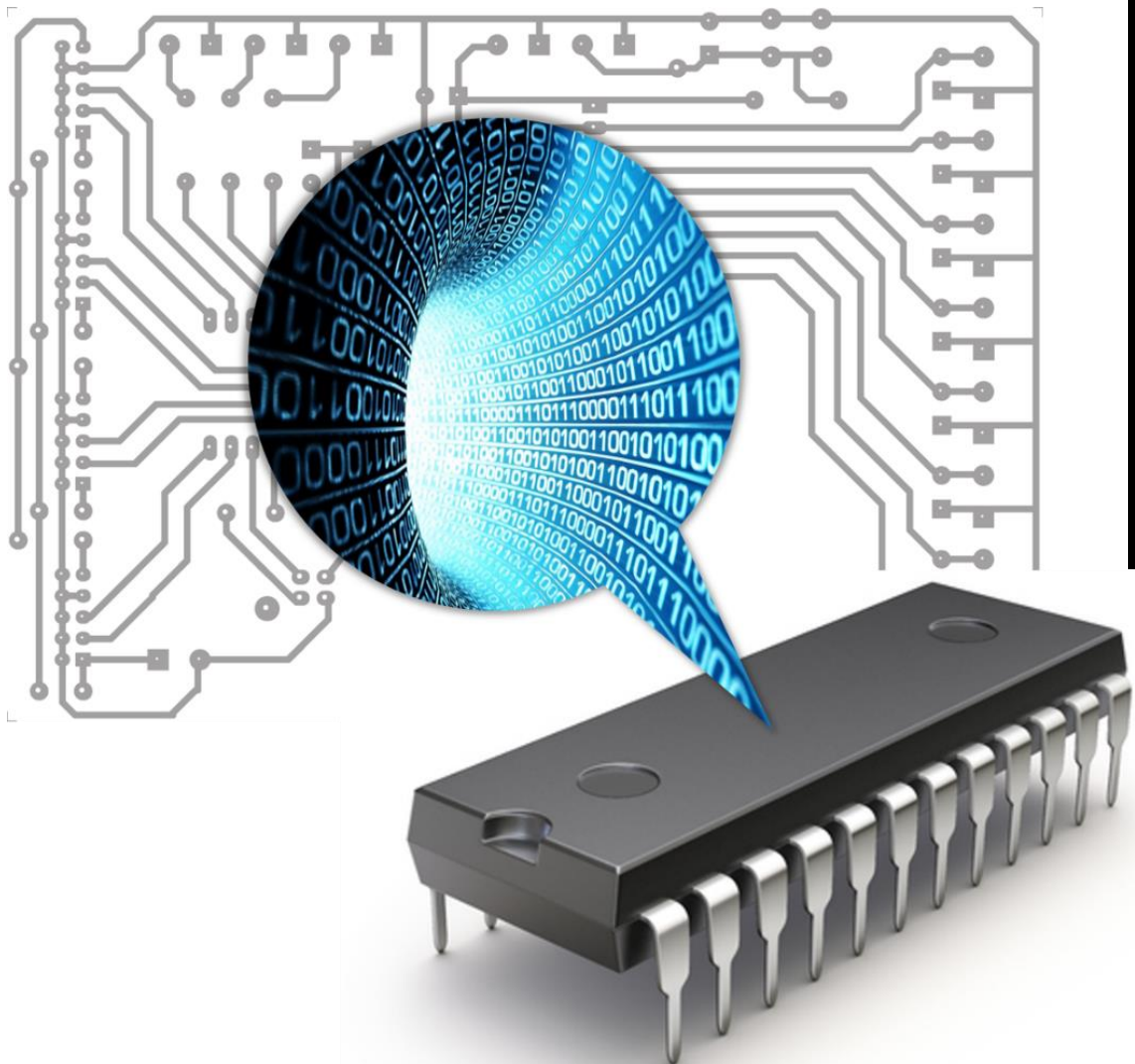


La Historia de un Byte

Dmitry Galuscenko




011100101010111ZEN

La Historia de un Byte

Dmitry Galuscenko

Traducción:
Vladimir Kameñar

 CelerSMS
Bogotá, Colombia

Tanto el autor, como el traductor y el editor de esta obra, no ofrecen garantía expresa o implícita de ningún tipo y no asumen responsabilidad por errores u omisiones involuntarias. Tampoco se asume responsabilidad por daños o perjuicios ocasionados por el uso de la información contenida en este libro.

No está permitida la reproducción total o parcial de esta obra sin autorización escrita de la Editorial.

CelerSMS S.A.S. es una editorial independiente.

