially, economically, and politically, with effects we are still attempting to measure and understand. In our world of algorithms identifying military targets and conducting drone strikes, and of enormous real-time data sets analyzed to identify potential terrorists using multiple networks of statistical regression, the instrumentalist perception of technology as a neutral, unbiased expression of disembodied objectivity is increasingly rejected.[222]

Looking back, the ideological legacy of hacker culture reveals many flaws, but perhaps the underlying ethic of community can show a way forward. Perhaps society is prepared to consider the ethical values behind our technology choices, and our collective responsibility for participation and agency as software users. The Internet embodies the hacker ethic of sharing, but also facilitates

[221] Borsook, "Cyberselfishness."
[222] Wilcox, "Embodying Algorithmic War: Gender, Race, and the Posthuman in Drone Warfare."

exploitation. Smartphones expand access to information in ways previously regarded as science fiction, but do little to disprove an old expression from the world of computer programmers, "garbage in, garbage out."[223] Our image of the stereotypical computer nerd, apolitical yet creating tools with political consequences, is precisely that: an unsustainable stereotype that demands cultural engagement and dialogue, regardless of validity. We're tasked with weighing freedom from interference against freedom of choice, but the success of the Free Software project also offers technology as a source of mutual liberty. Stallman understood that taking pleasure in an activity can be its own reward, but his project offers both the hacker and the user the freedom to discover a purpose beyond entertainment:

> Linus is out there to have fun. He wrote his kernel not for money but because programming is fun. He's not an idealist, but he's not an intense capitalist. He's an engineer. He has fun. I am too but I'm also a political idealist, which seems to me a higher calling.[224]

[223] Howe, "GIGO."
[224] Gross, "Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-Certified Genius."

Bibliography

Ancora, Lorenzo. “License: BSD-4-Clause.” Free Software Foundation, January 22, 2019. https://directory.fsf.org/wiki/License:BSD-4-Clause.

Anonymous. “BSD License Definition.” The Linux Information Project, April 22, 2005. http://www.linfo.org/bsdlicense.html.

Appignanesi, Lisa. “‘Dr. Death,’ a Review of Edith Scheffer’s Asperger’s Children: The Origins of Autism in Nazi Vienna.” *The New York Review of Books*, July 19, 2018.

Attwood, Tony. *The Complete Guide to Asperger’s Syndrome*. London: Jessica Kingsley Publishers, 2006.

Atwood, Jeff. “Bill Gates and DONKEY.BAS.” *Coding Horror: Programming and Human Factors* (blog), May 14, 2007. https://blog.codinghorror.com/bill-gates-and-donkey-bas/.

———. “Software Developers and Asperger’s Syndrome.” *Coding Horror: Programming and Human Factors* (blog), January 15, 2006. https://blog.codinghorror.com/software-developers-and-aspergers-syndrome/.

Barbrook, Richard, and Andy Cameron. “The Californian Ideology.” *Science as Culture* 6, no. 1 (1996): 44–72. http://www.imaginaryfutures.net/2007/04/17/the-californian-ideology-2/.

Baron-Cohen, Simon. “Is Asperger Syndrome Necessarily Viewed as a Disability?” *Focus on Autism and Other Developmental Disabilites* 17, no. 3 (2002): 186–91. https://doi.org/10.1177/10883576020170030801.

———. “The Essential Difference: Men, Women and the Extreme Male Brain.” In *Oxford Companion to the Mind*, edited by Richard Gregory, 2nd ed. Oxford: Oxford University Press, 2004.

Belsham, William. *Essays, Philosophical, Historical, and Literary*. Edited by Charles Dilly, 1789. https://books.google.com/books?id=Z6Y0AAAAMAAJ.

Benkler, Yochai. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven, CT: Yale University Press, 2006. https://cyber.harvard.edu/wealth_of_networks/.

Berlin, Isaiah. “Two Concepts of Liberty.” In *Four Essays on Liberty*, 121–54. Oxford: Clarendon Press, 1969.

