

**en
defensa
de
software
libre**

01

**en
defensa
de
software
libre**

01

**en
defensa
de
software
libre**

01

¡Hackers GNUíós!

Johan Söderberg



En Defensa del Software Libre

En Defensa del Software Libre es una revista de teoría sobre Software y Cultura Libres. Se edita en papel y se distribuye gratuita y libremente en formato digital.

©2019– En Defensa del Software Libre.

<https://endefensadelsl.org>

Salvo donde se exprese lo contrario, los artículos y la edición se liberan bajo la Licencia de Producción de Pares.

https://endefensadelsl.org/ppl_deed_es.html

Licencia de Producción de Pares

Ud. es libre de

- Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra
- Hacer obras derivadas

Bajo las condiciones siguientes:



Atribución – Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciente (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



Compartir bajo la Misma Licencia – Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.



No Capitalista – La explotación comercial de esta obra sólo está permitida a cooperativas, organizaciones y colectivos sin fines de lucro, a organizaciones de trabajadores autogestionados, y donde no existan relaciones de explotación. Todo excedente o plusvalía obtenidos por el ejercicio de los derechos concedidos por esta Licencia sobre la Obra deben ser distribuidos por y entre los trabajadores.

Entendiendo que

- **Renuncia** - Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor.
- **Dominio Público** - Cuando la obra o alguno de sus elementos se halle en el dominio público según la ley vigente aplicable, esta situación no quedará afectada por la licencia.
- **Otros derechos** - Los derechos siguientes no quedan afectados por la licencia de ninguna manera:
 - Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior;
 - Los derechos morales del autor;
 - Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo derechos de imagen o de privacidad.
- **Aviso** - Al reutilizar o distribuir la obra, tiene que dejar muy en claro los términos de la licencia de esta obra. La mejor forma de hacerlo es enlazar a esta página.

Índice general

1	¡Hackers GNUíos! — Johan Söderberg	9
	La izquierda política y la política de los hackers	9
	La máquina en el trabajo	13
	Descalificación de obreros, recalificación de usuarios . . .	17
	El software libre como una estrategia sindical	21
	Reconocimientos	30
	Bibliografía	31

¡Hackers GNUíos!

—*Johan Söderberg*—

Este artículo es la traducción al castellano de “Hackers GNUited!”, publicado en Free Beer 1.0 (2008). El libro completo en inglés puede encontrarse en <http://freebeer.fscons.org>. Traducción por fauno, revisión y correcciones por Leonardo G. De Luca, agosto, 2010. Este artículo y el original están licenciados bajo Creative Commons Atribución-CompartirIgual 2.5 (Corporation, s. f.)

La izquierda política y la política de los hackers

En este artículo voy a mirar el hacking desde la perspectiva del sindicalismo. El significado político del hacking informático ha intrigado a la vieja izquierda, aunque existen algunos vasos comunicantes entre el movimiento hacker y movimientos sociales más tradicionales. Los más notables de estos grupos dentro del underground informático son los autodenominados ‘hacktivistas’. Tratan de aplicar sus habilidades informáticas a una agenda política ya establecida, como el feminismo o el ecologismo (Jordan, 2002). Pero es más desafiante darle sentido a la agenda política del núcleo del movimiento hacker. De inmediato, uno se pregunta si el un-

derground informático puede siquiera considerarse un movimiento social. Muchos hackers, tal vez la mayoría, dirían que este no es el caso. A lo más, la política es tenida como un punto secundario frente a la alegría de jugar con tecnología informática (Torvalds & Diamond, 2001). Aún así, de esta apasionada afirmación de las computadoras han crecido ideas con ramificaciones políticas. Por caso, hackers que de otra manera no se considerarían ‘políticos’ tienden a oponerse a las patentes del software y la vigilancia estatal de Internet, para mencionar sólo dos ejemplos. De hecho, estos puntos de vista están tan compartidos en el underground informático que parecen más sentido común que consignas políticas. Algunos temas, como las campañas contra la expansión de las leyes de propiedad intelectual y la defensa de la libertad de expresión, han sido agregadas a agendas políticas y son promovidas activamente por grupos de lobby hacker, dos ejemplos de esto son la Fundación del Software Libre (FSF) y la Fundación Frontera Electrónica (EFF). Estas organizaciones están claramente involucradas en política, aunque sostienen que estos intereses cortan por ejes diferentes de los de la división derecha-izquierda tradicional. Cuando los científicos sociales han analizado los supuestos que descansan detrás de las declaraciones públicas de estos grupos de lobby hacker han encontrado sin embargo una cercana afinidad con el liberalismo (Coleman, 2008).

Un par de escritores de izquierda han hecho un corte en el que no interpretan el hacking como una ideología liberal. Muy al contrario, creen que el movimiento hacker podría revitalizar las viejas luchas de la izquierda, no sólo por la libertad individual sino también en contra de la injusticia y la desigualdad. El más renombrado enterado que ha expresado tales opiniones sobre el hacking es Eben Moglen. Es profesor de leyes y fue por un largo tiempo una figura maestra en la Fundación del Software Libre (FSF). Moglen es además el autor del *Manifiesto punto Comunista*, donde predijo que el anarquismo del desarrollo de software reemplazará a las firmas capitalistas como el modo más eficiente de organizar la producción en el futuro (Moglen, 1999). El estudioso de los medios Richard Barbrook razonaba en forma similar cuando desbancaba el hype acerca de los ‘mercados libres en el ciberespa-

cio' creado en los '90. En su lugar presentó su propia visión de una economía anarquista del don de alta tecnología.¹ El impulso a dar se genera automáticamente del hecho de que en Internet la gente tiene un interés propio en compartir información libremente antes que comercializarla en un mercado (Barbrook, 2002). Podría decirse que el surgimiento de Napster y generaciones posteriores de tecnologías de compartición de archivos probaría que Barbrook estaba en lo cierto. Aun más iconoclasta en su adopción de la retórica socialista es el filósofo esloveno Slavoj Zizek. Parafraseó el apoyo de Lenin a la electricidad diciendo, en tono de burla, que 'el socialismo es igual a libre acceso a Internet más todo el poder a los Soviets' (Zizek, 2002). Por lo menos unos pocos comunistas de la vieja escuela están tomando esta idea con seriedad. Creen que la tecnología informática ha provisto el vínculo faltante que al fin podría hacer de una economía planificada una alternativa viable a la economía de mercado (Pollack, 1998).

