

- En Buenos Aires, a los tres días del mes de noviembre de 2008, a la hora 18 y 5:

**Sra. Presentadora.**- Señoras y señores, y sobre todo, jóvenes presentes: bienvenidos a este Salón Auditorio de la Honorable Cámara de Diputados de la Nación.

Vamos a dar comienzo al acto de presentación del doctor Richard Stallman, físico graduado en la Universidad de Harvard y máximo referente del movimiento global del software libre, que dio origen al proyecto GNU. Es fundador de la Free Software Foundation.

Revolucionó el mundo de las nuevas tecnologías al impulsar la ética y la filosofía del software y los principios básicos de la libertad de expresión.

Stallman ha recibido el premio Grace Murray Hopper de la ACM, una beca de la Fundación MacArthur, el Premio Pionero de la Electronic Frontier Foundation y el Premio Takeda, por sus contribuciones a la mejora social y económica, así como numerosos doctorados *honoris causa* en universidades tales como la Universidad Nacional de Salta, en la Argentina; la Universidad Vrije de Bruselas, en Bélgica; la Universidad de Glasgow, en Escocia, y el Royal Institute of Technologoy, de Suecia, entre otras.

Son sus anfitriones los señores diputados Eduardo Macaluse, del bloque ARI, y José Manuel Córdoba, del Frente para la Victoria, autores de los proyectos de ley sobre licenciamiento de software para la administración pública que actualmente están siendo discutidos en la Comisión de Comunicación e Informática.

También está presente la señora diputada Nélide Belous, del bloque del ARI, integrante de la Comisión de Comunicaciones e Informática.

Presentamos asimismo a la licenciada Beatriz Busaniche, representante de la Fundación Vía Libre. A través de ella tenemos la presencia en nuestro país de este ilustre visitante que realmente nos enorgullece.

La Fundación es el referente a nivel nacional del movimiento de software libre. Es una organización civil sin fines de lucro radicada en Córdoba, Argentina, que desde el año 2000 trabaja para que en la incorporación de nuevas tecnologías a la vida cotidiana se preserven los derechos fundamentales de la ciudadanía.

La Fundación Vía Libre ha acompañado proyectos legislativos tendientes a fomentar el uso y la adopción del software libre por parte del sector público nacional, también como herramienta estratégica de respeto a la transparencia de los actos de gobierno.

Vamos a dar la palabra a la licenciada Busaniche para que pueda realizar las presentaciones correspondientes.

**Sra. Busaniche.**- Muchísimas gracias, y bienvenidos.

Sólo quiero hacer dos o tres comentarios muy breves, porque queremos dedicar todo el tiempo del que disponemos a escuchar a Richard.

Quiero agradecer profundamente a los legisladores, quienes en el brevísimo tiempo de una semana nos habilitaron la Cámara para traer a Richard a conversar aquí sobre la importancia de la ética y los principios filosóficos del software libre.

De manera que en primer lugar quisiera agradecer a los legisladores que hoy son nuestros anfitriones, y también a todas las personas que se han movido en esta semana, que ha sido una "semana de locos".

Les pedimos disculpas si algo no está saliendo bien. Hay una gran cantidad de voluntarios que han trabajado durante toda la semana para hacer posible esta

presentación, y quería entonces agradecer que nos hubieran abierto las puertas de esta casa. Es un signo político absolutamente trascendente que Richard Stallman esté hablando hoy en la Cámara de Diputados de la Nación.

Quiero agradecer asimismo a todos los voluntarios que han estado trabajando, a todos los que han corrido durante toda la semana para que esto pudiera ser concretado.

Y agradecer y recordar -como seguramente lo harán después los integrantes del panel- a una persona que si estuviera entre nosotros, estaría sentado en este panel. Me refiero al fallecido ingeniero Dragan que fue el primer político de esta casa que allá por el 2000, 2001 ingresó el primer proyecto de ley de uso de *software* libre en la administración pública. Haber hecho esto aquí es un justo homenaje a la visión que ha tenido de ser el primer impulsor de proyectos de ley de este tipo.

No queda más que dejarlos para que disfruten esta conferencia -que sé que va a ser así-, y agradecer a Richard porque en las pocas horas que tiene de paso en la Argentina se dedica a compartir con nosotros esta tarde. Muchas gracias. (*Aplausos.*)

Le cedo el uso de la palabra a la señora diputada Nélide Belous.

**Sra. Belous.-** En primera instancia quiero agradecer la presencia de todos y explicar por qué estamos hoy aquí.

Soy integrante del bloque SI ARI Tierra del Fuego, y para nosotros es sumamente importante poder discutir este tema del *software* libre porque consideramos que en que aquí está puesta la soberanía de nuestro país y la transparencia.

El señor diputado Macaluse -presidente de nuestro bloque- presentó a la comisión a la que yo pertenezco el

proyecto de régimen de incorporación y de desarrollo de programas informáticos en el sector público nacional.

A partir de ahí encontramos otro proyecto que fue presentado por el señor diputado José Manuel Córdoba. La comisión viene discutiendo desde hace dos meses este tema, y durante toda la semana se ha reunido y ha discutido esto para sacar un proyecto consensuado y comprometido para que esto del *software* libre pueda ser concretizado.

Me gustaría hacer mención a que dentro de la comisión se ha invitado a diferentes organizaciones para poder escuchar su palabra y poder incorporar su pensamiento dentro de lo que luego será el proyecto de ley. Dentro de los grupos que han participado estuvieron presentes representantes de la industria del Software CESSI y CICOMRA, los militantes del *Software* libre, Fundación *Software* libre, los profesores de la informática, asociaciones de usuarios, Usuaría; y por parte del Estado, el OMRI.

Realmente creemos que es una buena oportunidad para que los diputados presenten sus proyectos y para que el señor Stallman -con su vasta experiencia- pueda hacer aportes y críticas a estos proyectos para que desde esta Cámara saquemos algo consensuado y lo mejor posible.

Desde nuestro bloque consideramos que el *software* libre es en el Estado una cuestión estratégica porque es sinónimo de soberanía y a eso apunta este proyecto.

Por eso necesitamos también de la colaboración de ustedes y de sus aportes. (*Aplausos.*)

Los dejo con la palabra del señor diputado Córdoba para que nos cuente en que consta su proyecto.

**Sr. Córdoba.**- Quería agradecer la presencia de todos ustedes en este día importante para nosotros y para el país, porque más allá de los gobiernos, nacionales, provinciales y municipales, las cuestiones que nos motivan

muchas veces están dirigidas a los más jóvenes y a la cuestión educativa en todos sus niveles.

En la provincia de Santa Cruz iniciamos este camino en el año 2005 para poder ser aplicado en la Legislatura y en algunas áreas, cosa que ha sucedido. Hay gente de la provincia que me acompaña aquí y que ha participado del desarrollo de este tipo de tareas a partir de esa inquietud que ustedes tienen y que permiten que la actividad que va promoviendo a través de esta industria - seguramente con la recuperación a través de la capacidad del conocimiento y la experiencia de muchos en nuestro país- va a tener como desarrollo posibilidades y eventualmente un trabajo para mucha gente. Esto es lo que nos proponemos desde nuestro lugar.

Simplemente quería decir esto, acompañar esta iniciativa, agradeciendo al señor diputado Macaluse y a toda la gente que ha posibilitado que en el consenso de la comisión tengamos este proyecto que seguramente va a tener un fuerte impulso a partir de hoy.

La posibilidad de tratar este proyecto va a satisfacer las ansias que tienen muchos de ustedes, como la de nuestros hijos y nuestros nietos. Desde aquí, lo que nosotros podemos hacer, es ayudar a que tengamos un resguardo para esto, terminar con este tipo de lobbys y con las cosas que nos han venido sucediendo durante mucho tiempo en nuestra historia, y que lamentablemente han tenido más costos que beneficios.

**Sr. Macaluse.-** Voy a ser muy breve porque todos hemos venido aquí a escuchar a Richard Stallman, pero quiero decir tres o cuatro cosas que me parecen importantes.

En primer lugar quiero hacer un reconocimiento a Marcelo Dragan que fue el primer diputado que presentó un proyecto abriendo un camino en esta Cámara. El falleció,

pero como decía la diputada Nélide Belous sé que estaría muy a gusto entre nosotros.

Con Marcelo no coincidíamos casi nunca en términos políticos. Era un hombre de lo que se llamaría rígidamente "de la derecha" y hemos sabido pelearnos arduamente. Pero como él era un liberal en términos genuinos planteaba este tema en el que nosotros coincidíamos.

Tampoco el señor diputado José Manuel Córdoba pertenece al mismo bloque de la señora diputada Nélide Belous y mío y muchas veces nos hemos enfrentado en discusiones y debates. Sin embargo, creemos que ésta debe ser una política de Estado y por eso estamos juntos aquí.

En tercer lugar estoy muy contento que el promedio de edad de los asistentes a esta reunión sea muy bajo y que la Cámara haya abierto las puertas para que entren jóvenes con mucha fuerza para escuchar este tipo de cosas. Cuando nosotros queramos impulsar este proyecto vamos a necesitar de la fuerza de ustedes, porque ya han pasado por la comisión grandes corporaciones de la informática que no vinieron justamente a apoyarnos.

Por último quiero agradecer la presencia del señor Richard Stallman, que le está robando horas al sueño y al descanso para estar hoy entre nosotros, pero entendíamos que era muy importante su presencia para que se pudiera conectar con ustedes en forma directa, y sobre todas las cosas, para participar de este debate.

¿Por qué damos este debate? Desde mi ignorancia en la ciencia informática entendemos que no es un tema meramente científico sino que es una cuestión filosófica, ideológica y cultural.

Para nosotros el conocimiento es estratégico y debe haber una justa distribución del mismo en nuestras sociedades. El conocimiento es una construcción colectiva, por lo tanto debe devolverse a la sociedad lo que ella le

dio a uno y hacerlo circular en forma gratuita, si es posible, retribuyendo lo que la sociedad nos ha dado a todos nosotros.

Siempre se dijo que aque l científico que puede vislumbrar un poco más adelante que el resto es porque está parado en los hombros de otros científicos y de muchos integrantes de la sociedad que le han permitido poder ver muchísimo más allá.

Por eso, Richard, estás parado sobre los hombros de muchos que te apoyamos, te alentamos y te queremos escuchar. Muchas gracias. *(Aplausos)*

**Sr. Stallman.**- Voy a hablar de pie para no dormirme.

Lo que dijo el diputado Macaluse no es correcto. No me están robando nada porque habría pasado noches sin dormir para poder hablar con ustedes aquí.