Blankenship, Loyd. “The Conscience of a Hacker.” *Phrack Magazine*, January 8, 1986. http://www.phrack.org/issues/7/3.html.

Boaz, David. “Libertarianism.” In *Encyclopædia Britannica*. Chicago, IL: Britannica Group, January 30, 2009. https://www.britannica.com/topic/libertarianism-politics.

Borgmann, Albert. *Technology and the Character of Contemporary Life: A Philosophical Inquiry*. Chicago, IL: The University of Chicago Press, 1984.

Borsook, Paulina. “Cyberselfishness.” *Mother Jones*, July 1996. https://web.archive.org/web/20071018044511/http://www.motherjones.com:80/news/feature/1996/07/borsook.html.

Brodkin, Jon. “Linus Torvalds Defends His Right to Shame Linux Kernel Developers.” *Ars Technica*, July 16, 2013. https://arstechnica.com/information-technology/2013/07/linus-torvalds-defends-his-right-to-shame-linux-kernel-developers/.

Butterfield, Andrew, and Gerard E. Ngondi. “Computer Science.” In *A Dictionary of Computer Science*. Oxford: Oxford University Press, n.d. http://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-963.

Carlson, Jennifer D. “Libertarianism.” In *The Social History of Crime and Punishment in America: An Encyclopedia*, edited by Wilbur R. Miller. London: Sage Publications, Inc., 2012.

Chiou, Stefanie, Craig Music, Kara Sprague, and Rebekah Wahba. “A Marriage of Convenience: The Founding of the MIT Artificial Intelligence Laboratory.” MIT Structure of Engineering Revolutions Project, December 5, 2001. http://web.mit.edu/6.933/www/Fall2001/AILab.pdf.

Chomsky, Noam. “Three Models for the Description of Language.” *IRE Transactions in Information Theory* 2 (1956): 113–24.

Church, Alonzo. “An Unsolvable Problem of Elementary Number Theory.” *American Journal of Mathematics* 58, no. 2 (n.d.): 345–63.

Cohen, Noam. “After Years of Abusive E-Mails, the Creator of Linux Steps Aside.” *The New Yorker*, September 19, 2018. https://www.newyorker.com/science/elements/after-years-of-abusive-e-mails-the-creator-of-linux-steps-aside.

———. *The Know-It-Alls: The Rise of Silicon Valley as a Political Powerhouse and Social Wrecking Ball*. New York: The New Press, 2017.

Coleman, E. Gabriella. “Code Is Speech: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers.” *Cultural Anthropology* 24, no. 3 (August 2009): 420–54. https://www.jstor.org/stable/25619791.

———. “The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast.” *Anthropological Quarterly* 77, no. 3 (2004): 507–19. https://www.jstor.org/stable/3318232.

Coleman, E. Gabriella, and Alex Golub. “Hacker Practice: Moral Genres and the Cultural Articulation of Liberalism.” *Anthropological Theory* 8, no. 3 (September 1, 2008): 255–77. https://doi.org/10.1177/1463499608093814.

Computer History Museum. “IBM System/360.” Computer History Museum, n.d. https://www.computerhistory.org/revolution/mainframe-computers/7/161.

Copeland, B. Jack. “A Brief History of Computing: ENIAC and EDVAC.” AlanTuring.net, June 2000. http://www.alanturing.net/ turing_archive/pages/Reference%20Articles/BriefHistofComp.html.

———, ed. *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life: Plus the Secrets of Enigma*. New York: Oxford University Press, 2004.

Davis, Martin D. *Engines of Logic: Mathematicians and the Origin of the Computer*. New York: W. W. Norton & Company, 2001.

Duignan, Brian. “Ayn Rand.” In *Encyclopædia Britannica*. Chicago, IL: Britannica Group, January 14, 2019. https://www.britannica.com/biography/Ayn-Rand.