Pero estas afirmaciones positivas del hacking y la tecnología informática probablemente sean opiniones minoritarias dentro de la izquierda tradicional. Hay una sospecha profundamente arraigada entre los intelectuales de izquierda hacia la tecnología informática y, por extensión, a sus más celosos usuarios, es decir los hackers. El origen de Internet dentro de instituciones del EEUU de la Guerra Fría es suficiente para desanimar a muchos pensadores progresistas (Edwards, 1996; Shiller, 1999). Hay que agregar a esto el hype alrededor de Internet a mediados de los '90. Dió rienda suelta al viejo dilema de la 'Edad de la Información'. Esta noción data de los '50 y proviene de sociólogos conservadores estadounidenses que se propusieron desaprobando la relevancia y continuidad de los conflictos de clase. Al anunciar el fin de la sociedad industrial, buscaban probar que las tensiones entre clases se habían disuelto y la lucha ideológica entre liberalismo y socialismo se estaba volviendo obsoleta. En consecuencia, los académicos con tendencias de izquierda protestaron contra las nociones acerca del surgimiento de la Edad

¹ Traducimos "gift economy" por economía del don ya que es el término más extendido en castellano y podría decirse más exhaustivo que "regalo". Ver Ensayo sobre el don, de M. Mauss, y El enigma del don, de M. Godelier. (Nota de la traducción.)

de la Información e insistieron en la existencia y continuidad del industrialismo, capitalismo y conflicto de clases (Webster, 2002). Para probar este punto les basta con llamar atención sobre las condiciones inhumanas bajo las que la electrónica informática es manufacturada en zonas de exportación de países del tercer mundo (Sussman & Lent, 1998). Un informe de 2008 ha documentado cómo en China chicas de 16 años de edad trabajan de doce a quince horas diarias, seis o siete días a la semana, y apenas ganan para vivir (Weed, 2008). Estos descubrimientos resuenan junto a la circunstancia histórica de que las tarjetas perforadas, maquinaria de control numérico, mainframes y otros embriones de computadoras modernas fueron instrumental para volver redundantes a los obreros y degradar sus calificaciones laborales en el momento de la producción (Braverman, 1974; Kraft, 1977).

Ahora, habiendo delineado brevemente la relación confusa entre la izquierda tradicional y el empuje político de los hackers, este artículo procederá a examinar la importancia política de estos últimos a la luz de un viejo debate acerca de la maquinaria fabril y el trabajo asalariado. El Debate Braverman, como es conocido según el autor que inició la controversia, se retrotrae a los '70. Harry Braverman publicó un libro en el que argumentaba que la descalificación del trabajo era una cualidad inherente al capitalismo. La razón era que los gerentes tratan de independizarse de los obreros altamente calificados para mantener bajos salarios y debilitar políticamente a los sindicatos. Braverman encontró apoyo a su hipótesis en los escritos de los pioneros de la filosofía gerencial. La figura pivote entre ellos, Winston Taylor, fue el fundamento de lo que se conocería como 'gerenciamiento científico' o 'taylorismo'. Una idea central del gerenciamiento científico es que la línea de producción debe reestructurarse de manera que las tareas puedan realizarse mediante rutinas simples que requieran un mínimo de calificación de los empleados. Taylor argumentó que esto podía lograrse con la introducción de maquinaria fabril. Braverman mostró cómo esta estrategia se estaba desplegando en la industria pesada a mediados del siglo XX.

Este punto de vista puede servir como lente con la que mirar el significado político de la maquinaria informática y el hacking

de esta. La novedad de este argumento es que el análisis de los hackers está formulado desde una perspectiva orientada a la producción, en oposición a la perspectiva de los derechos del consumidor. Se argumentará que el surgimiento del Software Libre y Abierto (FOSS) puede trazarse hasta el conflicto industrial entre gerentes y trabajadores. Además, la similitud entre la lucha de los trabajadores contra la maquinaria fabril y la lucha del movimiento hacker contra el software propietario va a ser puesta en relieve. El libre acceso al código fuente, preocupación clave de los hackers, contradice el sistema fabril y la lógica del gerenciamiento científico en la programación informática (Hannemyr, 1999). Aunque la situación de los programadores comparada con la de los obreros es muy diferente en muchos aspectos, el artículo muestra que ambos grupos están preocupados en la meta de preservar sus calificaciones y la autonomía obrera frente al rápido cambio tecnológico. La demanda de los hackers de que el código fuente debe ser libremente accesible puede interpretarse como parte de una estrategia orientada a preservar el saber-cómo de los programadores y su control sobre las herramientas de su oficio.

La máquina en el trabajo

Los sentimientos ambivalentes de entusiasmo y miedo que a menudo evoca la tecnología informática entre la gente tienen un precedente histórico. En el amanecer de la revolución industrial, se debatió fuertemente en todos los segmentos de la sociedad qué efecto tendría la mecanización sobre el ser humano, tanto social como espiritualmente (Berg, 1980). Incluso algunos de los adelantados de la teoría económica liberal, como David Ricardo, admitieron que la clase trabajadora tenía buenas razones para resentir de la maquinaria fabril (Riccardo, 1821). La miseria que cayó sobre los trabajadores que estaban subyugados a la maquinaria y la disciplina fabril fue vívidamente descrita por James Kay, un reformista social que trabajó como doctor en las villas:

“Mientras la máquina anda la gente debe trabajar —
hombres, mujeres y niños están uncidos juntos con hie-

rrero y vapor. La máquina animal —frágil en el mejor de los casos, sujeta a mil fuentes de sufrimiento— está encadenada a la máquina de hierro, que no conoce sufrimiento ni fatiga.” (Kay, 1832)

Escritores sobre el gerenciamiento tempranos como Andrew Ure y Charles Babbage dieron la bienvenida a esta oportunidad y aconsejaron a los dueños de las fábricas sobre cómo diseñar maquinaria para mantener a los obreros dóciles e industriosos (Babbage, 1971; Ure, 1835). Sus testimonios informaron el análisis del capitalismo de Karl Marx, quien denunció la maquinaria fabril como el ‘modo material de existencia del capital’. Pero él también cualificó su crítica contra la tecnología agregando que: “Tomó tiempo y experiencia para que los trabajadores aprendieran a distinguir la maquinaria de su empleo por el capital, y en consecuencia a transferir sus ataques de los instrumentos materiales de producción a la formación social que utiliza esos instrumentos.” (Marx, 1976) Así Marx renunció a la estrategia de rompimiento de máquinas que fue el sello de los Luditas. Los Luditas consistían en peñadores, tejedores y artesanos que sintieron que su oficio era amenazado por la introducción de nuevos telares y una subsecuente reorganización de la industria textil. Se hicieron ataques nocturnos para destruir molinos de lana y marcos de tejido de los que los ‘maestros tejedores’ eran dueños. Estas actividades alcanzaron su punto cúlmine en 1811-1813 y una vez la corona inglesa tuvo que desplegar 14.400 soldados en la región para poder aplastar las insurgencias nocturnas. Remarcablemente, fueron movilizados más soldados ingleses contra los Luditas de los que habían sido enviados a Portugal cuatro años antes para enfrentar al ejército de Napoleón (Sale, 1995). En su clásico reexamen del levantamiento Ludita, Eric Hobsbawm mostró que el rompimiento de máquinas no fue una resistencia fútil contra la tecnología y el progreso, como se hizo ver después. En su lugar la interpretó como un método de ‘negociación colectiva mediante el disturbio’. Romper la maquinaria era una opción, pero los obreros también pudieron presionar a sus empleadores prendiendo fuego los lugares de trabajo o enviando amenazas anónimas. Hobsbawm concluye que, a juzgar por la habilidad de los trabajadores de preservar sus salarios y condiciones laborales, tu-

vieron un éxito moderado (Hobsbawm, 1952).