Es muy importante esta oportunidad y tengo la esperanza de aprovecharla bien, a pesar de tener sueño. *(Risas.)*

Nuestra sociedad lleva a la gente a buscar programas según criterios superficiales prácticos, tales como ¿es cómodo este programa? ¿es viable? ¿cuánto cuesta? Lo comprendo porque eso tiene importancia, pero la gente suele hacer caso omiso de otros criterios más profundos y más importantes tales como ¿cómo afectará mi libertad si uso este programa? ¿Cómo afectaré la solidaridad social de mi comunidad?

El movimiento de software libre enfoca estos asuntos más profundos. Un programa es software libre si respeta la libertad del usuario y la solidaridad social de su comunidad. De no ser libre un programa es software privativo porque priva de la libertad a sus usuarios y los mantiene en un estado de división e impotencia. División porque cada uno es prohibido de compartirlo con los demás e impotencia porque los usuarios no disponen del código fuente del programa. Por lo tanto, no pueden cambiarlo y ni

siquiera pueden averiguar lo que realmente les está haciendo, y puede hacerle cosas bastante malas.

El solo decir que estoy por la libertad no es decir mucho sino que es vago. Para realmente definir el software libre tengo que decir algo más específico. Un programa tiene software libre si tiene las cuatro libertades esenciales.

La "libertad cero" es la libertad de ejecutar el programa como quieras. La "libertad uno" es la libertad de estudiar el código fuente del programa y cambiarlo para que el programa haga lo que quieras. La "libertad dos" es la libertad de ayudar a tu prójimo, es decir, la libertad de hacer copias exactas del programa y distribuirlas a los demás cuando quieras, es decir, puedes regalar copias o venderlas, las dos opciones están permitidas.

Por último, la "libertad tres" es la libertad de contribuir a tu comunidad. Es decir, la libertad de hacer copias de tus versiones cambiadas y distribuirlas a los demás cuando quieras. Con estas cuatro libertades el programa es software libre, porque su sistema social de distribución de uso es un sistema ético, que respeta la libertad del usuario y la solidaridad social de su comunidad. Pero si falta alguna de estas libertades o es insuficiente, el programa es software privativo, es decir que su sistema social de distribución y de uso no es ético ni es justo.

Por lo tanto, cada programa privativo es un problema social, por su existencia misma y no debería existir. Diseñar un programa libre es una contribución a la sociedad más o menos según su utilidad, pero diseñar un programa privativo no es una contribución sino un golpe, un intento de acapararse de la sociedad, de los usuarios. El programa privativo funciona como una trampa al plan ético, atrayendo a usuarios a ceder su libertad y a traicionar a

sus comunidades. Si el programa tiene funcionalidades atractivas, ellas funcionan como el cebo de la trampa.

Por lo tanto, es mejor no diseñar nada que diseñar un programa privativo; se puede hacer cualquier otra cosa.

¿Por qué mencionar estas cuatro libertades esenciales? ¿Por qué definir el software libre así? Porque cada libertad tiene su razón.

En primer lugar voy a explicar la "libertad dos", que consiste en la libertad de ayudar a tu prójimo y de distribuir copias exactas cuando quieras. Se trata de una libertad esencial para poder vivir una vida ética, como buen miembro de tu comunidad. Si usas un programa sin la "libertad dos" estás en peligro de caer en un dilema moral.

Cuando tu amigo te pida una copia del programa, en ese momento tendrás que elegir entre dos males: un mal es darle una copia y romper la licencia del programa, y otro mal es negarle una copia y cumplir con la licencia del programa. Entonces, estando en el dilema deberías elegir el menor mal, que es darle una copia del programa a tu amigo. (*Risas y aplausos.*) ¿Por qué este mal es el menor? Porque cuando no puedes evitar de hacer mal a alguien es mejor hacerlo a alguien que lo merece. (*Risas y aplausos.*)

Podemos suponer que tu amigo es un buen amigo, un buen miembro de tu comunidad y normalmente merece tu cooperación. Mientras que, por contraste, el desarrollador del programa privativo ha bajado deliberadamente la solidaridad social de tu comunidad. Si es inevitable hacer mal a uno o a otro: hazlo al desarrollador. (*Risas.*)

Pero hacer el menor mal no es ser bueno. Nunca es bueno hacer un acuerdo y romperlo, aunque varios tipos de acuerdos son malévolos en sí y cumplir con ellos es peor que romperlos. Entonces, hacer el acuerdo y romperlo no es bueno. Si le das una copia, tu amigo tendrá una copia no

autorizada de un programa privativo. Eso es algo bastante malo, casi tan malo como una copia autorizada. *(Risas.)*

Por lo tanto, en este dilema lo que realmente debes hacer es evitar tener dilema. Conozco dos maneras de evitarlo: una es no tener amigos *(Risas.)*, que es la manera sugerida implícitamente por los desarrolladores de software privativo. *(Aplausos.)* Es decir, su meta es que cada uno esté aislado.

La otra manera es no tener software privativo, es decir, rechazarlo, y esa es mi manera de evitar el dilema. Si alguien me ofrece un programa atractivo pero que tenga la condición de no compartirlo con ustedes, lo rechazo, porque mi conciencia no me permite aceptar tales condiciones, dado que sería traicionar a mi comunidad. Por lo tanto, no lo acepto. Eso es lo que debes hacer también. Es decir, rechazar el software privativo al igual que la propaganda que los desarrolladores de software privativo usan para demonizar el compartir. Ellos utilizan términos como "pirata". Es decir, a los que ayudan a sus prójimos los llaman "piratas", equiparan ayudar a tu prójimo con atacar barcos. *(Risas.)* Nada podría ser más falso porque atacar barcos es muy malo pero ayudar a tu prójimo es correcto.

Por lo tanto, no lo llamemos pirata ni piratería. Cuando me preguntan qué pienso de la piratería digo que atacar barcos es muy malo. *(Risas.)*

Cuando me preguntan qué pienso de la piratería informática o musical les digo que yo sepa los piratas no usan ni computadoras ni la música para atacar barcos sino armas. *(Risas y aplausos.)*

Es la razón para la libertad 2: la libertad de ayudar a tu prójimo, la libertad de distribuir copias exactas del programa cuando quieras.

La libertad 0 es esencial por otro motivo, para tener el control de tu informática. Hay programas privativos que restringen, incluso, el uso de las copias autorizadas por la licencia. Restringen quiénes quieren usarlas, en qué computadoras, por cuánto tiempo o para qué propósito. Obviamente no debe haber control de tu informática.

Por lo tanto, la libertad 0 es esencial. Es tan obvia que durante años no pensaba necesitar mencionarla, pero luego me di cuenta de que sí. La libertad 0 es fundamental, pero no basta porque es la libertad de hacer o no lo que el código del programa permita. Es decir, que el desarrollador sigue decidiendo por ti no por la licencia, sino por el código del programa. Por lo tanto para tener el control de tu informática necesitas la libertad 1, la libertad de estudiar el código fuente del programa y cambiarlo para que haga lo que quiera. Así decides tú y no el desarrollador para ti.

Si usas un programa sin libertad 1, ni siquiera puedes averiguar lo que realmente te hace, y puede hacerte cosas malévolas. Por ejemplo, muchos programas privativos vigilan al usuario. Un programa privativo que vigila al usuario que quizás conoces se llama *Microsoft Windows*. (*Risas.*) Cuando el usuario de *Windows* -y no digo tú, porque no usarías tal programa malévolo- invoca la funcionalidad del menú para buscar una palabra en sus archivos, *Windows* envía un mensaje a *Microsoft* diciendo cuál fue la palabra usada, y cuando *Windows XP* pide una actualización por la red, envía a *Microsoft* la lista de todos los programas instalados en la máquina. Dos funcionalidades de vigilancia al usuario, pero *Microsoft* nunca anunció la presencia de esas dos funcionalidades malévolas, fueron descubiertas por la investigación de terceros.

Es decir, que es posible que otras funcionalidades de vigilancia queden por descubrirse dentro

de *Windows*, quién sabe, sólo *Microsoft*. Pero no sólo *Windows* vigila al usuario, también *Windows Media Player* hace vigilancia total. Envía a *Microsoft* todas las cosas que el usuario mira, pero no pienses que *Microsoft* siendo únicamente malévolos es el único que actuaría así, porque *Real Player* hace vigilancia igual. Según parece lo hizo primero. *Microsoft* es más conocido por imitar que por inventar. (*Risas.*) Y muchas empresas lo hacen, pero hay peores aún. Está la funcionalidad de no funcionar. (*Risas.*) Y no hablo de errores, es una funcionalidad deliberada, cuando el programa te dice: "No quiero mostrarte el contenido de este archivo aunque esté en tu computadora", "No quiero que copies una parte de este archivo, aunque esté en tu computadora", "No quiero imprimir este archivo para ti, porque no te amo". (*Risas y Aplausos.*) Pero realmente no es el programa quien no te ama, sino su desarrollador.

El desarrollador habrá desarrollado el programa para restringir al usuario. Es decir que el programa no está para servirte sino para ser tu guardia de prisión. Este tipo de funcionalidad se llama en inglés *Digital Restrictions Management* o DRM que en castellano sería Gestión Digital de Restricciones. Es la funcionalidad deliberada de no funcionar para ti.

Esto ataca tu libertad en dos niveles a la vez porque su objetivo es restringir el uso de tus obras publicadas. Privarte de lo que serían tus derechos legales es atacar tu libertad, pero el método empleado es publicar las obras en formatos encriptados, de manera que sólo con un programa privativo puedas acceder a tus copias. De cualquier manera, y usando un programa privativo, también pierdes tu libertad. Por eso digo que son dos ataques a tu libertad a la vez.

Por lo tanto debes rechazar todos los productos con DRM. Nunca debes aceptar un producto concebido para atacar tus libertades. Si no tienes personalmente las medidas para romper las esposas digitales no debes aceptar el producto.

Por ejemplo, los DVD tienen normalmente la película en formato encriptado para la gestión digital de restricciones. Si tienes una copia del programa libre capaz de leer la película del DVD, es aceptable comprar, alquilar o regalar DVD. Pero si no tienes el programa libre no debes aceptar ningún DVD encriptado.

Muchas empresas tienen este tipo de funcionalidad malevolente. Por ejemplo, Microsoft, Apple -que lo hace mucho- Google, Sony y Emerson y muchas más. Usualmente cada ejemplo de gestión digital de restricciones es montado por una conspiración de empresas y no por una sola.

¿Cómo funciona la gestión digital de restricciones en los DVD? Al comienzo un grupo de empresas diseñaron el formato encriptado y armaron su conspiración exigiendo que cada fabricante de lectores de DVD se una a la conspiración prometiendo mantener en secreto el formato, y que sus lectores restrinjan al usuario como todos los demás lectores. Sin participar en la conspiración no pueden recibir el secreto.