Eisenhower, Dwight D. “Farewell Address to the Nation,” January 17, 1691. http://avalon.law.yale.edu/20th_century/eisenhower001.asp.

Floridi, Luciano. “Information.” In *The Blackwell Guide to the Philosophy of Computing and Information*, edited by Luciano Floridi, 379. Blackwell Philosophy Guides. New York: Wiley-Blackwell, 2004.

Foucault, Michel. “The Birth of the Prison.” In *Discipline and Punish*. New York: Pantheon Books, 1977.

Fritz, W. Barkley. “The Women of ENIAC.” *IEEE Annals of the History of Computing* 18, no. 3 (1996).

Galilei, Galileo. *Il Saggiatore*. Rome, IT: Accademia dei Lincei, 1623.

Gates III, William Henry. “An Open Letter to Hobbyists,” February 3, 1976.

http://www.blinkenlights.com/classiccmp/gateswhine.html.

GBD 2015 Disease and Injury Incidence and Prevalence Collaborators. "Global, Regional, and National Incidence, Prevalence, and Years Lived with Disability for 310 Diseases and Injuries, 1990-2015: A Systematic Analysis for the Global Burden of Disease Study 2015." *The Lancet* 388, no. 10053 (October 8, 2016). https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(16)31678-6/fulltext#%20.

Graham, Paul. "Revenge of the Nerds," May 2002. http://www.paulgraham.com/icad.html.

Grandin, Temple. "Choosing the Right Job for People with Autism or Asperger's Syndrome." Indiana University Bloomington, November 1999. https://www.iidc.indiana.edu/pages/Choosing-the-Right-Job-for-People-with-Autism-or-Aspergers-Syndrome.

Gross, Michael. "Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-Certified Genius." In *The More Things Change: Why the Baby Boom Won't Fade Away*. New York: Cliff Street Books, 1999. http://mgross.com/writing/books/my-generation/bonus-chapters/richard-stallman-high-school-misfit-symbol-of-free-software-macarthur-certified-genius/.

Hazard, John. "Does Asperger's Make You a Better Software Coder?" *ZDNet Between the Lines* (blog), February 24, 2011. https://www.zdnet.com/article/does-aspergers-make-you-a-better-software-coder/.

Heidegger, Martin. *The Question Concerning Technology, and Other Essays*. Translated by Williams Lovitt. New York: Harper & Row, Publishers, Inc., 1977.

Hilbert, David, and Wilhelm Ackermann. *Grundzüge Der Theoretischen Logik*. Berlin, DE: Springer-Verlag, 1928.

Himanen, Pekka. *The Hacker Ethic, and the Spirit of the Information Age*. New York: Random House, 2001.

Howe, Denis. "GIGO." In *Free On-Line Dictionary of Computing*. London, October 3, 2004. http://foldoc.org/GIGO.

Hutson, Matthew. "AI Researchers Allege That Machine Learning Is Alchemy." *Science*, May 3, 2018. http://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy.

Ihde, Don. "Technics and Praxis." In *Boston Studies in the Philosophy of Science*, edited by Robert S. Cohen and Marx W. Wartofsky, 24:59–74. Boston, MA: D. Reidel Publishing Company, 1979.

Inahkto, Andy. "Microsoft CEO Takes Launch Break with the Sun-Times." *Chicago Sun Times*. June 1, 2001.

Iyer, Ravi, Spassena Koleva, Jesse Graham, Peter Ditto, and Jonathan Haidt. "Understanding Libertarian Morality: The Psychological Dispositions of Self-Identified Libertarians." *PLOS ONE* 7, no. 8 (August 21, 2012): 1–23. https://doi.org/10.1371/journal.pone.0042366.

Jaarsma, Pier, and Stellan Welin. "Autism as a Natural Human Variation: Reflections on the Claims of the Neurodiversity Movement." *Health Care Analysis* 20, no. 1 (February 11, 2011): 20–30. https://doi.org/10.1007/s10728-011-0169-9.