La lectura equivocada de la rebelión Ludita como alocada, irresponsable y, más importante, sin relación alguna con la política, se asemeja al retrato de los hackers que se hace en los medios de noticias actuales. Andrew Ross protestó contra la imagen del hacker como un criminal menor, un bromista juvenil o, alternativamente, un yuppie de la Era de la Información. Hace hincapié en que el sabotaje espontáneo hecho por empleados contribuye la mayor parte de tiempo muerto en las oficinas. Estos ataques a menudo no son reportados ya que los gerentes prefieren culpar a adversarios externos. Con esta observación en mente, sugiere una definición de hacking más amplia:

“Mientras un pequeño número de usuarios de computadoras se categorizaría a sí mismo como ‘hacker’, existen razones de peso para extender la definición restringida de *hacking* por debajo y a través de la jerarquía de analistas de sistemas, diseñadores, programadores y operadores para incluir a todos los trabajadores de alta tecnología –sin importar cuán inexpertos– que puedan interrumpir, molestar y redirigir el suave flujo de comunicaciones estructuradas que dicta su posición en las redes sociales de intercambio y determina el paso de su agenda de trabajo.” (Ross, 1991)

La sospecha de Andrew Ross es confirmada por estudios conducidos por organizaciones de empleadores. Que el personal provoque la falla del equipamiento informático de sus empleadores es el escenario más común, más costoso y más temido por las firmas que la intrusión de usuarios informáticos externos. De acuerdo a una encuesta realizada en 1998 por la Iniciativa por la Seguridad Informática en conjunto con el FBI, el costo medio de un ataque informático exitoso en los EEUU por un usuario externo es de US\$56.000. En comparación, el costo medio de actos maliciosos hechos por usuarios internos (es decir, empleados) se estimó en US\$2,7 millones (Shell & Dodge, 2002). La afición de los empleados por atacar los sistemas informáticos de sus empleadores subraya el rol de la computarización en la transformación de las

condiciones de trabajo de los trabajadores de cuello blanco. La comparación de Ross con el sabotaje seguramente despertará algunas objeciones entre los hackers ‘reales’. Aquellos en el movimiento hacker que quieren adecuarse a la definición tratan de contrarrestar el estereotipo negativo de los hackers que hacen los medios al diferenciar entre hackers originales y los así llamados crackers. El primer nombre está reservado a usos creativos que contribuyen a proyectos de software socialmente útiles. Las connotaciones negativas del crimen informático están reservadas para el último grupo².

Estos esfuerzos por mejorar las relaciones públicas de los hackers meramente subrayan el paralelo histórico con la militancia laboral sugerido arriba. El movimiento sindicalista también ha reescrito su propia historia para que el sabotaje, las huelgas ilegales y los actos de violencia queden fuera de la foto. En efecto, los sindicatos han tenido bastante éxito en formalizar el conflicto entre trabajo y capital como un problema de negociación institucionalizada. No obstante, puede decirse que la negociación colectiva del trabajo todavía descansa sobre la amenaza velada del sabotaje, las huelgas y los disturbios (Brown, 1977). De la misma manera, entiendo la distinción entre hackers y crackers como una construcción discursiva que no retrata certeramente las raíces históricas ni el solapamiento actual de esta subcultura. En cambio, busca redefinir el significado de hacking y dirigirlo en una dirección particular. A pesar del éxito de esta retórica, la liberación de warez, la rotura de cifrados, y el crackeo de servidores corporativos juegan sin embargo un papel en la lucha más amplia por mantener la información libre.

Habiendo dicho esto, el lector estaría en lo correcto al objetar que la motivación de los Luditas y trabajadores para rechazar la maquinaria fabril y de oficina es muy diferente de la motivación de los hackers que luchan contra el software propietario. Para el último grupo, las computadoras se revelan como bienes de consumo y fuentes de estímulo. Podría decirse que su relación con la

²Por ejemplo el *Jargon File*, que es considerada la fuente autorizada de argot hacker, hace malabares para distinguir entre crackers y hackers ‘reales’: <http://ur1.caf6o3> (última visita: 27-05-2009).

tecnología es de pasión más que de hostilidad. Aún cuando los hackers (crackers) sabotean servidores corporativos, este es un acto de alegría. Los trabajadores de oficina descontentos podrían obtener placer al destruir la computadora de su empleador, pero todavía es significativo decir que su acto nace del resentimiento contra su situación. Esta diferencia en motivación, sin embargo, no descarta la posibilidad de que los hackers compartan terreno con los antiguos rompedores de máquinas. Ambos están atrapados en un combate luchado en el terreno del desarrollo tecnológico. Incluso podría ser que la apasionada afirmación de la tecnología hecha por los hackers ofrezca una línea de ataque aún más subversiva, en comparación a, por ejemplo, la insurgencia Ludita. Aunque es incorrecto decir que los Luditas estaban en contra de la tecnología *per se*, es verdad que defendían una tecnología desactualizada contra la nueva del sistema fabril. Así es que parece que su causa estaba perdida antes de empezar. Los hackers, en contraste, tienen una tecnología propia sobre la que trabajar. Pueden declarar plausiblemente que su modo de escribir código es más avanzado que el ‘modelo fabril’ de desarrollo de software propietario.

Descalificación de obreros, recalificación de usuarios

Es una dialéctica extraña la que llevó a la situación actual en la que los hackers recuperan tecnología informática de las compañías e instituciones gubernamentales. Las pistas de cómo se llegó a esta situación pueden encontrarse en lo que se llamó el Debate Braverman. La controversia tuvo lugar contra el telón de la idea acerca de la llegada de una era post-industrial (Bell, 1973). Dos décadas después, la misma idea fue reempaquetada como ‘el surgimiento de la Era de la Información’ o la ‘Sociedad en Red’. Esta noción ha tenido muchos tonos pero invariablemente pinta un futuro brillante donde el capitalismo avanzará más allá del conflicto de clases y el trabajo monótono. Crucialmente, esta transición no fue traída a través de la lucha social sino que se debe exclusivamente a la trayectoria interna del desarrollo tecnológico.