Personalmente pienso que tales conspiraciones de empresas para restringir el acceso del público a la tecnología deben ser delitos graves, castigados con prisión. Los jefes de esas empresas deben estar en prisión por lo que han hecho, pero no es así porque usualmente los Estados están al lado de las empresas y en contra de sus ciudadanos.

¿Cómo sé que existe esa conspiración? No soy investigador. Si fuera una conspiración secreta no sería yo quien pudiera descubrirlo. Pero no es secreta. Están tan

seguras del apoyo de los Estados que ni siquiera intentan esconder su conspiración.

Tienen un sitio Web, es decir que tienen orgullo de su conspiración en contra de nuestra libertad. Por eso, estoy consciente de su existencia, porque es un ataque organizado a nuestra libertad. Tenemos que contestar organizadamente.

La Fundación de Software Libre ha lanzado una campaña de protestas en contra de la Gestión Digital de Restricciones, se radica en el sitio web: [www.defectivebydesign.org](http://www.defectivebydesign.org). Hemos protestado ante eventos de *Microsoft*, de *Apple*, de bibliotecas públicas cuando distribuyen archivos con DRM, ante los cines, en cualquier lugar donde se apoyen las esposas digitales.

Pero existe algo peor aún, existe la funcionalidad malevolente de atacar al usuario. Es decir, las puertas traseras. Un programa privativo que contiene una puerta trasera y que quizás conocen se llama *Microsoft Windows*. (Risas.)

Con *Windows XP*, *Microsoft* comenzó a determinar más o menos la identidad del usuario. Por lo tanto, cuando *Windows XP* pide una actualización por la red, *Microsoft* puede entregarle una actualización específicamente para él. ¿Esto qué implica? *Microsoft* puede tomar el control de su computadora, tratarle como quiera y el usuario queda indefenso. Es la puerta trasera, cuya existencia podemos deducir de los hechos conocidos. ¿Hay otras? Es posible.

Hace unos años, en India, me informaron que habían arrestado a algunos de los diseñadores de *Windows* acusándolos de trabajar paralelamente para *Microsoft* y *Al Qaeda*, intentando introducir otra puerta trasera que *Microsoft* no debería conocer. Según parece aquel intento fracasó. ¿Hubo otros que se lograron? No podemos averiguarlo porque el código fuente es secreto, pero se sabe que *Microsoft* mismo ha introducido una puerta trasera para el uso de otra organización terrorista más violenta

aún, es decir, el gobierno de los Estados Unidos (*Risas y aplausos.*), específicamente para la Agencia de Seguridad Nacional y fue descubierta en el año 1999, inclusive antes del régimen de Bush.

Entonces, vemos que no es posible confiarse racionalmente de un programa privativo sin "libertad uno". Estos programas exigen una fe ciega del usuario, porque no hay otro modo posible de usarlos.

Con *Windows XP*, hay una manera de evitar actualizaciones malévolas que consiste en desactivar completamente la funcionalidad de pedir actualizaciones. Sabemos que si la funcionalidad está activa, *Microsoft* tiene la capacidad de introducir cambios de *software* sin pedir la aprobación del usuario. Pero, que sepamos, no puede hacerlo si la funcionalidad es desactivada totalmente. Puede ser peligroso desactivarlo y no recibir más actualizaciones, pero por lo menos es una defensa contra las actualizaciones malévolas.

Por lo tanto, con *Windows Vista*, *Microsoft* ha eliminado esa opción.

*Windows Vista* es un avance, sobre todo en la tecnología de restringir y atacar al usuario como nunca antes. *Microsoft* tiene el poder de imponer cambios de *software* en cualquier momento y el usuario no tiene la posibilidad de rechazarlas.

Es decir que *Microsoft* ha acaparado la máquina, pero no pienses que *Microsoft* es el único malévolo y que solamente *Microsoft* actuaría así. *Apple* ha hecho lo mismo con la versión X de *Mac Os*, dado que puede imponer cambios de *software* en cualquier momento.

Por lo tanto, decir que *Microsoft* y *Apple* pueden tomar el control completo de la máquina es insuficiente, porque realmente siempre tienen el control completo de la máquina. Este es el destino del *software* privativo, un

instrumento de control, incluso del poder del desarrollador sobre los usuarios.

Este tipo de puerta trasera, que otorga el poder total sobre la máquina al desarrollador también se encuentra frecuentemente en teléfonos. Los programas en los teléfonos móviles suelen ofrecer al operador la misma capacidad de imponer cambios de *software* en cualquier momento.

Entonces, debes rechazar el *software* sin "libertad uno", porque esa libertad es esencial. Pero no digo que todos los programas sin "libertad uno" contienen funcionalidades malevolentes. Unos desarrolladores los introducen y otros no. Supongo que hay bastantes programas sin "libertad uno", sin funcionalidades malevolentes, pero no podemos identificar ninguno, porque no hay manera de averiguarlo sin código fuente.

Entonces, podemos dividir los programas sin "libertad uno" en dos grandes clases: los programas en los cuales conocemos funcionalidades malevolentes y los programas en los cuales no conocemos ninguna. Dentro de esta clase hay programas que sí tienen funcionalidades malevolentes desconocidas y otros que no las tienen. No podemos saber cuáles son cuáles, pero podemos decir algo respecto de los programas que no tienen funcionalidades malevolentes, incluso sin poder identificarlos. Es que sus desarrolladores son humanos y, por lo tanto, cometen errores. El código de estos programas contiene errores y el usuario de un programa privativo sin "libertad uno" es igualmente impotente frente a un error accidental como frente a una funcionalidad malevolente deliberada. Si usas un programa sin "libertad uno" estás prisionero del programa que usas.

Nosotros, los desarrolladores de *software* libre también somos humanos y también cometemos errores. El código de nuestros programas libres también contiene

errores, como cualquier programa, pero si encuentras algún error en nuestro código o cualquier cosa que no te guste, tienes la libertad de cambiarlo porque no te hemos hecho prisionero. No podemos ser perfectos pero sí podemos respetar tu libertad.

Entonces, la "libertad uno" es esencial, pero no alcanza, porque es la libertad de estudiar y cambiar personalmente el código fuente del programa. Digo que no basta porque hay millones de usuarios de computadoras que no saben programar y no son capaces de ejercer directamente esta libertad.

Pero ni siquiera para programadores como yo basta la "libertad uno", porque hay tanto *software* en el mundo, tanto *software* libre en el mundo que nadie es capaz de estudiar personalmente todo el código fuente y hacer personalmente todos los cambios que desea.

Es demasiado para una persona. La única manera de tener completamente el control de nuestra informática es hacerlo juntos, colaborando. Para eso necesitamos la libertad 3: la libertad de contribuir a tu comunidad, la libertad de distribuir copias que tú puedas cambiar cuando quieras. Con esta libertad podemos colaborar.

Si alguno publica un programa libre y muchos quieren usarlo, pero le falta algo, desean algún progreso en el programa, uno puede comenzar con la versión disponible, hacer una mejora y publicar su versión. Otro puede tomar esa versión, hacer sus progresos y publicar su versión. Y otro puede comenzar aquí a hacer más progreso, publicar su versión, y luego tendremos el progreso que habíamos deseado y estaremos satisfechos. Creemos que ellos habrán colaborado para hacer este progreso empleando su libertad 3.

Entonces, la libertad 3 es esencial para tener el control de nuestra informática y los que no saben ejercer

las libertades 1 y 3 -porque no saben programar- pueden ejercerlas indirectamente. Por ejemplo, un usuario que desee un cambio y no sepa hacerlo puede buscar a un programador que desee hacerlo por él. Puede entregarle al programador la copia ejerciendo su libertad 2. Luego el programador puede hacer el cambio deseado ejerciendo su libertad 1 y entregar una copia de su versión cambiada al usuario ejerciendo su libertad 3. Y en seguida -cobardemente- nuestro usuario le pagará algo al programador.

Es muy útil -por ejemplo- para usuarios empresariales. Muchas empresas usan computadoras y usan *software* y usualmente no saben programar porque su negocio es otra cosa, pero pueden desear cambios en el *software* que usan. Con las cuatro libertades pueden conseguir los cambios que desean. Y también ofrecer una oportunidad para una industria local de *software*.

Entonces, las cuatro libertades son esenciales y todos los usuarios reciben los beneficios de ellas. Cualquier usuario puede ejercer las libertades 0 y 2, la libertad de ejecutar el programa como quiera y de distribuir copias exactas. Si no saben programar -sin embargo- pueden ejercer esas libertades.

Las libertades 1 y 3 exigen programar, por lo tanto, cada usuario puede ejercer estas libertades según sepa programar. Existen varios niveles de programación; es muy útil aprender a hacer cambios fáciles aunque no aprendas bastante para ser programador de profesión, como es muy útil aprender a hacer algunos mantenimientos de tu coche, incluso sin aprender bastante para ser mecánico profesional. Pero muchos no quieren aprender nada de la programación y no pueden ejercer directamente las libertades 1 y 3, pero cuando otros -los programadores- ejercen estas libertades y publican sus versiones cambiadas, todo el mundo puede instalarlas o no, como

quiera. Así recibe los beneficios de vivir en una sociedad donde todo el mundo tiene estas libertades.

Las cuatro libertades conjuntas nos proporcionan la democracia.

Un programa libre se desarrolla democráticamente bajo el control de sus usuarios. Cada usuario puede participar cuando quiera en la decisión social sobre el futuro del programa, que no es sino el conjunto de las decisiones individuales de qué hacer con el programa.

Por el contrario, el programa privativo, se desarrolla bajo la dictadura de su desarrollador. El desarrollador tiene el control completo del uso del programa, de su funcionamiento, etcétera. El programa privativo funciona como un instrumento de su poder. Su motivo es realmente tener el poder y si el programa hace algo útil es sólo para conseguir el poder sobre el usuario. Por eso frecuentemente hay funcionalidades malevolentes en el software privativo, y raramente en el software libre.

Cuando todo el mundo es libre, nadie tiene poder sobre nadie. Con el software libre uno puede interrogar su propia versión de un programa libre. Teóricamente también puede introducir una funcionalidad malevolente. Pero ¿qué pasaría si lo hace? Que otros pueden detectar la funcionalidad malevolente -porque pueden estudiar el código fuente-, luego publicar su versión mejorada que no tiene la funcionalidad malevolente, y decir: "¡Mira lo que he descubierto en este programa que se ha publicado! ¡Mira qué cosa mala!" De esta manera quien lo ha publicado perdería su reputación y fracasaría, y eso no es muy tentador.

El desarrollador de un programa privativo tiene el control completo del código y el poder sobre los usuarios. De manera que si decide introducir una funcionalidad malevolente nadie puede eliminarla. Pensemos que ya es bastante difícil sólo el hecho de detectarla. De

manera que piensa que puede ganar, y eso sí es muy tentador. Por eso lo hacen frecuentemente.