Jefferson, Thomas. "No Patents on Ideas (Letter to Isaac McPherson)," August 13, 1813.

Jost, John T., and David M. Amodio. "Political Ideology as Motivated Social Cognition: Behavioral and Neuroscientific Evidence." *Motivation and Emotion* 36, no. 1 (November 13, 2011): 55–64. https://doi.org/10.1007/s11031-011-9260-7.

Kapp, Steven K., Kristen Gillespie-Lynch, Lauren E. Sherman, and Ted Hutman. "Deficit, Difference, or Both? Autism and Neurodiversity." *Developmental Psychology* 49, no. 1 (January 2013): 59–71.

https://doi.org/10.1037/a0028353.

Kawasaki, Guy. *The Macintosh Way: The Art of Guerilla Management*. New York: Scott Foresman and Company, 1990.

Kazin, Michael, Rebecca Edwards, and Adam Rothman, eds. "The History of Meritocracy." In *Princeton Encyclopedia on American History*, 142, 2010.

Khazan, Olga. "Autism's Hidden Gifts." *The Atlantic*, September 23, 2015. https://www.theatlantic.com/health/archive/2015/09/autism-hidden-advantages/406180/.

Klee, Miles. "The Rise of 'Weaponized Autism.'" *MEL Magazine* (blog), October 2, 2017. https://melmagazine.com/the-rise-of-weaponized-autism-e38472124c6d.

Knuth, Donald E. "George Forsythe and the Development of Computer Science." *Communications of the ACM* 15, no. 8 (July 1972): 721–26.

Kort, Alicia. "The Story of Steve Jobs, Xerox, and Who Really Invented the Personal Computer." *Newsweek*, March 19, 2016. https://www.newsweek.com/silicon-valley-apple-steve-jobs-xerox-437972.

Kuhn, Bradley. "On Psychological Help." *LWN.Net*, September 20, 2018. https://lwn.net/Articles/765859/.

Laqueur, Walter Z. *The New Terrorism: Fanaticism and the Arms of Mass Destruction*. New York: Oxford University Press, 1999.

Lee, Dave. "Google Staff Walk out over Women's Treatment." *BBC News*. November 1, 2018, sec. Technology. https://www.bbc.com/news/technology-46054202.

Leeson, Peter T. "Better Off Stateless: Somalia Before and After Government Collapse." *Journal of Comparative Economics* 35, no. 4 (2007): 689–710.

Leland, John. "For Autistic Boys, the Subway Is Actually Soothing." *The New York Times*. November 8, 2018. https://www.nytimes.com/2018/11/08/nyregion/autistic-or-spectrum-boys-and-the-subway.html.

Levy, Steven. *Hackers: Heroes of the Computer Revolution*. Sebastopol, CA: O'Reilly Media, 2010.

Licklider, J. C. R. "The Computer as Communications Device." *Science and Technology*, April 1968.

Light, Jennifer S. "When Computers Were Women." *Technology and Culture* 40, no. 3 (1999).

Lloyd, William Forster. *Two Lectures on the Checks to Population*. Oxford: Oxford University Press, 1833.

Lohr, Steve. "Group Formed to Support Linux as a Rival to Windows." *The New York Times*. January 22, 2007. https://www.nytimes.com/2007/01/22/technology/22linux.html.

Lucas, Rob. "Richard Stallman: Talking to the Mailman." *New Left Review*, September 2018. https://newleftreview.org/II/113/richard-stallman-talking-to-the-mailman.

MacArthur Fellows Program. "Richard M. Stallman." MacArthur Foundation, January 1, 2005. https://www.macfound.org/fellows/412/.

Maher, Jennifer Helene. "Software Evangelism and the Rhetoric of Morality: Coding Justice in a Digital Democracy." Dissertation, Iowa State University, 2006. https://lib.dr.iastate.edu/rtd/13941.