Harry Braverman apuntó a uno de sus supuestos principales, la de que las capacidades de los trabajadores se modernizarían cuando los trabajos de obrero fueran reemplazados por trabajos de cuello blanco. Insistió en que la lógica del capital es la de descalificar la fuerza de trabajo, sin tener en cuenta si se emplea en una fábrica o en una oficina. En lugar de una modernización general de calificaciones en la sociedad, predijo que el crecimiento de la así llamada ‘economía del servicio’ pronto resultaría en que los trabajadores de cuello blanco confronten la rutinización y descalificación así como la que obreros fabriles habían pasado antes.

“Por lejos lo más importante en la producción moderna es la división de procesos complejos en tareas simples que son realizadas por trabajadores cuyo conocimiento es virtualmente nulo, cuyo así llamado entrenamiento es breve, y que entonces pueden ser tratados como partes intercambiables.” (Braverman, 1998a)

Su afirmación fue rebatida por los sociólogos industriales. Reconocían que la descalificación del trabajo está presente en industrias maduras, pero argumentaban que esta tendencia se contrabalanceaba por el establecimiento de nuevos puestos de trabajo con mayores calificaciones en otros lugares de la economía. A primera vista, la emergencia de la profesión de programador pareció haber probado que sus críticos tenían razón. Uno de ellos, Stephen Wood, reprochó a Braverman por idealizar al trabajador artesanal del siglo XIX. Wood señaló la difusión de la alfabetización para probar que las calificaciones también habían aumentado en la sociedad moderna (Wood, 1982). Su comentario es intrigante ya que trae una sutileza que se perdió en el calor del intercambio. No es la descalificación *per se* el objetivo del capital, sino el hacer reemplazables a los trabajadores. Cuando las tareas y cualificaciones se estandarizan, el trabajo será barato y falto de fuerza política. Desde este punto de vista, no importa realmente si las calificaciones de los trabajadores se nivelan en un equilibrio más bajo o más alto. La alfabetización universal es un ejemplo de lo último.

En este sentido puede decirse que la alfabetización es análoga en el presente a las campañas por la alfabetización informática

que llaman a cerrar ‘la brecha digital’. En un sentido trivial, las calificaciones han aumentado en la sociedad cuando más gente sabe cómo usar computadoras. Uno puede sospechar que un fuerte ímpetu para esto es, sin embargo, que la alfabetización digital reduce una mayor inercia en el esquema de ‘aprendizaje de por vida’, esto es, el tiempo que toma a los humanos aprender nuevas habilidades. Una vez que los trabajadores adquieren habilidades básicas para navegar en un ambiente digital, toma menos esfuerzo aprender una nueva ocupación cuando su oficio anterior se vuelve redundante. Esta interpretación de alguna manera cínica de la alfabetización informática puede ilustrarse con una referencia a la industria de la imprenta. Los oficios gráficos tradicionales toman muchos años para dominarse y requieren grandes y costosas instalaciones. La militancia sindical que caracterizó la industria de la imprenta se fundaba sobre este monopolio del conocimiento por parte de los trabajadores. La introducción de procesos informáticos fue decisiva para romper la fuerza de los trabajadores gráficos (Zimbalist, 1979). Las computadoras personales pueden verse como una extensión de este desarrollo. La mediación por el software permite que la simple capacidad de navegar una interfaz gráfica se convierta en múltiples capacidades. Con una computadora corriendo GNU/Linux y Scribus, por ejemplo, el usuario puede comandar el lenguaje de máquina de la computadora e imitar los oficios de la imprenta y la tipografía. Se requiere muy poco entrenamiento para usar estos programas, comparado al tiempo que le toma a un trabajador gráfico dominar su oficio. Esto sugiere que la alfabetización informática reduce la inercia del aprendizaje humano y vuelve las calificaciones de los trabajadores más intercambiables. Los escritores liberales interpretan este desarrollo como un ejemplo del crecimiento lineal del aprendizaje y la educación correspondiente a la llamada ‘sociedad del conocimiento’. Desde la perspectiva de la teoría del proceso laboral, muy al contrario, el mismo desarrollo es visto como una degradación de las capacidades de los trabajadores y en última instancia apunta a debilitar la posición negociadora de los sindicatos.

El clásico estudio de David Noble sobre la introducción de maquinaria de control numérico en la industria pesada a mediados

del siglo XX provee el vínculo faltante entre el argumento de Braverman sobre la descalificación y la discusión actual sobre computadoras y hackers. Una cosa sobre la que su estudio arroja luz es cómo la universalidad de la herramienta informática estaba pensada para trabajar en ventaja de los gerentes. Su esperanza era la de debilitar la posición de los maquinistas calificados. La maquinaria de propósito especial había fallado en reemplazar a estos obreros, ya que aún tenían que tomarse iniciativas en la línea de producción para integrar los estadios separados de la producción especializada. En contraste, las máquinas de propósito general simulaban la versatilidad de los seres humanos, por lo que estaban mejor capacitadas para reemplazarlos (Noble, 1984). Esta conexión histórica es importante de enfatizar porque ahora es un lugar común que la universalidad de las herramientas informáticas se asuma como una cualidad inherente de la tecnología de la información en sí misma. De ahí que la trayectoria hacia herramientas universales se haya desprendido de la lucha y se atribuya en cambio a la gracia del desarrollo tecnológico.

Decir eso no nos obliga a condenar la tendencia a un nivelamiento de las calificaciones productivas y el crecimiento de herramientas universales como las computadoras. Al contrario, en fuerte contraste con el retrato negativo de Harry Braverman como un neo-Ludita, Braverman reconoce que la unificación de la fuerza de trabajo causada por la maquinaria acarreó un potencial positivo.