ISNI 0000 0005 0265 593X

ISIL OCLC-COCEL

Ringgold ID 598732

Publons ID 18644

Para mayor información: info@celersms.com

Sitio web: www.celersms.com

Datos de catalogación bibliográfica

Galuscenko, Dmitry.
La historia de un byte, 1a ed.
CelerSMS, Bogotá, 2022
ISBN: 978-958-53602-2-8
Páginas: 16
004.4

© 2022, CelerSMS

Todos los derechos reservados

ISBN 978-958-53602-2-8

OCLC 1294014192

Fecha de publicación: 31 de enero de 2022

Traducido de: «*История одного байта*» (2001)

Contenido

<i>Prefacio</i>	i
El programador	1
El chip	4
Una segunda opinión	6
Monólogo	10
Las flores	12
<i>Acerca del autor</i>	16

Prefacio

«La impecabilidad no es otra cosa que el uso adecuado de la energía»

Carlos Castaneda, 1974

Hasta finales de los años 90, los microcontroladores de la época tenían recursos muy limitados. Los modelos más básicos tenían menos de 1Kb de RAM¹ y capacidades de ROM² o EPROM³ de 8Kb, 16Kb, etc. Para aprovechar de manera eficiente estos recursos la programación se hacía principalmente en lenguaje ensamblador. Cada byte era un tesoro que no se podía desperdiciar.

Hoy en día, hasta los modelos más económicos tienen decenas de kilobytes o megabytes de RAM. La capacidad de EEPROM⁴ es suficientemente amplia para albergar código compilado en C y lenguajes de más alto nivel. La optimización extrema ya no es fundamental, a diferencia de hace 20 años. Sin embargo, aún quedan áreas de aplicación, en las cuales cuentan cada byte y cada ciclo de reloj. Una de estas áreas es el microcódigo.⁵ Un solo byte de microcódigo puede causar grandes problemas y también resolverlos. El rendimiento de las CPU más poderosas depende directamente del microcódigo que tienen provisionado.

Algo similar sucede en el desarrollo de los núcleos de sistemas operativos. La microoptimización de código en estos núcleos busca obtener el mejor rendimiento posible, aprovechando los recursos de la máquina. Muchas veces los fabricantes de hardware (CPU, GPU, etc.) participan activamente en esta microoptimización para demostrar la superioridad de sus productos en comparación con la competencia.

¹ **RAM** (en inglés, *Random-Access Memory*) es una memoria de acceso aleatorio, generalmente utilizada para almacenar variables volátiles.

² **ROM** (en inglés, *Read-Only Memory*) es una memoria de solo lectura, generalmente usada para almacenar el código del programa.

³ **EPROM** (en inglés, *Erasable Programmable Read-Only Memory*) es una memoria ROM reprogramable un número limitado de veces, generalmente por medio de luz ultravioleta.

⁴ **EEPROM** o **E²PROM** (en inglés, *Electrically Erasable Programmable Read-Only Memory*) es una memoria reprogramable eléctricamente, antecesora de la tecnología **Flash**.

⁵ <https://es.wikipedia.org/wiki/Microc%C3%B3digo>

Otra área de optimización extrema son las producciones de Demoscene.⁶ Estas producciones se clasifican por categorías, según el tamaño de archivo en disco. Entre más pequeño el tamaño mayor posibilidad de aplicar en una categoría de peso más ligero, como en el boxeo. Los programadores de Demoscene buscan compactar sus producciones usando técnicas de compresión y empaquetamiento. Luego juegan con el formato del ejecutable para reducir su tamaño aún más. Cuando les sobran unos pocos bytes para lograr el tamaño reglamentario, entra en acción la microoptimización. Se invierten horas o días de trabajo para exprimir esos pocos bytes.

Por lo tanto, no es extraño que los programadores, especialmente en lenguaje ensamblador, tengamos cierta fascinación por la optimización del código. La mayoría de las restricciones físicas que hacían necesaria esta optimización han quedado en el pasado. El deseo de ahorrar hasta el último byte se ha convertido en una filosofía o un deporte. Por ejemplo, el movimiento de la programación *Zen* se inspiró en esta misma historia, publicada originalmente el 7 de mayo de 2001.⁷ Si omitimos la parte audiovisual en Demoscene, dejando solamente el código, el resultado podría ser un ejemplo perfecto de programación *Zen*.

La microoptimización tiene parecido con la micropintura y nanoescultura. Esta última exhibe figuras en el ojal de una aguja, entre otros espacios diminutos. Para el artista o escultor resulta extremadamente satisfactorio culminar con éxito estas obras únicas, aprovechando al máximo el espacio disponible. La microoptimización, la lucha por el último byte, es un desafío intelectual que produce una satisfacción o una frustración similares.

Vladimir Kameñar
Bogotá, Colombia
Enero de 2022

⁶ <https://es.wikipedia.org/wiki/Demoscene>

⁷ <https://groups.google.com/g/fido7.ru.embedded/c/QGrENEJAoUE/m/RMcRFDb6aq8J>

El programador

Me hacía falta un byte. Solamente uno. Uno de esos que constan de ocho bits. ¿Cómo? No, no estoy loco, aunque creo haber estado muy cerca de la locura. Pero bueno, vayamos en orden.