No hay ninguna defensa perfecta contra la maldad, pero la libertad y tener el control de nuestra informática es la mejor defensa conocida.

Por ello los invito a todos a vivir con nosotros en el mundo libre. El mundo libre es el nuevo continente que hemos construido en el ciberespacio para poder vivir en libertad.

Es imposible vivir en libertad en el viejo mundo del ciberespacio, donde cada programa tiene su señor feudal, sin una revolución.

Para vivir en libertad hemos construido un nuevo continente, y como es un continente virtual tiene espacio para todos. Allí no hay restricciones para la inmigración (*Risas y aplausos.*) Y como nunca hubo pueblos indígenas en el ciberespacio -que en sí mismo es bastante nuevo- tampoco se desterró a nadie y todo el mundo puede, ética y legítimamente, venir al mundo libre para vivir en libertad.

Que el viejo mundo desaparezca es nuestra meta. Queremos la liberación del ciberespacio y de todos sus habitantes porque, mereces la libertad, uses computadoras o no. Pero si usas computadoras, la única manera de vivir en libertad es rechazando el software privativo y venir al mundo libre.

.12Llegué a esas ideas, aunque no en su forma actual -no tan lindas y claras- en el año 83, es decir, quería poder vivir en libertad y seguir usando computadoras, pero me preguntaba cómo. Eso era imposible, porque la computadora no funciona sin sistema operativo y en el año 83, todos los sistemas operativos para las computadoras modernas de la época eran privativos. Por lo tanto, era imposible comprar

una PC y usarla en libertad. ¿Cómo podría cambiar esa situación?

Siendo una sola persona, sin mucha fama, sin talento político ni influencia especial no pensaba poder cambiarlo con un movimiento clásico de protestas de exigencia política, pero sabía hacer muy bien una cosa: desarrollar *software*, sobre todo el *software* del sistema operativo.

El problema existía porque todos los sistemas operativos eran privativos. Entonces, podía cambiar la situación diseñando otro sistema operativo y, siendo el autor, podría hacerlo libre y luego todo el mundo podría vivir en libertad con mi sistema operativo. Solo tendría que escribir un sistema operativo y podría corregir ese problema social con un trabajo técnico de mi propio campo. Es como si un trabajo para hacer hubiera caído en mis brazos, era consciente de un problema social grave y creciente, que la mayoría no reconocía como problema. Tenía la capacidad necesaria para intentar corregir el problema y parecía que nadie lo haría de no hacerlo yo.

Por lo tanto, había sido elegido por las circunstancias para hacer el trabajo y fue mi deber de ciudadano. Es como si ves a alguien ahogándose y sabes nadar y no hay nadie más -y que no sea Bush (*Risas y aplausos.*)-, tienes el deber moral de rescatarlo. Pero quizás he dicho algo demasiado fuerte. (*Risas.*) Quizás hay otros tiranos, los cuales no quiero nombrar, que tendrías el deber moral de rescatar. De todos modos, para mí es un asunto teórico, porque no sé nadar. (*Risas.*)

En este caso, la tarea que tenía que hacer no fue nadar sino diseñar un sistema operativo y esto sí lo sabía hacer. Por lo tanto, decidí diseñar un sistema operativo libre o morirme intentando, he de suponer, de vejez (*Risas.*), porque en esa época, el nuevo movimiento de *software* libre que estaba lanzando no tenía enemigos

activos, muchos no estaban de acuerdo pero solo se reían un momento y no prestaban más atención. Entonces, el obstáculo no era oposición sino un montón de programas que tendría que diseñar para tener todo un sistema operativo libre.

La tarea fue grande pero decidí hacerla y luego decidí reclutar a otros para participar y acabar el sistema más pronto. Luego decidí seguir el diseño básico de UNIX, para hacer un sistema portable capaz de funcionar en varios tipos de computadoras. Reconocía que el desarrollo duraría años y durante esos años se podrían cambiar las arquitecturas de computadoras. No quería diseñar un sistema y descubrir que ya estaba obsoleto, porque ya no fabricaban las computadoras en las cuales podría funcionar.

Entonces, decidí seguir el diseño de UNIX, que era un sistema operativo portable exitoso. Luego decidí hacerlo compatible con UNIX para que los usuarios de UNIX pudieran fácilmente migrar a mi sistema. Entonces, para comenzar solo necesitaba un nombre.

En los años 70, participaba en una comunidad de *software* libre, es decir, que entre los desarrolladores compartíamos el *software* que teníamos, incluso los programas que habíamos escrito.

En el laboratorio donde yo trabajaba, teníamos un sistema operativo libre y teníamos muchos desarrolladores que antojaban desarrollar *software* del sistema. La mitad éramos empleados y la otra mitad eran estudiantes, pero eso era secundario. El encanto de programar fue el motivo principal. Nos llamábamos *hackers*. ¿Eso que quiere decir realmente? Significa ser encantados por el uso de la inteligencia en un espíritu juguetón, con computadoras, pero incluso en otros medios uno puede ser *hacker*. No necesariamente tiene que ver con la seguridad, es un error de los periodistas, más o menos en el año 80.

De todos modos, programábamos por el placer de programar y para divertirnos más solíamos poner nombres graciosos a nuestros programas. En los años 70, la programación a nivel del sistema no era portable. Cada programa normalmente funcionaba solo en un tipo de computadora. Entonces, era bastante común ver un programa interesante y querer usarlo, pero no funcionaba en tu computadora. Entonces, el único remedio era diseñar otro programa parecido. Eso era normal en el campo informático en los años 70. Pero en nuestra comunidad, para divertirnos más, teníamos una costumbre graciosa para tales casos, que consistía en poner como nombre un acrónimo recursivo diciendo que tu programa no es el otro, es una manera divertida de reconocer al otro programa.

Por ejemplo, en el año 76 desarrollé el primer editor de textos EMACS. Después fue un editor extensible, y el usuario podía programarlo, es decir, cambiarlo, y todavía puede hacerlo. Luego, había treinta imitaciones de EMACS para otros tipos de computadoras. Algunos se llamaban algo EMACS, bastante obvio y no muy divertido. Pero también había FINE, para FINE IS NOT EMACS; SINE para SINE IS NOT EMACS; EINE para EINE IS NOT EMACS; y MINCE para MINCE IS NOT COMPLETE EMACS. (*Risas.*) La versión 2 de EINE se llamaba ZWEI para ZWEI WAS EINE INITIALLY.

Se pueden divertir mucho con los acrónimos recursivos. Busqué un acrónimo recursivo para decir ALGO IS NOT UNIX, es decir, "algo INU" pero ninguna combinación es una palabra en inglés. Entonces, qué hacer, sin otro significado no es divertido, no es una broma. Entonces, intenté eliminar la I, haciendo una contracción para tener "ALGO NU", es decir, SOMETHING IS NOT IUNIX. Intenté cada inicial ANU, BNU, CNU, DNU, ENU, FNU, ¡GNU!, que es la palabra más cargada de humor del idioma inglés. No pude resistir usar esa palabra. ¿Por qué está tan cargada de humor? Porque según el diccionario la "G" es muda y se

pronuncia "NU", como NEW, que significa nuevo en castellano. Entonces, cada vez que quieres escribir NEW -nuevo-, puedes escribirlo GNU y es un juego de palabras.

Entonces cada vez que quiere escribir *new* como nuevo, puedes escribirlo con *gnu* y es un juego de palabras, quizás no muy bueno, (*Risas.*) pero no me pude resistir. Cuando es el nombre de nuestro sistema, en inglés debes no seguir el diccionario porque si dices *The new sistem* puedes confundir a la gente porque hemos trabajado casi veinticinco años en su desarrollo y lo hemos usado más de dieciséis años, es decir, ya no es nuevo. (*Risas.*) Pero sigue siendo *gnu* y siempre será *gnu*, a pesar de los muchos que los llaman erróneamente *linux*.

En castellano es aceptable pronunciar el nombre como *new*, como si comenzara con Ñ que es el nombre del mismo animal y es más fácil de pronunciar para ustedes.

En español hay posibilidad de confusión, pero en inglés debe ser *gnu* y no *linux*. Pero ¿cómo comenzó ese error? Fue un accidente de la historia porque durante los años 80 nuestra tarea fue desarrollar todos los componentes para tener un sistema libre. En los años 90 faltaba un componente importante del sistema, el *cromo*, que es el componente que suministra los recursos de la máquina a todos los programas que ejecuta y la Fundación de *Software Libre* empleó en los años 90 a alguien para desarrollar un *cromo*.

Elegí un diseño que pensaba que nos proporcionaría el camino de acabarlo pronto, quizás me equivoqué, pero no sé por qué, porque no lo escribí yo; pero el desarrollo duraba aproximadamente seis años para tener la reacción de prueba y todavía no funciona muy bien. Es triste pero no es un desastre.

En el año 1991 un estudiante irlandés desarrolló su propio *cromo* y lo hizo funcionar a un nivel mínimo

dentro de un año. Ese *cromo* se llama *linuxy* que en sus comienzos no fue libre. Cuando salió la primera vez tenía una limitación no aceptable en contra de la distribución comercial.

Un programa para ser libre debe ser libre para cualquier tipo de usuario, en cualquier campo de la vida, no solo para las diversiones o en la escuela, sino también haciendo su negocio.

Entonces en 1991, *linux* no era libre, pero en 1992 el desarrollador cambió la licencia, adoptó la licencia pública general de GNU o GPL que es una de las licencias libres. Esta licencia que había escrito yo para usarla como la licencia de la mayoría de los programas del sistema *gnu*, pero la escribí de manera que todo el mundo pudiera usarla para sus programas, y el señor Torvalds la adoptó en el 92. Desde entonces fue libre y la combinación del sistema casi completo *gnu* y *linux* fue un sistema completo y libre.

La primera vez era posible comprar una PC y usarla en libertad con este sistema, básicamente el sistema *gnu* controla *linux*, pero los que combinaron *linux* con los componentes del sistema *gnu* no se dieron cuenta de lo que estaban haciendo y llamaron ese sistema como un sistema *linux* haciendo caso omiso de que la mayoría fue el sistema *gnu*.

Entonces así comenzó este error y es muy difícil de corregir, pero les pido que reconozcan nuestro gran trabajo, por lo menos el reconocimiento virtual porque hemos comenzado el trabajo, hemos descargado la porción del *software* de todos esos proyectos que han contribuido. Pienso que por eso, igual, merecemos reconocimiento. No pido más. Por lo tanto, por favor, cuando hablen del sistema, llámenlo GNU con LINUX.