“El proceso re-unificado en el que la ejecución de todos los pasos es construida en el mecanismo de trabajo de una sola máquina parecería ahora volverlo adecuado a un colectivo de productores asociados, ninguno de los cuales necesita gastar toda su vida en una sola función cualquiera y donde todos pueden participar en la ingeniería, diseño, mejora, reparación y operación de estas máquinas aún más productivas.” (Braverman, 1998b)

Con una herramienta universal, la computadora, y la casi universal calificación de usarla, el público puede involucrarse en cual-

quier cantidad y tipo de actividades productivas. Es desde este ángulo que podemos empezar a dar sentido a la tendencia actual de ‘empoderamiento del usuario’. En otras palabras: El desplazamiento del trabajo organizado de los baluartes del aparato de producción capitalista, a través de la combinación de descalificación y recalificación, ha preparado el terreno para esquemas de innovación asistidos por computadoras y centrados en el usuario. Porque programas como *Inkscape* y *Scribus*, y sus equivalentes propietarios, están substituyendo formas tradicionales de imprenta y tipografía, una multitud de gente puede producir pósteres y panfletos, instantáneamente aplicables a sus luchas locales. Las compañías tienen más difícil el control de la actividad productiva ahora que cuando los instrumentos de trabajo estaban concentrados en las manos de unos pocos, aunque relativamente poderosos, empleados. Lo que es cierto para el diseño gráfico se aplica igualmente a la escritura de código de software y el desarrollo de la tecnología informática. Aquí la cara de Jano del software se pone en primer plano: con la misma flexibilidad y precisión con la que el código de software puede diseñarse para controlar trabajadores subordinados, esta misma facilidad permite a muchos más tomar parte en el proceso de escribirlo. Aunque formas embrionarias de tecnología informática, como la maquinaria de control numérico, fueron introducidas en los lugares de trabajo por los gerentes para independizarlos de trabajadores sindicalizados y calificados; como un efecto lateral, la tecnología informática ha contribuido al establecimiento de procesos productivos centrados en el usuario, parcialmente independientes de gerentes y fábricas. La comunidad de desarrollo de software libre puede tomarse como ilustración de esto.

El software libre como una estrategia sindical

El apoyo corporativo hacia la comunidad de Software Libre y Abierto (FOSS) debe verse contra el trasfondo de un mercado de trabajo reestructurado. Durante las últimas décadas, los sociólogos

industriales han documentado una tendencia donde la fábrica está perdiendo su anterior estatus como el modelo de producción principal. La producción se vuelve cada vez más descentralizada y dispersa en una red de subcontratistas, trabajadores independientes, esquemas de trabajo en casa y franquicias (McChesney, Wood, & Foster, 1998). Las compañías ahora pueden agregar a las comunidades voluntarias de desarrollo a la lista de formas heterogéneas de contratación de trabajo. O, para decirlo con una frase pegadiza, el trabajo es *outsource* y *opensource*. La oportunidad para recortar drásticamente los costos laborales para el mantenimiento de software ha atraído a instituciones estatales, vendedores, proveedores de servicio y manufactureras de hardware hacia el FOSS. Los ahorros que hacen gigantes como IBM, el ejército de EEUU y la ciudad de Munich, para mencionar un par de casos de alto perfil, ha creado el espacio para que firmas de software especializadas vendan productos y servicios de software libre. Este análisis es consistente con la crítica de Tiziana Terranova en la que el involucramiento de trabajo libre se ha vuelto estructural a la economía cultural. Ella protestaba contra los muchos ahelos y reclamos hechos sobre la tendencia al consumo mediático activo, primero celebrada en la disciplina de los estudios culturales desde los '80 en adelante y más recientemente actualizada con el hype alrededor de la Web 2.0. En respuesta a estas declaraciones a menudo infundadas, Terranova dijo que el capital siempre ha anticipado al consumidor activo en sus estrategias de negocio (Terranova, 2000). Su argumento provee un correctivo a los aplausos acríticos a la subcultura del *fan fiction*, la licencia Creative Commons y otras expresiones de 'medios participativos'. No obstante, en mi opinión, los críticos de izquierda como Terranova han estado ansiosos de denunciar la explotación económica del trabajo voluntario y han fallado así en ver el cambio político en potencia que también existe en algunos de esos casos.

La relevancia de mi objeción debe decidirse caso por caso. Mientras concedo que la interactividad de los videojuegos y los esfuerzos voluntarios de escritores de *fan fiction* probablemente no resulten en ningún cambio político sustancial, la interactividad y el donar de los desarrolladores de software libre no puede descri-

birse con el mismo pincel. Aquí debe tomarse en cuenta que el código de software es regalado conjuntamente con una meta política claramente articulada: hacer del software libre el estándar en computación. Es verdad que este punto de vista no es anti comercial en un sentido concreto. Como el lector probablemente sepa, la Licencia Pública General (GPL) protege el derecho del usuario a correr el software con cualquier propósito, incluido el comercial (Gay, 2002). En la práctica, por supuesto, esta opción está limitada por el hecho de que la GPL también permite que las copias vendidas sean copiadas y distribuidas gratuitamente. Mientras la licencia libre reside perfectamente en un mercado libre idealizado, difícilmente lo sea dentro del mercado existente que presupone siempre los cuasi monopolios y las regulaciones estatales (Polanyi, 2001).

Esto explica de alguna manera por qué la derecha política tiene dos visiones sobre las licencias de software libre. Los autoproclamados *libertarians*³, como Eric Raymond, ven el crecimiento de los modelos de negocio basados en código abierto como una mejor aproximación al libre mercado. Detrás de esta afirmación descansa un entendimiento del capitalismo como básicamente idéntico a sus instituciones, es decir propiedad privada, libre mercado y contratos. Pero esa mirada desecha otras posibles definiciones de capitalismo que hacen énfasis en el capital como autoexpansión del dinero, o, en otras palabras, en la acumulación. Este último punto de vista es central al análisis que hace Marx del capitalismo, pero también es muy cercano a las preocupaciones de los ‘capitanes de la industria’. Con esto en mente, puede ser interesante notar investigaciones de mercado que *señalan que la adopción de aplicaciones de FOSS* en negocios está comiéndose los ingresos anuales de vendedores de software propietario en US\$60 billones por año. Crucialmente, las pérdidas del mercado de software propietario son desproporcionadas con respecto al tamaño de los nuevos mercados de FOSS, por la simple razón de que mucho de este no se paga.⁴ De ahí que la oposición al FOSS por parte de sectores de

³No traducimos *libertarian* para no confundir con libertarias de raíz anarquista. (Nota de la traducción.)