Marx, Karl. "Theses on Feuerbach." In *Ludwig Feuerbach and the End of Classical German Philosophy*. Stuttgart, DE: J. H. W. Dietz, 1888.

Marx, Karl, and Friedrich Engels. *Capital, A Critique of Political Economy*. Vol. 1. 3 vols. N: International Publishers Co., Inc., 1967.

Mayor, Tracy. "Asperger's and IT: Dark Secret or Open Secret?" *Computerworld*, April 2, 2008.

https://www.computerworld.com/article/2536193/it-management-asperger-s-and-it-dark-secret-or-open-secret.html.

McCarthy, John. “Recursive Functions of Symbolic Expressions and Their Computation by Machine.” *Communications of the ACM* 3, no. 4 (April 1960): 184–95.

McClellan, John L. The Copyright Act of 1976 (1978). https://www.copyright.gov/title17/92appa.pdf.

McFarland, Matt. “Why Shades of Asperger’s Syndrome Are the Secret to Building a Great Tech Company.” *The Washington Post*. April 3, 2015. https://www.washingtonpost.com/news/innovations/wp/2015/04/03/why-shades-of-aspergers-syndrome-are-the-secret-to-building-a-great-tech-company/.

McPartland, James, and Ami Klin. “Asperger’s Syndrome.” *Adolescent Medicine Clinics* 17, no. 3 (October 2006): 771–88. https://doi.org/10.1016/j.admecli.2006.06.010.

Microsoft. “Microsoft Global Diversity and Inclusion Autism Hiring Program.” Inside Microsoft, n.d. https://www.microsoft.com/en-us/diversity/inside-microsoft/cross-disability/hiring.aspx.

Mitcham, Carl, and Robert Mackey, eds. *Philosophy and Technology*. New York: The Free Press, 1972.

Morozov, Evgeny. “The Meme Hustler: Tim O’Reilly’s Crazy Talk.” *The Baffler*, April 2013. https://thebaffler.com/salvos/the-meme-hustler.

Moye, William T. “ENIAC: The Army-Sponsored Revolution.” US Army Research Laboratory, January 196AD. https://ftp.arl.mil/~mike/comphist/eniac-story.html.

Mulligan, Thomas. “What’s Wrong with Libertarianism: A Meritocratic Diagnosis.” In *The Routledge Handbook of Libertarianism*, edited by J. Brennan, D. Schmidtz, and B. van der Vossen, 77–91. New York: Routledge, 2018.

Muzikar, Debra. “An Interview with Steve Silberman Author of Neurotribes.” The Art of Autism, October 5, 2016. https://the-art-of-autism.com/an-interview-with-steve-silberman-author-of-neurotribes/.

O’Brien, Hettie. “How Silicon Valley Is Being Reshaped by Trade Unions.” *New Statesman*, March 13, 2019. https://www.newstatesman.com/world/2019/03/how-silicon-valley-being-reshaped-trade-unions.

O’Connell, Henry, and Michael Fitzgerald. “Did Alan Turing Have Asperger’s Syndrome?” *Irish Journal of Psychological Medicine* 20, no. 1 (2003): 28–31. http://crackingtheenigma.blogspot.com/2012/06/did-alan-turing-have-asperger-syndrome.html.

Oxford English Dictionary. “Nerd, n.” In *OED Online*. Oxford University Press, March 2019. http://www.oed.com/view/Entry/126165.

Parry, Richard. “Episteme and Techne.” In *The Stanford Encyclopedia of Philosophy*, June 22, 2014. https://plato.stanford.edu/entries/episteme-techne/.

Pinheiro, John. “AT&T Divestiture & the Telecommunications Market.” *Berkeley Technology Law Journal* 2, no. 2 (September 1987).

PRS for Music. *Gary Numan: “My Asperger’s Is an Absolute Advantage.”* London, 2017. https://www.youtube.com/watch?v=aTKHZeP-IZQ.

Raymond, Eric Steven. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O’Reilly Media, 1999.