Es verdad que el crédito no es tan importante como asunto ético, y si sólo fuera por el crédito no

mercería tanta atención. Pero hay otra cosa mucho más importante en juego en la elección del nombre: tu libertad; por cierto, en forma indirecta, porque en forma directa tu elección de nombres no cambia nada en la política, pero sí determina el mensaje que comunicas a los demás.

Con tu elección de palabras eliges qué decir. El nombre de GNU, desde hace veinticinco años ha sido asociado con las ideas de libertad que acabo de explicar aquí. Mientras el nombre LINUX está asociado a las ideas del desarrollador de LINUX, el señor Torvalds.

¿Y qué piensa él? No está de acuerdo con nuestras ideas de libertad y solidaridad social y ridiculiza públicamente la idea de que los usuarios merezcan libertad. Por supuesto que tiene derecho a sus opiniones, pero no es bueno que el gran trabajo que hemos hecho para tu libertad se atribuya a él y sirva para amplificar su voz y disminuir la nuestra.

Él quiere llevar a la gente a no valorar su propia libertad. Y es lo que sucede. Incluso en nuestra comunidad de usuarios y desarrolladores de software libre, muchos no valoran o no aprecian su libertad. Muchos, nunca han oído estas ideas, y hoy en día, para hacer llegar nuestras ideas al público, tenemos que superar el obstáculo de que la gente piense que el sistema es LINUX y que todo proviene de la visión ética del señor Torvalds que les dice que la libertad del usuario es absurda.

Eso es un peligro. La libertad, frecuentemente, está amenazada. Para mantenerla hay que defenderla. Para averiguar esto basta mirar a los Estados Unidos, donde nuestros propios líderes nos han quitado derechos humanos fundamentales en nombre de protegernos. Siempre dicen que quieren protegernos y por eso nos privan de libertad. Obviamente, nuestros líderes son nuestros enemigos principales. Los otros son nuestros enemigos secundarios.

Para defender la libertad hay que valorarla, y para valorarla hay que tener el concepto de libertad. En otros campos de la vida, el debate sobre los derechos humanos duró siglos. Se requirió mucho tiempo hasta llegar a conclusiones sobre los derechos humanos que todo el mundo merece y difundir mundialmente esas ideas. Desafortunadamente eso no garantiza que logremos defender la libertad.

En estos últimos años la libertad está perdiendo en casi todos los países, pero por lo menos se proporciona una base para intentarlo. Sin embargo, la informática es un campo nuevo de la vida. Sólo desde hace quince años, en algunos países avanzados, la mayoría de la población practica la informática.

Es poco tiempo para el debate, si hubiera un debate. Pero casi todo el mundo que usa computadoras, comenzó con el *software* privativo, rodeado por otros usuarios de *software* privativo y dio por supuesto que el *software* privativo es legítimo.

Por lo tanto, la sociedad dejó a los desarrolladores de *software* privativo dictar la respuesta a la pregunta de cuáles son los derechos humanos que el usuario de un programa merece y dictaron casi ninguno, como respuesta muy cómoda para ellos y mala para nosotros.

Entonces, el movimiento de *software* libre quiere iniciar un debate pero hoy en día para hacer llegar nuestras ideas de libertad a los usuarios -incluso de nuestros sistema GNU- tenemos que superar el obstáculo de que no se reconocen como usuarios del GNU. Ellos piensan que son usuarios de LINUX y tienen la tendencia de descartar lo que decimos, porque dicen "¡Ah! Esta idea del proyecto GNU; soy usuario de LINUX, no de GNU. ¿Por qué prestarles atención?" ¡Qué irónico! Si supieran que realmente el sistema que aprecian es GNU y existe por nuestro deseo de entregarles la posibilidad de la libertad.

Si supieran todo eso, quizás nos prestarían más atención y quizás tendríamos la posibilidad de convencerlos de defender su propia libertad, que es también tu libertad, y siendo más, lograríamos mucho en defensa de nuestra libertad.

Hoy en día, en muchos países ni siquiera oyen el término *software libre*, porque en los años 90 hubo una división en la comunidad de *software libre*, entre los que valoraban la libertad -como yo-, y los que sólo valoraban los criterios prácticos de comodidad, viabilidad, precio, etcétera. En el año 98 los demás inventaron otro término para utilizar, en vez de *software libre*, decían Open Source o código abierto, como manera de no decir "libre", porque no querían ni aludir a la idea de libertad.

Cuando miras lo que dicen en nombre del código abierto verás que no plantean los asuntos éticos de libertad o poder del desarrollador sino sólo ventajas prácticas. Sí es verdad que a veces el *software libre* tiene ventajas prácticas, porque vivir en libertad suele proporcionar ventajas prácticas. Pero eso es superficial. No debemos juzgar la libertad sólo según ventajas prácticas. Si gano más este año vendiendo mi libertad, ¿debo vender mi libertad? Seguro que no, pero suelen hacerlo.

Entonces, el código abierto es otra idea que no apoyo. Nunca apoyo las iniciativas en nombre del código abierto, porque quiero hacer llegar la idea de la libertad a más gente. Este hecho de que la libertad está frecuentemente en peligro no sólo es teórico y se puede ver en nuestro campo de *software libre*, en la comunidad de *software libre*. Tres veces hemos perdido la libertad que teníamos antes por no valorarla.

En el año 92, con la combinación del sistema GNU y kernel LINUX teníamos un sistema operativo libre, pero unos años

más tarde, en 95, ya había varias versiones del sistema, se llaman distribuciones, en competencia. Los desarrolladores de una distribución dijeron: "Podemos tener una ventaja comercial sobre las otras distribuciones introduciendo unos programas privativos que hacen cosas para las cuales no hay *software libre*". Eso atraería a usuarios y si ganaban así, ganaban más éxito.

Los desarrolladores de otras distribuciones decían: "Están ganando con sus programas privativos, tenemos que introducir los mismos programas privativos en nuestra distribución". Entonces, a través de los años ya no había ninguna distribución libre, todas las distribuciones del sistema GNU con LINUX contenían programas privativos.

Después de mis conferencias me preguntaron dónde puedo conseguir una copia, tuve que contestar: "No conozco ningún sitio que pueda recomendarle, porque todas las distribuciones contienen programas privativos y no puedo dirigir a la gente hacia ellas." No sería ético, porque no es ético recomendar el uso de un programa privativo a otros. ¡Qué triste! Durante un período teníamos la libertad y la perdimos por falta de atención a ella.

Estoy contento de decir que hoy en día conozco distribuciones completamente libres. Por ejemplo, UTUTO, que ha sido desarrollado en la Argentina (*Aplausos.*); BLAG, que quiere decir BLAG LINUX AND GNU (*Aplausos.*), y también está GNU SENSE. Esas distribuciones tienen la política de borrar cualquier cosa no libre que se descubra en el código, pero no son distribuciones muy conocidas.

Las distribuciones más conocidas y exitosas siguen teniendo programas no libres. Es decir que hemos comenzado a recuperar la libertad perdida pero sólo comenzado. Para llegar de nuevo a la situación de que el sistema GNU con LINUX siempre sea libre tenemos más camino que hacer.

Años más tarde perdimos la libertad de otra manera, porque el señor Torvalds implementó en LINUX la funcionalidad de carga dinámica de *drivers*. Y algunos fabricantes de dispositivos fabricaron *drivers* privativos. Esas empresas tienen una mala política, quieren vender un producto y no quieren decirte cómo usarlo. Su modo de empleo es secreto. En lugar de decirte cómo usar el producto que has comprado, te ofrecen un *driver* privativo, es decir, un programa privativo para usar el dispositivo. ¿Cómo podemos escribir programas libres para usar este dispositivo sin saber los comandos para usarlos?

Uno tiene que hacer una ingeniería inglesa para descubrir cómo. Pero los demás podemos ejercer nuestro poder de mercado rechazando la compra de los productos de modo de empleo secreto. Debemos hacerlo y siendo quizás cien millones tendríamos mucho poder en el mercado si lo usáramos. La mayoría no lo usa porque no reconoce el asunto, porque piensa que es código libre y que el único motivo está en sus ventajas prácticas. No tienen la idea de que merecen la libertad, no tienen la idea de defender su libertad, por lo tanto aceptan los *drivers* privativos. Podríamos ganar esta batalla fácilmente si lo intentáramos.

Entonces tenemos que educar a más usuarios para que valoren su libertad y la defiendan, porque nuestro futuro depende sobre todo de nuestros valores. Pero mientras tanto hemos perdido la libertad de otra manera porque hoy en día el gnu linux no es totalmente software libre. Su supuesto código fuente contiene listas de miles de números que realmente son programas binarios. Es decir, que realmente no son código fuente, pero fingen serlo. Los desarrolladores de *linux* han aceptado estos programas no libres, porque no vienen con código fuente, lo han puesto dentro de los supuestos código fuente de *linux*.

Entonces, hoy en día tenemos una versión libre de *linux*, es decir, el *cronox*. Se llama *linux* libre. Hemos borrado esos programas libres, por lo tanto, esos dispositivos no funcionan, porque estas porciones no libres no ejecutan en el procesador central sino están para cargar en otros procesadores en los dispositivos. El sistema debe copiarlo cada vez que enciende el dispositivo entonces exigen que hayan aceptado el programa no libre. Pero demuestra que tan fácil es perder la libertad si no la valoran.

Todavía necesitamos a más desarrolladores para que hagan más programas libres, pero necesitamos más y más gente para difundir las ideas de libertad porque nuestro futuro depende sobre todo de nuestros valores. Si valoramos la libertad podemos intentar defenderla, si no la valoramos vamos a perderla. Por lo tanto necesitamos tu ayuda.

Quiero tratar unos temas específicos. Uno es el *software* libre y el desarrollo. Muchos piensan fomentar el desarrollo con donaciones de computadoras. Hay quienes tienen la idea de que si todo el mundo practica informática seríamos un país desarrollado. No necesariamente es así, porque usar el *software* privativo no es desarrollo, es dependencia, es colonización informática. Usar un programa privativo es aceptar la colonización en forma digital. La colonización digital no se debe aceptar. Otros hablan de cerrar la brecha digital. Pienso que hay un error porque contiene la suposición de que todo uso de computadora necesariamente es bueno.

No estoy de acuerdo. Sólo es bueno si viene con libertad. Es decir que implantar más uso de computadoras con *software* libre sí es bueno, pero implantar computadoras con *software* privativo es retroceder en el plan ético de la libertad.

Usar *software* privativo es implantar una dependencia a empresas extranjeras. No es bueno para el futuro de ningún país y de ninguna sociedad.

Otro tema es el software libre y la economía. ¿Qué efectos tendría en la economía que todo el mundo migre al software libre?

El software privativo es un sistema muy efectivo para la concentración de la riqueza. Por lo tanto tiene efectos económicos malos. Eliminar el software privativo, por lo tanto, tiene efectos económicos buenos, porque facilita una industria local de software en cualquier región.