⁴El reporte de investigación de mercado al que se hace referencia es Ten-

la industria no sea necesariamente una postura equivocada, como generalmente se quiso hacer pasar. Esta oposición alcanzó su clímax en la corte en el caso del Grupo SCO contra vendedores corporativos de GNU/Linux que terminó en 2007. Durante el caso, el oficial ejecutivo del Grupo SCO, Darl McBride, escribió una carta abierta al Congreso de EEUU donde acusaba a sus competidores de ser ingenuos al apoyar las licencias FOSS: ‘A pesar de esto, estamos determinados a llevar estos casos legales hasta el final porque estamos firmes en nuestra creencia de que la difusión incontrolada del software de Código Abierto, bajo la GPL, es una amenaza mucho más seria para nuestro sistema capitalista de lo que las corporaciones estadounidenses creen.’⁵

Por lo menos, estas preocupaciones entre algunas partes de la industria informática muestran que los desarrolladores de software libre no pueden adscribirse como meras víctimas insospechadas de la explotación comercial. Tal vez esté más justificado decir que los hackers, al ofrecer su trabajo libremente, están chantajeando a las corporaciones para que adopten y difundan el modelo de desarrollo de FOSS. Ninguna compañía que responda al imperativo del mercado de reducir costos puede argumentar contra el trabajo libre (libre como en cerveza gratis). Mi hipótesis es que la militancia por las licencias libres puede interpretarse a la luz de la profesión emergente del programador de computadoras. Esta sugerencia está lejos de ser obvia ya que la identidad hacker está atada a la noción de ser un hobbista, o, en otras palabras, un no-profesional, no-empleado. Contradiciendo esta autoimagen, sin embargo, los números dicen que la mayoría de la gente que contribuye a proyectos de software libre está trabajando en la industria informática o se está entrenando para convertirse en profesionales informáticos (Lakhani & Wolf, 2005). Por lo tanto, no sería inverosímil conectar los puntos entre los hackers y el mercado laboral que los espera. En efecto, esta línea de razonamiento ya ha sido intentada en el famoso artículo de Josh Lerner y Jean Tirole

dencias en Código Abierto y fue publicado por el *Grupo Standish*. Dado que el acceso a este informe es restringido, la información sobre este viene de los medios de noticias (Broersma, 2008)

⁵<http://ur1.caffo4> (fecha de acceso: 01-11-2009).

(Lerner & Tirole, 2002). Ellos trataron de enmarcar el supuesto altruismo de los desarrolladores de software libre en la presunción de la teoría económica neoclásica acerca del ‘hombre racional económico’. Los dos autores concluyen que los hackers regalan código a cambio de nada para crearse una reputación y mejorar sus chances de ser empleados más tarde. Sin negar que estos casos puedan existir, no estoy de acuerdo con la presunción del individualismo metodológico que sostiene su pensamiento. Cuando digo que las licencias de software libre pueden beneficiar los intereses laborales de los programadores informáticos, no quiero decir que es una estrategia racionalmente calculada o que es una explicación exhaustiva de por qué los hackers licencian su software bajo la GPL. Además, en contraste con Lerner y Tirole, no pienso que estos intereses laborales se persiguen exclusivamente a través de estrategias individuales. Adicionalmente a mejorar su propia reputación, los hackers individuales contribuyen a cambiar el mercado de trabajo para los programadores como un colectivo.

Suena contraintuitivo que los programadores mejoren su fuerza de negociación vis-à-vis las firmas al regalar su trabajo a empleadores potenciales. Déjeme empezar por regresar a Harry Braverman. Él enfatizó que el mismo diseño de la fábrica ponía al operador de la maquinaria en desventaja. El obrero sólo podía emplear sus calificaciones cuando le daban acceso a la maquinaria. Desafortunadamente, la escala y el modo de organización de la fábrica ya estaban sesgados jerárquicamente. El capitalista tenía una ventaja gracias a la propiedad de las máquinas y los edificios, sin los cuales los obreros no podían emplear sus habilidades. Los únicos pedacitos de negociación que tenían los obreros eran sus calificaciones y su conocimiento íntimo del proceso de producción. Fue así como Braverman explicó también la tendencia de los capitalistas a desarrollar nuevas tecnologías que reducen el trabajo calificado. Lo que pasó desde que Harry Braverman hizo su análisis en los ‘70 es que la fábrica fordista de gran escala se volvió obsoleta en muchos sectores de la economía. Esto es particularmente verdadero para la industria informática. Las herramientas productivas (computadoras, redes de comunicación, algoritmos de software e información) están disponibles en tales cantidades que se han convertido en el

estándar común en lugar de ser una ventaja competitiva contra otros propietarios (capitalistas) y un umbral para los no propietarios (obreros). Una horda de sociólogos industriales y filósofos gerenciales han escrito sobre esta tendencia desde principios de los '80 (Zuboff, 1998). Es un truísmo de este cuerpo literario afirmar que los empleados, no el parque fabril, son en la actualidad el recurso más valioso en una corporación moderna. Está afirmación está nublada por la retórica, pero su validez puede probarse contra la adopción de 'acuerdos de no divulgación' dentro de la industria informática. Es aquí donde se establece que el empleado no tiene permitido divulgar información sensible de la firma. Otro tipo de cláusulas que a veces se incluyen en el contrato de empleo suelen producir el mismo efecto, es decir, prevenir fugas, prohibir al programador trabajar en tareas similares para un competidor después de haber dejado al empleador actual. Estos acuerdos pueden tomarse como testimonios de que el conocimiento y las calificaciones de los programadores se están volviendo cada vez más preciosos para que las firmas quieran controlarlos. Argumentaré que estas prácticas, aunque tienen muy poco que ver con la regulación del copyright, sin embargo apoyan mi afirmación de que las licencias propietarias y libres afectan la posición negociadora de los desarrolladores de software.

La justificación de estos diferentes tipos de acuerdos contractuales es la necesidad de prevenir la fuga de secretos de negocio hacia la competencia. No obstante, como un efecto lateral, previene que los programadores se muevan libremente a otras posiciones similares de su oficio. Dado que el programador se convierte en un especialista en el campo en el que está trabajando, puede hacerse difícil encontrar trabajo en un puesto distinto. El significado de esta observación se vuelve más claro si se lo contrasta con el estudio etnográfico de Sean O'Riain de un grupo de técnicos de software de una firma informática de Irlanda. Aquí se prueba que es muy difícil para los sindicatos organizar a estos trabajadores. Ya que los trabajos se proveen a destajo, las estrategias colectivas de los sindicatos carecen de agarre. Una de las conclusiones de O'Riain es que la movilidad se convierte en el medio principal por el que los empleados negocian sus condiciones laborales y salarios

(O’Riain, 2004). Al tener en cuenta este hecho, el significado de los acuerdos contractuales mencionados arriba debe ser reconsiderado. Las limitaciones que ponen a la capacidad de los empleados a ‘votar con los pies’ significa que la firma regana la ventaja. En cuanto al alcance en que los acuerdos de no divulgación y otras cláusulas son utilizados en la forma maquiavélica que se esboza aquí, es algo que queda por investigar empíricamente. Lo que me interesa en este artículo, no obstante, es que el mismo argumento puede aplicarse a las licencias propietarias en general.