Riordan, Michael, and Lillian Hoddeson. *Crystal Fire: The Birth of the Information Age*. New York: W. W. Norton & Company, 1997.

Ritchie, Dennis M. “The Development of the C Language.” *ACM SIGPLAN Notices* 28, no. 3 (March 1993):

201–8. http://www.bell-labs.com/usr/dmr/www/chist.html.

———. “The Evolution of the Unix Time-Sharing System.” In *Lecture Notes in Computer Science #79: Language Design and Programming Methodology*, 25–35. Sydney, Australia: Springer-Verlag, 1979.

Rogers, John. “Ephemera 2009 (7).” *Kung Fu Monkey* (blog), March 19, 2009. https://kfmonkey.blogspot.com/2009/03/ephemera-2009-7.html.

Rogers, Mikeal. “Libertarianism, Meritocracy and Equality.” *GitHubGist: 7835772* (blog), December 6, 2013. https://gist.github.com/mikeal/7835772.

Rosenzweig, Roy. “Wizards, Bureaucrats, Warriors, and Hackers: Writing the History of the Internet.” *The American Historical Review* 103, no. 5 (December 1, 1998): 1530–52. https://doi.org/10.1086/ahr/103.5.1530.

Rothbard, Murray N. “The Libertarian Heritage: The American Revolution and Classical Liberalism,” January 1970. https://www.lewrockwell.com/1970/01/murray-n-rothbard/our-libertarian-heritage.

Russell, Dean. “Who Is a Libertarian?” Foundation for Economic Education, May 1, 1955. https://fee.org/articles/who-is-a-libertarian/.

Salus, Peter H. *The Daemon, the Gnu & the Penguin*. Seattle, WA: Reed Media Services, 2008.

Saner, Emine. “It Is Not a Disease, It Is a Way of Life.” *The Guardian*, August 7, 2007. https://www.theguardian.com/society/2007/aug/07/health.medicineandhealth.

Schneider, Daniel. “Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List.” Berlin, DE: ACM, 2016. https://doi.org/10.1145/2957792.2957801.

Scott, Red. “I’ve Seen a Breakdown Video of RMS. Is Richard Stallman Autistic?,” May 31, 2018. https://www.quora.com/Ive-seen-a-breakdown-video-of-RMS-Is-Richard-Stallman-autistic.

Siegert, Richard J., and Tony Ward. “Clinical Psychology and Evolutionary Psychology: Toward a Dialogue.” *Review of General Psychology* 6, no. 3 (2002): 235–59.

Silberman, Steve. “The Geek Syndrome.” *Wired*, December 1, 2001. https://www.wired.com/2001/12/aspergers/.

Slegers, John. “Why the Tech Industry Needs More Autism.” *Hacker Noon* (blog), October 29, 2015. https://hackernoon.com/why-the-tech-industry-needs-more-autists-7024a2bcc34c.

Spolsky, Joel. “My First Bill G Review.” *Joel on Software* (blog), June 16, 2006. https://www.joelonsoftware.com/2006/06/16/my-first-billg-review/.

Stallman, Richard M. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. 3rd ed. Boston, MA: Free Software Foundation, 2015.

———. “How I Do My Computing.” Richard Stallman’s Personal Site, 2006. https://stallman.org/stallman-computing.html.

———. MEME 2.04. Interview by David S. Bennahum. Web, 1996. http://memex.org/meme2-04.html.

———. “The GNU General Public License, Version 3.” Free Software Foundation, June 29, 2007. https://www.gnu.org/licenses/gpl-3.0.en.html.

———. “The GNU Manifesto.” Free Software Foundation, June 2, 2015. https://www.gnu.org/gnu/manifesto.en.html.

———. “The GNU Project.” Free Software Foundation, December 15, 2018. http://www.gnu.org/gnu/the-gnu-project.html.

StatCounter. “Mobile Operating System Market Share Worldwide.” StatCounter Global Stats, n.d. http://gs.statcounter.com/os-market-share/mobile/worldwide.