Con el software privativo, los usuarios, pagan a unas pocas empresas y no es posible competir con esas empresas localmente. Con el software libre, los usuarios, no pagan permisos para ejecutar el programa sino que pagan soporte, cambios, extensiones, adaptaciones, etcétera. En ese negocio, las empresas locales, tienen una ventaja. Es decir que con los clientes que habitan en Buenos Aires, los programadores de Buenos Aires, siempre tendrán una ventaja. No me refiero a una ventaja total ya que los programadores de otras ciudades y otros países tienen una oportunidad, pero por lo menos, en cada región, los desarrolladores locales tienen con unos clientes alguna ventaja. Con el software privativo no tienen ninguna ventaja con ningún cliente ya que tendrían que competir con los grandes.

Otra cuestión es el software libre y el empleo. Algunos dicen que si los usuarios migraran al software libre perderíamos todo el empleo del sector informático y eso sería un desastre para los informáticos.

Eso es absurdo. Aquí está el sector informático. La programación pagada es una pequeña fracción del sector. Pero en la programación pagada el desarrollo del software privativo es una pequeña fracción, porque la gran mayoría es desarrollo y programas a medida para el uso de un cliente.

Si los usuarios migran a un software libre perderíamos entonces esa pequeña fracción de una pequeña

fracción. Y esa es una buena pérdida. (*Risas.*) No sería un desastre. Incluso en el peor de los casos no ocasionaría un desastre en el empleo.

Además, el software libre también genera empleo de adaptación y extensión de software libre. Es decir que perderíamos algún empleo y ganaríamos algún empleo. No soy economista y no intento predecir si ganaríamos más o perderíamos más. Lo que puedo ver es que, incluso en el peor caso, no hay nada que temer.

Otro aspecto específico es el software libre en las agencias públicas.

Las agencias públicas deben usar únicamente el software libre. Tienen como cualquier usuario el derecho de tener el control de su informática.

Como cualquier usuario, merecen las cuatro libertades pero en este caso puedo decir más, es su deber mantener estas libertades, porque deben mantener su soberanía informática.

Una agencia pública tiene el deber de asegurar que las cosas se hagan bien, según los derechos de los ciudadanos. Es decir que no puede nunca dejar el control de su informática a manos privadas. Sin embargo, usar un programa privativo es dejar el control de su informática a las manos privadas del desarrollador del programa, y esto no se debe hacer nunca.

Las agencias públicas deben migrar al *software* libre. Como primera etapa deben rechazar la nueva instalación de programas privativos y luego eliminar los que ya tienen. Este proceso puede durar años, porque eliminar algo tan grande no se hace en una semana. La migración necesita tiempo, pero realmente para concretarla hay que dar pasos que no sean microscópicos.

También existe otro motivo en relación con la migración de las agencias públicas. Me refiero a la misión del Estado, que es organizar la sociedad para mantener la libertad y el bienestar de la población. En el campo

informático, un aspecto de mantener la libertad y bienestar es dirigir a la sociedad al *software* libre. Cuando las agencias públicas migran aumenta el mercado de soporte al *software* libre. Entonces, se crean empresas de soporte al *software* libre y cuando existen esas empresas, facilitan la migración de otros usuarios, como las empresas.

Entonces, hay dos niveles. Es importante que las agencias migren, pero la migración más importante que puede hacer el Estado no es de las agencias públicas sino de las escuelas públicas. (*Aplausos.*) Las escuelas deben enseñar únicamente *software* libre por cuatro razones. (*Aplausos.*)

La primera razón es para economizar, porque en cualquier país las escuelas no tienen suficiente dinero sino que están limitadas por la falta del mismo. Hay cosas que deberían hacer que no hacen porque no tienen dinero. Entonces, no deben desperdiciar su dinero pagando permiso de usar *software* privativo. Esta razón es superficial pero incluso los que no comprenden lo que realmente quiere decir el *software* libre comprenden esta razón. (*Risas.*)

Entonces cuando citamos esta razón, tenemos que decir que es secundaria, porque realmente es para la libertad. Hay empresas de *software* privativo que suelen eliminar esta razón regalando copias gratuitas o casi gratuitas de su *software* no libre a las escuelas. ¿Por qué lo hacen? Pienso que quieren emplear a las escuelas como instrumentos para imponer a la sociedad entera una dependencia a sus productos.

Su plan funciona así: dan unas copias casi gratuitas a la escuela, la escuela enseña el uso del producto a sus alumnos y esto crea una dependencia del alumno al producto. Entonces, los alumnos se gradúan con la dependencia y después de graduarse no reciben más las ofertas de copias gratuitas y menos aún las reciben las empresas para las cuales trabajan después de graduarse.

Entonces, el plan es que la escuela dirija a los alumnos camino a la dependencia, porque tienen consigo al resto de la sociedad. Las copias son gratuitas, es como tomar ampollas de droga adictivas en la escuela que se inyectan a los alumnos. (*Aplausos.*) Es decir que para crear una dependencia a la droga la primera dosis es gratis. (*Risas.*) Después de la dependencia tienes que pagar. Estas empresas tienen un departamento de "ser vicio" educativo.

Las escuelas rechazarían las dosis gratis de drogas y también deben rechazar las donaciones de *software* privativo, porque la escuela tiene la misión social de educar a la próxima generación como buenos ciudadanos de una sociedad capaz, fuerte, independiente, solidaria y libre. Esto sólo se logra enseñando el *software* libre. La escuela solamente debe enseñar el *software* libre, únicamente el *software* cuyo uso es el desarrollo, y no el *software* privativo.

Hay una razón más profunda, que se vincula con la formación de los mejores programadores. Hay algunos programadores natos, quienes a la edad de 13 años aproximadamente, quieren conocer todo el funcionamiento de la computadora y su sistema. Entonces, si usan un programa quieren saber cómo funciona.

Cuando el joven pregunta al profesor cómo funciona el programa y si es privativo, sólo pueden contestarle "No lo sé y no lo puedes aprender porque es secreto". Es decir que el *software* privativo no permite la educación sino que va en contra de su espíritu.

En cambio, si el programa es libre, el profesor puede explicárselo hasta que sepa y luego ofrecerle una copia del código fuente del programa, diciéndole "léelo y comprenderás todo" y lo leerá todo. Así aprende cómo programar bien, porque cada vez que encuentra algo que no comprende aprende que no se debe escribir así porque no es

claro. (Risas.) Si ni siquiera él es capaz de comprenderlo, realmente no es claro y está mal escrito.

A los programadores natos no hay que enseñarles a programar, leen el manual y dicen: "Ah, programar es esto. ¿Qué otra cosa podría ser?" Para ellos es natural, pero programar bien es otra cosa. Uno aprende a escribir bien el código cuando lee y escribe mucho código, y sólo puede leer mucho código con los son programas libres. Para escribir los códigos de los programas grandes, para aprender como principiantes, deben escribir pequeñas porciones de códigos para programas grandes, porque el resto de los programas grandes no aparecen en los programas pequeños. No se aprende escribiendo programas pequeños, tienen que escribir el código de programas grandes. Como principiantes no pueden escribir programas grandes desde cero, entonces tienen que practicar escribiendo cambios en programas grandes existentes, es decir, en los programas libres que utilizan.

Cada vez que el programa libre que usás no funciona a tu gusto, puedes cambiarlo. Y así se aprende, así aprendí yo. En la universidad, en MIT trabajando en laboratorio de Inteligencia Artificial porque teníamos un sistema operativo libre y mi trabajo era mejorar el sistema. Tenía que leer muchos programas y escribir cambios. Después de hacerlo cientos de veces, sabía hacerlo bien.

Hoy en día cualquier escuela puede ofrecer la misma oportunidad sólo si se piensa en una escuela de *software* libre. Pero hay una razón más profunda aún: para la educación moral, para la educación en la ciudadanía, porque la escuela debe enseñar no sólo hechos y métodos, por sobre todo el espíritu de buena voluntad, el hábito de ayudar a tu prójimo.

Por lo tanto, la escuela debe tener esta regla. Alumnos si traes un programa a la clase, no puedes guardarlo para ti, debes compartirlo con el resto de la

clase. (*Aplausos.*) Y debes mostrar el código fuente porque aquí es un lugar para difundir en el conocimiento.

La escuela debe seguir la misma regla para dar el ejemplo, es decir, que el profesor en la clase también debe compartir los programas con los alumnos. Las escuelas deben tener únicamente el *software* libre en la clase, todos los niveles de escuela, desde la maternal a la universidad. Todos deben rechazar el *software* privativo como enemigo del desarrollo social, de la educación, del espíritu de buena voluntad y de la sociedad. (*Aplausos.*)

Quiero mencionar unos sitios *web* para el sistema operativo *gnu* y en el movimiento de *software* libre podemos encontrar el sitio [www.gnu.org](http://www.gnu.org), para la Fundación de Software libre está el sitio [www.fsf.org](http://www.fsf.org) y en ese sitio puedes inscribirte como miembro de la fundación y encontrar nuestro directorio de *software* libre con más de cinco mil programas libres útiles. En [www.directv.org](http://www.directv.org). También puedes encontrar nuestras páginas de recursos, como por ejemplo recursos de *Hardware*, es decir, para muchas categorías tenemos la lista de los dispositivos que sí funcionan bien con el *software* libre sin *software* privativo. Son los productos que son buenos de comprar así puedes ejercer tu poder de mercado para presionar a los fabricantes que cooperen con nosotros publicando el modo de empleo de sus productos.

También [www.directdesigner.org](http://www.directdesigner.org), nuestra campaña de protesta en contra de las restricciones de las esposas digitales. También está la Fundación de Sitio Libre de América Latina, cuyo sitio se llama [www.fsfla.org](http://www.fsfla.org) y allí puedes participar en el movimiento. Aunque también hay organizaciones en el país que luchan por el *software* libre.

T.23

*Schiavoni*

*Paep*

Para finalizar voy a presentarles ahora mi otra identidad.

- El señor Stallman se disfraza de santo. (Risas y aplausos.)

**Sr. Stallman.**- Soy el San iGNUcio (*Risas*) de la iglesia de EMACS (*Risas*). Bendigo tu computadora hijo mío. (*Risas y aplausos.*)

EMACS fue al comienzo un programa editor de textos extensible que se volvió una manera de vida para muchos usuarios porque podían hacer todas sus tareas informáticas sin salir nunca de EMACS.

Por fin se volvió una iglesia, con el lanzamiento del grupo de noticias alt.religion.emacs.

Hoy en día en la iglesia de EMACS tenemos un cisma grande entre varias versiones rivales de EMACS, y también tenemos santos; pero no tenemos dioses. En lugar de dioses adoramos un editor. (*Risas*).