La propiedad intelectual⁶ también se justifica por la necesidad de las firmas de proteger su conocimiento de los competidores. Una justificación complementaria es que la propiedad intelectual es requerida para que los productores puedan cobrar por la información a los consumidores del mercado. Pero la propiedad intelectual también afecta la relación entre la firma y sus empleados, tema menos discutido. Puede decirse que las licencias propietarias previenen la movilidad de los empleados. Asegura que los conocimientos de los programadores queden encerrados en un estándar propietario del que la firma es dueña. Puede trazarse un paralelo en cómo el obrero depende del parque de maquinarias del que el industrialista es dueño. Sin acceso a la fábrica el trabajador no puede emplear sus habilidades productivamente. En la industria de la programación, como se mencionó anteriormente, muchas de las herramientas con las que trabaja el programador están disponibles como bienes informáticos baratos (computadoras, etc.). De ahí que la compañía no tiene ventaja sobre el trabajador para proveer estas instalaciones. Pero cuando el código fuente está encerrado por el copyright y las patentes de software, se requieren grandes cantidades de capital para acceder a las herramientas de programación. Como consecuencia, la licencia de software garantiza a la compañía una ventaja sobre el trabajadorprogramador.

⁶Muchos críticos de la regulación del copyright y las patentes rechazan las palabras ‘propiedad intelectual’. En su opinión, estas palabras están cargadas de connotaciones que confunden al público. En su lugar utilizan las palabras ‘monopolio intelectual’. No estoy convencido por este argumento aunque no queda espacio para desarrollar mi contraposición aquí. Es suficiente decir que uso las palabras ‘propiedad intelectual’ en el artículo porque pienso que la asociación a otras formas de propiedad está enteramente justificada.

Este razonamiento teórico es tanto más difícil de probar empíricamente que la afirmación hecha antes de que las cláusulas en el contrato de empleo podrían usarse para restringir la movilidad de los trabajadores. Aún más, podría ser en un orden de magnitud superior en importancia para las condiciones de trabajo en el sector informático. En efecto, este aspecto de las licencias propietarias orientado a la producción podría ser tan significativo como las justificaciones oficiales a favor de la regulación de la propiedad intelectual, es decir a favor de regular la relación entre la compañía y sus clientes y competidores. Si mis razonamientos hasta ahora son correctos, entonces la Licencia Pública General debería leerse bajo la misma luz. Fui llevado a este pensamiento cuando leí el estudio autorizado de Glyn Moody sobre el modelo de desarrollo del FOSS. Hace la siguiente observación acerca de las condiciones excepcionales de las compañías especializadas en vender servicios relacionados al software libre:

“Porque el ‘producto’ es código abierto, y está disponible libremente, los negocios deben necesariamente basarse alrededor de un tipo distinto de escasez: las capacidades de la gente que escribe y provee ese software.” (Moody, 2001)

En otras palabras, cuando el código fuente se hace disponible públicamente bajo la GPL, las únicas cosas que se mantienen escasas en el mercado son las habilidades que se requieren para emplear productivamente las herramientas de software. Y este recurso es inevitablemente la facultad del ‘trabajo vivo’, para seguir la terminología de Karl Marx. Es así que los programadores pueden obtener una ventaja sobre el empleador cuando están negociando sobre su salario y condiciones laborales. La licencia libre nivela el campo de juego al asegurar que todos tengan acceso al código fuente. Terranova y estudiosos similares están en lo cierto cuando señalan que las compañías multinacionales están en mejor posición para explotar el valor comercial de las aplicaciones de software libre que cualquier programador individual. El ahorro que hace IBM al usar Apache en sus servidores es, medido en números absolutos, muchas veces mayor a los golpes de suerte de cualquier programador que ha contribuido al proyecto. Aún

así, en una segunda lectura, el programador podría estar mejor si existiera un mercado de trabajo para desarrolladores de software libre, en comparación a que no hubiere ocupaciones de este tipo disponibles. Al publicar software bajo licencias libres, el hacker individual no solo está mejorando su reputación y expectativa de empleo, como han enfatizado Lerner y Tirole. También contribuye al establecimiento de un mercado de trabajo donde las reglas del juego se han reescrito, para él y para todos los demás de su oficio. Puede interpretarse como una forma de acción colectiva adaptada al tiempo del individualismo rampante.

Queda por verse si el establecimiento de un mercado de trabajo en desarrollo de software libre se traduce en mejores condiciones laborales, salarios más altos y otros beneficios asociados al activismo sindical. Tal hipótesis debe sustanciarse con datos empíricos. Se necesita investigación comparativa entre gente trabajando independientemente como programadores de software libre y aquellos que trabajan con software propietario. Tal comparación no debe, sin embargo, enfocarse en aspectos monetarios. Tanto más importante es el costado subjetivo de la programación. Un ejemplo de esto es el descubrimiento consistente de que los hackers consideran más divertido participar en proyectos de software libre que trabajar con código propietario (Lakhani & Wolf, 2005). Tampoco creo que las estrategias sindicales escondidas son la única explicación de que los hackers publiquen bajo la GPL. Muy probablemente, la preocupación por las libertades civiles y un ethos antiautoritario dentro de la subcultura hacker sean factores más importantes. Los hackers son un grupo muy heterogéneo para incluirlos a todos en una sola explicación. Pero me atrevo a decir que la expectativa de trabajo merece más atención de la que se le ha dado en la prensa popular y la literatura académica hasta ahora. Aunque no faltan críticas hacia la regulación de la propiedad intelectual, estas objeciones tienden a formularse como una defensa de los derechos del consumidor dentro de la tradición política liberal.

Hay, por supuesto, algunas excepciones dignas de notar. Gente como Eben Moglen, Slavoj Žižek y Richard Barbrook han reaccionado contra la ideología liberal implícita en mucha de la charla acerca de Internet y temas relacionados. Lo han hecho al cortejar

la retórica revolucionaria de la Segunda Internacional. Sus ideas son originales y atractivas y a menudo ricas en entendimiento. Sin embargo, la retórica revolucionaria suena extrañamente fuera de lugar cuando se aplica a hackers pragmáticos. Los militantes del software libre podrían estar mejor si buscaran un contrapeso a la hegemonía del liberalismo en la rama reformista del movimiento obrero, es decir en el sindicalismo. Creo que tal estrategia tendrá más sentido mientras más madure la industria informática. De acuerdo con la línea general de argumentación de Harry Braverman, la profesión de ingeniería de software ya ha sido privada de mucho de su estatus anterior. En efecto, desde los tempranos '60, los escritores de revistas de gerenciamiento han llamado repetidamente al subyugamiento de los programadores bajo el mismo régimen fabril que previamente, y en parte a través de la introducción de la maquinaria informática, había sido impuesto a los trabajadores de cuello azul (Dafermos & Söderberg, 2009). Con esta historia en mente me gustaría proponer que la militancia por el software libre, en lugar de caer en la enmienda de libre expresión de la Constitución estadounidense, pueda tomar su credo de la 'Declaración de Derechos Tecnológicos'. Esta declaración fue escrita en 1981 por la Asociación Internacional de Maquinistas en el medio de un conflicto industrial rabioso:

“Las nuevas tecnologías de automatización y las ciencias que las sostienen son el producto de [un proceso] mundial de acumulación de conocimiento que duró siglos. De este modo, los trabajadores y sus comunidades tienen el derecho a compartir las decisiones y el provecho sobre la nueva tecnología.” (Shaiken, 1986)

Reconocimientos

El autor quisiera agradecer al editor, Stian Rødven Eide, así como a Michael Widerkrantz y Don Williams, por sus comentarios constructivos a los borradores de este artículo.