Stephenson, Neal. *In the Beginning Was the Command Line*. New York: HarperCollins Publishers, 1999.

Sykes, Charles J. “Some Rules Kids Won’t Learn in School.” *San Diego Union Tribune*. September 19, 1996. https://www.ime.usp.br/~rbrito/teaching/mack/loo/interessante.html.

Tanenbaum, Andrew S. “Some Notes on the ‘Who Wrote Linux’ Kerfuffle.” Andrew S. Tanenbaum, Professor at the Vrije Universiteit, May 21, 2004. https://www.cs.vu.nl/~ast/brown/.

Taplin, Jonathan. “Rebirth of a Nation: Can States’ Rights Save Us From a Second Civil War?” *Harper’s Magazine*, November 2018.

Tedre, Matti. “Computing as a Science: A Survey of Competing Viewpoints.” *Minds & Machines* 21 (2011): 361–87. https://doi.org/10.1007/s11023-011-9240-4.

Tennant, Don. “Asperger’s Oxymoron.” *Computerworld*, October 6, 2008.

The Center for Cognitive Science, University of Minnesota. “The Center for Cognitive Science’s Millenium Project.” The Cognitive Science Millenium Project, January 8, 2000. https://web.archive.org/web/20111011092046/http://www.cogsci.umn.edu/OLD/calendar/past_events/millennium/final.html.

The Linux Foundation. “Our Corporate Members.” The Linux Foundation, 2018. https://www.linuxfoundation.org/membership/members/.

Tiemann, Michael. “History of the OSI.” Open Source Initiative, September 19, 2016. https://opensource.org/history.

Torvalds, Linus. “Linux 4.19-Rc4 Released, an Apology, and a Maintainership Note,” September 16, 2018. https://lkml.org/lkml/2018/9/16/167.

Torvalds, Linus, and David Diamond. *Just for Fun: The Story of an Accidental Revolutionary*. New York: HarperCollins Publishers, 2001.

Vleck, Tom van. “Kenneth Lane Thompson - A.M. Turing Award Laureate.” Association for Computing Machinery, n.d. https://amturing.acm.org/award_winners/thompson_4588371.cfm.

Vries, Marc J. de, Sven Ove Hansson, and Anthony W.M. Meijers, eds. *Norms in Technology*. Vol. 9. Philosophy of Engineering and Technology. Dordrecht, NL: Springer Science+Business Media, 2013.

Warford, J. Stanley. *Computer Systems*. Sudbury, MA: Jones & Bartlett Publishers, 2009.

Wayner, Peter. *FREE FOR ALL: How Linux and the Free Software Movement Undercut the High-Tech Titans*. HarperCollins Publishers, 2000.

Wilcox, Lauren. “Embodying Algorithmic War: Gender, Race, and the Posthuman in Drone Warfare.” *Security Dialogue* 8, no. 1 (December 12, 2016): 11–28. https://doi.org/10.1177/0967010616657947.

Williams, Sam. *Free as in Freedom: Richard Stallman’s Crusade for Free Software*. Sebastopol, CA: O’Reilly Media, 2002.

Witwer, Andrea N., and Luc Lecavalier. “Examining the Validity of Autism Spectrum Disorder Subtypes.” *Journal of Autism and Developmental Disorders* 38, no. 9 (October 1, 2008): 1611–27. https://doi.org/10.1007/s10803-008-0541-2.

World Health Organization. “Autism Spectrum Disorder without Disorder of Intellectual Development and with Mild or No Impairment of Functional Language.” ICD-11 for Mortality and Morbidity

Statistics (ICD-11 MMS). Geneva, CH: World Health Organization, 2018. https://icd.who.int/browse11/l-m/en#/http://id.who.int/icd/entity/120443468.

Young, Michael Dunlop. "Down With Meritocracy." *The Guardian*, June 28, 2001. https://www.theguardian.com/politics/2001/jun/29/comment.