Para hacerte miembro de la iglesia de EMACS tienes que profesar la fe. Tienes que decir: "No hay otro sistema que GNU..." (*Risas.*) "...y Linux es uno de sus clones". (*Risas y aplausos.*)

También en la iglesia de EMACS, está la virgen de EMACS, que es una mujer que no ha aprendido a usar EMACS. (*Risas y aplausos*). Quitarle la virginidad de EMACS es un acto bendito en nuestra iglesia. (*Risas y aplausos.*)

Nuestra iglesia tiene algunas ventajas comparada con otras iglesias que no voy a mencionar. (*Risas.*) Por ejemplo el ser santo en la iglesia de EMACS no exige el celibato. (*Risas y aplausos.*) Pero sí exige vivir una vida pura y ética. Tienes que hacer

el exorcismo de todos los sistemas privativos que hayan poseído computadoras bajo tu control o armadas para tu uso, e instalar un sistema completamente santo y libre. *(Risas.)* Tienes que instalar únicamente software libre sobre el sistema.

Si haces este voto y lo cumples, también serás santo y tendrás derecho a llevar una aureola, si encuentras una porque no las fabrican más. *(Risas y aplausos prolongados.)*

A veces me preguntan si en la iglesia de EMACS el uso del otro editor VI es un pecado. *(Risas.)* Es verdad que VI, VI, VI es el número de la bestia. *(Risas.)* Pero el uso de una versión libre de VI no es un pecado; es una penitencia. *(Risas y aplausos.)*

A veces me preguntan si mi aureola es de verdad un viejo disco rígido de computadora. No es un disco duro; es mi aureola... *(Risas.)*... aunque sí era un disco rígido en una existencia anterior. *(Risas.)* Muchas gracias. *(Aplausos prolongados.)*

- El público canta a coro:  
"¡Richard, Richard!"

**Sr. Stallman.**- La libertad es más importante que yo. *(Risas.)* Deben decir: "¡Libertad, libertad!"

Prefiero que me formulen las preguntas por escrito, aunque otros pueden intentar preguntarme en voz alta para no desperdiciar el tiempo.

**Participante.**- Conozco MIT y para mí es una garantía en cuanto a la calidad del compañero profesor, quien me quitó varias preguntas en el transcurso de su exposición.

Yo estoy preparando un trabajo que se titularía "¿Revolución digital o colonización imperial?". Usted habló de valores no sólo de técnicas aunque trabaja en un área técnica, fundamentalmente de la revolución digital.

Me quedo con la palabra "ordenadora" más que con la PC -Personal Computer. Soy arquitecto pero los cómputos no los hago a través de la computadora, yo busco otra cuestión. Quizás se corre un riesgo muy serio con la traducción mecánica de Human Settlements, sobre asentamientos humanos, que es una tremenda "burrada". Eso fue un error de la primera reunión de Hábitat, de Vancouver.

Quiero hacerle una sola pregunta porque ya me contestó las otras. ¿Ha pensado algo sobre el concepto de sostenibilidad y sobre el concepto de velocidad de una revolución, que es una progresión geométrica, pero corre el riesgo de ser usada por el mercado en función del consumo? Es decir que usted se compra una computadora y a los seis días le dicen cómprese otra y otra. (*Aplausos.*)

**Sr. Stallman.**- Pienso que debemos pensar no como consumidores sino como ciudadanos. No sólo en términos de la comodidad de los productos que queremos consumir sino en qué tipo de vida queremos. También hay que notar que el *software* no se consume cuando uno lo usa. Entonces, decir consumidor de *software* es un poco absurdo.

**Participante.**- Pero hay una tendencia hacia eso.

**Sr. Stallman.**- Exacto. Y estoy en contra. Este movimiento está en contra.

**Sr. Participante.**- Yo también. (*Risas y aplausos.*)

**Sr. Stallman.**- En esta requisitoria escrita me preguntan qué pienso de la computación de NUBES. Es un término muy nebuloso (*Risas.*), que se aplica a muchos tipos de cosas. Centralmente pienso que es un intento comercial de cómo ver la idea de hacer tu informática con tus datos en el

servidor de otro, y así pierdes el control de tu informática. Entonces, no debes hacerlo. Es sencillo. No hay remedio, si haces tu informática en el servidor de otro, pierdes el control y que el *software* del servidor sea libre no es un remedio. Si bien el *software* del servidor es libre, es decir que el dueño del servidor tiene el control de su computadora y debe tenerlo, esto no te ayuda a ti. Entonces, debes hacer tu informática con tus datos.

Cuando haces tu informática con tus datos debes hacerla en tu computadora, con tu copia de un programa libre, pero ese término es muy nebuloso y hay muchos tipos de servidores WEB que hacen una variedad de cosas. No estoy en contra de todas ni de la mayoría. Hay muchos tipos de servidores en los cuales no hacés tu informática con tus datos y usualmente no son malos o a veces sí son malos por diversas razones o asuntos.

**Participante.-** Mi nombre es José, de FM La Tribu. Básicamente, para posibilitar el *software* libre te basaste en las legislaciones del derecho de autor y concebiste el concepto de copyleft.

**Sr. Stallman.-** No es verdad.

**Participante.-** La pregunta es si desarrollaste un método, el *copyleft*, para hacer todo lo contrario de lo que habitualmente las legislaciones de derechos de autor protegen. ¿Cómo sería un sistema de derechos de autor justo?

**Sr. Stallman.-** Pienso que el *software* y otras obras de uso funcional práctico en la vida deben ser libres. Por ejemplo, las obras de referencia, las de enseñanza, fuentes de caracteres para texto, etcétera, deben ser libres, porque son obras de uso práctico. También las recetas de

cocina, porque existen para uso práctico en la vida, aunque más o menos son libres, porque los cocineros las tratan de ese modo sin ser castigados.

Entonces, fuera de este campo, hay otros tipos de obras, como las de opinión, de testimonios, de arte y divertimento. No digo que deben ser libres sino que deben ser por lo menos, compartibles.

Cada uno que posea una copia de una obra publicada debe tener la libertad de distribuir copias exactas de la obra de una forma que no sea comercial, porque el compartir es una parte de la sociedad, y atacar el compartir es atacar la sociedad.

Entonces, tenemos que rechazar todos los intentos de dificultar el compartir, diciendo que compartir es bueno.

¿Qué tiene en el frasco? (*Risas.*) Es un jarabe anti tos. ¿Por qué no habla de BSD?

Básicamente hay dos sistemas libres como *gnu*, está *gnu and new*, y BSD. Con historias separadas aunque tuve algo que ver con el desarrollo de BSD como un sistema libre porque en el año 1984 o 1985, no recuerdo bien, fui a pedir a los desarrolladores de BSD que era una versión de *unix* y era privativo, separar su código del código de *unix* para poder usar su código como *software* libre porque deseaba poder usarlo en el sistema *gnu*. Después lo comenzaron a hacer, pero seguían reemplazando *unix* para hacer paralelamente. Por lo general hay estos dos sistemas y cada sistema tiene varias versiones. Pero que sepa yo ninguna de las distribuciones de BSD es completamente libre.

**Sra. Participante.**- Antes que nada quiero manifestar que a nivel personal coincido plenamente con todo lo que ha expresado, pero como soy periodista tengo que preguntar algo molesto.

**Sr. Stallman.**- ¿Es un deber preguntar algo molesto? (*Risas y aplausos.*)

**Sra. Participante.**- Yo llevo muchos años hablando con gente de la industria, especialmente con gente que tiene decisión de compra en informática y lo que me suelen decir es que si compran algo que lo hicieron unos cuantos chicos y lo desarrollaron solos, no saben si mañana seguirán existiendo. En cambio si compran algo de Microsoft, puede salir mal, pero nadie me puede echarlo por ello.

**Sr. Stallman.**- Es un tonto porque el soporte al *software* privativo suele ser muy malo porque es un monopolio, sólo *Microsoft* tiene la posibilidad de cambiar algo dentro de *Windows*, pero con el *software* libre el soporte es un mercado libre.

Es decir que esa empresa que piensa adquirir *software* podrá conseguir el soporte a través de un mercado libre, en el cual las empresas de soporte necesitan satisfacer a sus clientes. *Microsoft* no necesita satisfacer a sus clientes de soporte y suelen tratarlos muy mal. Dicen páguenos para contarnos el problema. (*Risas.*) Y si el usuario paga, le dicen que en seis meses habrá una actualización, cómprela y verá si hemos corregido su problema y verá qué problemas nuevos le proporcionaremos. (*Risas y aplausos.*)

Si realmente piensan que el mercado libre tiende a entregar mejores productos y servicios racionalmente deberían exigir comprar su soporte en un mercado libre porque es verdad que con el *software* libre o privativo el desarrollador puede desaparecer.

Con el software libre otros podrían retomar el desarrollo. La empresa puede pagar a alguien para hacerlo. Con el software privativo es imposible.

La próxima pregunta dice: "¿cree usted que los Estados que usan software privativos para resguardar los datos de los ciudadanos están incumpliendo su obligación de protección de los datos de los ciudadanos?"

Puede ser. Y también están en peligro de perder el acceso a sus propios archivos por los cambios de formato.

Los desarrolladores de software privativo suelen cambiar el formato y abandonar las versiones anteriores o los productos que usaban. Luego de diez o veinte años resulta muy difícil acceder a los archivos del pasado.

El software libre también a veces implementa nuevos formatos pero el código del soporte del formato anterior siempre queda disponible. La agencia puede guardar el código de la versión para acceder a los archivos antiguos siempre y en caso de necesitarlo después de muchos años puede emplear a un programador para actualizar el programa para que funcione.

**Sr. Participante.-** En Argentina, entre varias empresas que trabajamos con software libre estamos fundando CADESOL que es la Cámara Argentina de Empresas de Software Libre. Quería saber si conoce alguna otra experiencia como ésta.

**Sr. Stallman.-** No mucho. No soy experto en ese campo comercial. Me parece que usted sabe más que yo en ese sentido.

La otra pregunta que tengo por escrito dice: "¿qué piensa sobre ReactOS?"

ReactOS es un intento de desarrollar un sistema más o menos libre, compatible con Windows, pero la parte libre del software no funciona sola. Por lo tanto no podría

recomendarlo. Además, ¿quién quiere un sistema parecido a Windows? (*Risas y aplausos.*)

Teóricamente reemplazar Windows por un sistema libre es un buen proyecto. Es ético si se hace bien. Es decir si se hace éticamente.

Otra de las preguntas dice: ¿qué opinas del proyecto OLPC One Laptop Per Child?