Bibliografía

Babbage, C. (1971). *On the economy of machinery and manufactures*. New York: Kelley.

Barbrook, R. (2002). The Regulation of Liberty: Free Speech, Free Trade and Free Gifts on the Internet. *Science as Culture*, 11.(2.), 150-177.

Bell, D. (1973). *The Coming of Post-Industrial Society: A Venture in Social Forecasting*. New York: Basic Books.

Berg, M. (1980). *The Machinery Question and the Making of Political Economy*. Cambridge: Cambridge University Press.

Braverman, H. (1974). *Labor and Monopoly Capital*. New York: Monthly Review Press.

Braverman, H. (1998a). *Labor and Monopoly Capital* (p. 318). New York: Monthly Review Press.

Braverman, H. (1998b). *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century* (p. 320). New York: Monthly Review Press.

Broersma, M. (2008). Proprietary Vendors Lose £30bn to Open Source. ZDNet.

Brown, G. (1977). *Sabotage – A study in Industrial Conflict*. Nottingham: Spokesman.

Coleman, G. (2008). Hacker practice: Moral Genres and the Cultural Articulation of Liberalism. *Anthropological Theory*, 8, 255-277.

Corporation, C. C. (s. f.). Licencia Creative Commons Reconocimiento-CompartirIgual 3.0. Accedido desde <http://ur1.ca/21mi5>

Dafermos, G., & Söderberg, J. (2009). The Hacker Movement as a Continuation of Labour Struggle. *Capital & Class*, 97, 53-73.

Edwards, P. (1996). *The Closed World: Computers and the Politics of Discourse in Cold War America*. Cambridge: MIT Press.

Gay, J. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston: Cop.

Hannemyr, G. (1999). Technology and Pleasure: Considering Hacking Constructive. *First Monday*, 4(2).

Hobsbawm, E. (1952). The Machine Breakers. *Past and Present*, 1, 57-70.

Jordan, T. (2002). *Activism!: Direct Action, Hacktivism and the Future of Society*. London: Reaktion Books.

Kay, J. (1832). Moral and Physical Conditions of the Operatives Employed in the Cotton Manufacture in Manchester.

Kraft, P. (1977). *Programmers and Managers: The Routinisation of Computer Programming in the United States*. New York: Springer-Verlag.

Lakhani, K., & Wolf, R. (2005). *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Projects in: Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K., Cusumano, M. & Shirky, Clay (eds.) Perspectives on Free and Open Source Software*. Cambridge: MIT Press.

Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. *Journal of Industrial Economics*, 50 (2), 197-234.

Marx, K. (1976). *Capital vol. I*. (pp. 554-555). London: Penguin Books.

McChesney, R., Wood, E., & Foster, J. (1998). *Capitalism and the Information Age: The Political Economy of the Global Communication Revolution*. New York: Monthly Review Press.

Moglen, E. (1999). Anarchism Triumphant: Free Software and the Death of Copyright. <http://ur1.ca/f6nx>; First Monday.

Moody, G. (2001). *Rebel Code: Linux and the Open Source Revolution* (p. 248). London: Penguin.

Noble, D. (1984). *Forces of Production – A Social History of Industrial Automation*. New York: Alfred A Knopf.

O’Riain, S. (2004). Net-Working for a Living: Irish Software Developers in the Global Workplace in: Ash, A. & Thrift, N. (Eds.). *Cultural Economy Reader*.

Polanyi, K. (2001). *The Great Transformation: The Political and Economic Origins of Our Time*. Boston: Beacon Press.

Pollack, A. (1998). in (ed.) Robert McChesney, Ellen Meiksins Wood & John Bellamy Foster, *Capitalism and the Information Age – The Political Economy of the Global Communication Revolution*. New York: Monthly Review Press.

Riccardo, D. (1821). *Principles of Political Economy and Taxation*, 3rd edition.

Ross, A. (1991). *Strange Weather – Culture, Science, and Technology in the Age of Limits*. London: Verso.

Sale, K. (1995). *Rebels Against the Future – The Luddites and Their War on the Industrial Revolution, Lessons for the Computer Age*. Reading, Mass.: Addison-Wesley Publishing Company.

Shaiken, H. (1986). *Work Transformed: Automation and Labor in the Computer Age* (p. 272). Lexington, Mass.: Lexington Books.

Shell, B., & Dodge, J. (2002). *The Hacking of America: Who’s Doing it, Why, and How*. Westport, CT.: Quorum Books.

Shiller, D. (1999). *Digital Capitalism: Networking the Global Market System*. London: MIT Press.

Sussman, G., & Lent, J. (1998). *Global Productions – Labour in the Making of the 'Information Society'*. Cresskill: Hampton Press.

Terranova, T. (2000). Free Labour: Producing Culture for the Digital Economy. *Social Texts*, 18, 33-57.

Torvalds, L., & Diamond, D. (2001). *Just For Fun – The Story of an Accidental Revolutionary*. New York: HarperCollins Publisher.

Ure, A. (1835). *The Philosophy of manufactures: or, an exposition of the scientific, moral and commercial economy of the factory system of Great Britain*. London: Knight.

Webster, F. (2002). *Theories of the Information Society*. New York: Routledge.

Weed. (2008). The Dark Side of Cyberspace – Inside the Sweatshops of China's Computer Hardware Production. <http://url.ca/f6ny>.

Wood, S. (1982). *The Degradation of Work? Skill, Deskilling and the Labour Process*. London: Hutchinson.

Zimbalist, A. (1979). *in: (ed.) Zimbalist, A. A Case Studies of the Labor Process*. New York: Monthly Review Press.

Zizek, S. (2002). A Cyberspace Lenin: Why Not? *International Socialism Journal*, 95.

Zuboff, S. (1998). *In The Age Of The Smart Machine: The Future Of Work And Power*. New York: Basic Books.