Hace unos años, cuando Negroponte lanzó ese proyecto desconfiaba porque él no apoyaba de ninguna manera la libertad. Sin embargo parecía desear instalar un sistema que funcionara con software libre, y aunque yo siempre preveía que nos traicionaría, no lo hizo. Me sorprendió.

Las OLPC estaban saliendo y con el sistema GNU con Linux decidí usar uno para poder escapar del bios privativo que tienen todas las PC portátiles.

Hay un BIOS libre.

El obstáculo es recibir bastantes datos sobre un modelo de computadora para hacerlo funcionar en ese modelo. Hay modelos de computadoras para escritorios y de servidor en los cuales la BIOS libre funciona bien, pero nunca hubo un mapa PC sobre el cual teníamos bastantes datos. Entonces, el OLPC fue la primera máquina portátil disponible para usar un BIOS libre.

De todos modos, decidí migrar a OLPC. Luego, Negroponte anunció que el OLPC sería también una plataforma para *Windows* y algunos del proyecto dimitieron como protesta y es muy triste, porque ha decidido aplicar las máquinas de manera que sea posible instalar *Windows*, introduciendo una pequeña plaqueta, como de cámara.

Supongo que *Microsoft* distribuirá muchas tarjetas para poder convertir las máquinas de los niños a *Windows*. Entonces, es mejor comprar máquinas en las cuales *Windows* no sabe funcionar.

Por esto, en julio abandoné el OLPC, porque en China encontré gente de una empresa que fabrica

computadoras, que no son PC, tienen un proceso de Word, de tipo más o menos NIPS, y Windows nunca ha funcionado en ese tipo de procesador. (Risitas.) Esa empresa está diseñando su pequeña computadora para venderla en forma económica. Si compran ese tipo de computadora y las distribuyen, estarán seguros de que después no coloquen *Windows*.

Pienso que entregar computadoras a los niños con *software* concebido para provocar una dependencia no es tratarlos bien.

**Sr. Participante.-** ¿Cuál es la empresa?

**Sr. Stallman.-** Se llama LEMOTE. El producto todavía no está listo. Uno está esperándome pero no han empezado a producir cantidades masivas.

**Sr. Participante.-** Con respecto a los Estados que usan *software* privativo, más allá del tema de los formatos que usan, ya explicó que hay un tema de seguridad de que el dueño del *software* privativo tome el control de la máquina, pero ¿no estaría incumpliendo su obligación de guardar los datos privados de los ciudadanos?

**Sr. Stallman.-** Puede ser, porque existe la posibilidad de que el desarrollador del programa tenga una puerta trasera para el acceso, pero también es posible que con una *firewall* se evite.

Hay varias maneras de usar un poder injusto y todas son posibles. ¿Piensa que las redes libres pueden proveer Internet? No estoy completamente seguro de lo que quiere decir una red libre, porque no es el mismo significado de "libre" como en el caso del *software* libre, es otro asunto.

No pienso que ser proveedor comercial de Internet necesariamente sea malo, pero el peligro que veo viene más

de los Estados, es un peligro de vigilancia total y censura. Es la tendencia actual. Australia está proponiendo un sistema de censura tipo China.

**Sr. Participante.**- Soy Claudio Regis, del programa Dominio Digital.

Aprovechando el lugar donde estamos, quiero conocer su opinión sobre el voto electrónico. ¿Está a favor o en contra de que se vote con computadoras?

**Sr. Stallman.**- Es peligroso. Usar computadoras para la votación es un riesgo absurdo, porque una computadora permite cambios de programas fácilmente. Parece que Bush ha robado su segunda elección con computadoras en el Estado de Ohio. En algunos condados donde usaban computadoras para contar los votos existía una discrepancia entre los totales oficiales y los puntajes de salida. Sin embargo, en otros condados donde no usaban computadoras no existía tal discrepancia. Entonces, parecía que con la utilización de las computadoras robaron la elección, también robó la primera elección en el año 2000, negando el voto a cerca de 50 mil votantes en Florida, pero es otra cosa, no fue un problema de la informática.

La votación con computadoras no es de confiar. Es muy fácil hacer algo para aprobar la elección. En Brasil también hay sospechas de fraude. Un destacado profesor ha escrito artículos sobre cómo la comisión electoral brasileña ha eliminado toda la supervisión del proceso de votación y toda manera de contar de nuevo. No sé si hubo fraude pero ahora hay un camino abierto al fraude. También en algunos estados de los Estados Unidos.

En India usan máquinas electrónicas para la votación, pero no son computadoras, no son programables, son muy sencillas y no hay manera de cambiarlas, excepto cortando alambres. Eso es más seguro y quizás es aceptable.

**Sr. Participante.**- La primera pregunta no está destinada a Richard Stallman sino que tiene la intención de conocer la posición de la mesa en este sentido: ¿están dispuestos los diputados de la mesa a que un *software* libre, como Ututo, a través de la ley, sea instalado tanto en las escuelas como en la administración pública? ¿Cuál es la posición de ustedes, como políticos, en relación con la posición de Richard Stallman?

Necesito hacer una pregunta a Richard Stallman que es la siguiente. Uno de los mayores lugares de producción de *software* privativo es la industria bancaria y muchas herramientas financieras surgen a partir de la posibilidad de la producción de estas herramientas de *software* bancario. Una de estas herramientas da la posibilidad, por ejemplo, a la generación de los derivados, que fue la consecuencia de la actual crisis mundial.

Es decir que, intrínsecamente, la producción de *software* privativo está relacionada con los avances que se producen y lo que se reproduce sobre personas como nosotros, porque la crisis va a afectarnos fuertemente.

Quiero saber si tiene idea de los alcances de lo que usted está planteando. Además, la mesa al principio lo definió como filósofo: ¿usted se define como un filósofo, un político o un activista?

**Sr. Stallman.**- Un activista.

Ellos deben poder contestar primero, porque para ellos fue la primera pregunta. (*Risas.*)

**Sr. Macaluse.**- No quiero hablar en nombre de los demás. Comparto las posiciones que ha expresado Richard hoy aquí. Soy uno de los autores de uno de los proyectos que se están tratando para el *software* libre en la Argentina.

Creo que sería bueno que en las escuelas también se pudiera comenzar, es parte de una batalla cultural. No

es sólo un problema económico o político, como expliqué al comienzo.

No quiero explayarme más porque todos queremos escuchar a Richard y quiero que opinen los otros dos compañeros.

**Sr. Córdoba.**- Yo dije al principio que al menos en la parte que nos tocó a nosotros -y habló en plural porque nos tocó a mucha gente- ya comenzamos con la tarea.

Acá hay un profesor con cuatro alumnos de mi pueblo que estuvieron aquí en una olimpiada y se quedaron para compartir este momento, pero en la provincia quedó mucha gente que nos ayudó desde un principio. O sea que esto ya se estaba haciendo. No es si queremos o si lo vamos a hacer. Vamos a seguir ayudando a lo que mucha gente ya hizo.

Recién les comentaba a mis colegas qué bueno sería si pudiéramos tener este tipo de trabajo en algunos lugares más del país. Seguramente no habría ninguna cosa más que hacer que seguir ayudando a aquellos que -con esta presentación y con este esfuerzo- vienen a darnos su respaldo y seguir con la unificación de criterios que hubo entre nosotros -entre los diferentes partidos- para poder tomar la decisión de este proyecto.

Creo que la mejor cosa que podemos hacer en la vida es aprender a escuchar para de esa forma dar paso a todo lo que podamos hacer.

Yo voy a decir algo que mencioné en su momento pero que es la base del trabajo que he tenido durante muchos años y que inicié en mi provincia y ahora sigo acá en Nación. Tengo cinco hijos y seis nietos, mi hijo mayor tiene 30 años y el menor tiene 3. Los dos manejan perfectamente el tema informática; yo no sé encender una máquina.

**Sra. Belous.** Si bien no soy autora de ninguna de esos proyectos, sí acompañé el proyecto del diputado Macaluse porque realmente estoy convencida de que la única forma de desarticular los monopolios económicos que tenemos en nuestro país y fundamentalmente informático, es precisamente participando y apoyando al *software* libre.

Esto nos facilita y nos abre la puerta de la creatividad. Tener cosas enlatadas, resueltas por otras personas hace que simplemente nos remitamos a ser unos oprimidores de teclas. En cambio, si tenemos la posibilidad, y especialmente los jóvenes, de crear los *softwares* y manejarlos, indagarlos, modificarlos, es que se está en la posibilidad de crear una nueva generación de gente inquieta y realmente creativa. Por eso estoy de acuerdo con este proyecto, entre otras cosas. (*Aplausos.*)

**Sr. Macaluse.-** Antes de que conteste Richard, yo le quiero hacer una pregunta a ustedes. El día que se vengan las corporaciones informáticas a defender el *software* privativo y paguen publicidades en los grandes medios de comunicación y hablen mal de nosotros y hablen a favor de lo bueno que es el sistema privativo, ¿vamos a poder contar con ustedes?

**Sres. Asistentes.-** Sí. (*Aplausos.*)

**Sr. Stallman.-** Para contestar su pregunta tengo que explicar que mayormente cuando habla de los bancos y sus derivados y sus *softwares* para hacer comercio financiero, usualmente son programas de uso privado, no se publican.

Una empresa desarrolla su propio *software* para usarlo. Eso no es *software* privativo. *Software* privativo es cuando alguien te ofrece una copia del programa de manera de subyugarte.

Si desarrollás un programa y lo usás, generalmente no veo ningún mal en eso. Aunque sí podrías

hacer cosas malas, como siempre, con o sin computadoras. Eso es otro asunto.

Lo que sí critico son esos derivados financieros. Aunque pienso que el problema no es el uso mismo de las computadoras sino la falta de reglamentación. Y ¿por qué no hay reglamentación? Por la corrupción.

Los políticos comprados votaron en los Estados Unidos permitir derivados sin límites. El presidente Clinton intentó, en parte, vedar esa ley, pero hubo bastantes congresistas que la han votado a pesar de Clinton.

Entonces, fue la pura corrupción y con esa corrupción habrían podido llegar al mismo resultado sin computadoras, como en 1929. Se hizo igual. Sin reglamentaciones, porque nadie había pensado en la necesidad de reglamentar antes de la gran depresión.

Con el espíritu de especulación siempre fue posible usar menos de tu dinero con préstamos de otros. Así, tú y otros pierden muchísimo con una pequeña baja del mercado. Es lo que está sucediendo ahora. Es decir, es el mismo fenómeno. No es por causa de las computadoras. El problema es de corrupción porque el mercado totalmente libre es inestable. Tiene tendencia a cambios grandes y absurdos, incluso, que nadie desea.

Por eso hay que reglamentar el mercado financiero para evitar tales cosas. Saben hacerlo; lo que faltó fue la voluntad. (*Aplausos prolongados.*)

- Es la hora 20 y 38.