Soy programador. Pertenezco a la casta de programadores de *micros*.⁸ ¿Sabes de qué se trata? Te lo explicaré, si tienes paciencia. No puedo omitir las especificidades, ya que de otra forma no comprenderás lo que sigue.

Programo micros o *SoC*.⁹ En términos generales, esto es cuando todo el computador está en un solo chip. La memoria del programa y los datos están separados y no interactúan entre sí. El programa no se ejecuta en la memoria volátil. La profundidad de la pila¹⁰ es limitada. Solo puedo contar con ocho niveles de anidamiento y no puedo alterar esta profundidad máxima. ¡No creas que esto es malo! Tengo abundancia de recursos: ¡hasta 128 bytes de memoria volátil! Esto es para todo: variables y demás. Ya lo imaginaste, ¿cierto? La memoria del programa también es amplia: hasta ocho kilobytes. Es bastante fácil de usar: primero debo activar por código el banco de memoria; luego ejecuto el procedimiento deseado en él; al terminar, no debo olvidar retornar a la dirección subsiguiente. Además, hay que tener en cuenta que dentro del banco solo puedo moverme por saltos y llamadas a subrutinas, y los saltos condicionales tienen alcance máximo de una sola página, es decir, 256 bytes.

Esto significa que, al codificar un salto condicional, luego de una comparación, si el destino del salto está a más de 256 bytes, éste es un salto al más allá. En el mejor de los casos el compilador te advierte que dar pasos tan amplios puede romper los pantalones. ¡Esto es tan sólo la punta del iceberg! Resulta que no existe una instrucción de resta. ¡En absoluto! Solamente existe la suma. La multiplicación y división son lujos imposibles.

⁸ Coloquialmente, microprocesadores o microcontroladores.

⁹ **SoC** (en inglés, *System on Chip*) es un único circuito integrado, en el cual se albergan todos o gran parte de los componentes de un computador.

¹⁰ La pila (en inglés, *stack*) es usada para almacenar información acerca de las subrutinas activas, como la dirección de retorno, por lo que el tamaño de la pila limita la capacidad de anidamiento en las llamadas a subrutinas.

Aun así, necesito implementar todas las operaciones aritméticas decimales, lo cual es muy entretenido. El código para estos chips se perfecciona hasta niveles inimaginables, especialmente cuando se trabaja cerca del límite de capacidad de memoria. El código fuente se reescribe muchas veces. Para mí, no es suficiente resolver el problema. ¡Lo tengo que hacer caber dentro de ese chip!

Existen límites para las variables, el tamaño del código completo, el tamaño de cada subrutina y el número de llamadas. El más mínimo descuido hace que la pila se desborde y uno termina en el lugar menos esperado. El compilador es de poca ayuda, porque no controla estos límites. ¡Aún falta más! Mi código funciona en tiempo real... Entonces, para cada subrutina y fragmento de código debo calcular el tiempo de ejecución en milisegundos. Mis módulos no pueden exceder un tiempo máximo de ejecución, porque al mismo tiempo debo escanear el teclado y actualizar la pantalla, no descuidar los sensores y emitir señales de control. Todo el código debe funcionar sin interferir con el escaneo. De lo contrario se puede perder una pulsación de teclado o producir un parpadeo desagradable de pantalla. Además, tengo una interfaz.

Es el típico RS232C¹¹ que tiene cualquier PC, el que llamamos puerto COM. No creas que se trata de un chip externo, de los que envían y reciben bytes de manera autónoma. Para ahorrar costos lo hago todo por código, manipulando el único bit del puerto. Manualmente emulo las transiciones de bits de inicio, parada y datos.

Resumiendo, si escribí el código que resuelve la tarea, pero el código no cabe en la memoria del chip, la tarea no está resuelta. Si creé muchas variables, las cuales no caben en la memoria volátil, la tarea no está resuelta.

Si todo está bien, pero las subrutinas son demasiado largas e interfieren con la operación en tiempo real, la tarea no está resuelta. Si las subrutinas son cortas, pero son muchas, entonces la pila se desborda y la tarea no está resuelta.

Un programador promedio, en estas condiciones, habría renunciado al día siguiente. En mi opinión, para trabajar en este ámbito se requieren nervios

¹¹ <https://es.wikipedia.org/wiki/RS-232>

de acero y una voluntad inquebrantable. Rara vez tenemos contacto con los programadores normales, porque no tenemos mucho de qué hablar. No es que seamos prepotentes o engreídos, no es eso. Simplemente no concebimos la programación sin límites de recursos, utilizando lenguajes de alto nivel. ¿No hay espacio en disco suficiente? Puedes comprar uno más grande, ¿cuál es el problema? ¿Necesitas más memoria RAM? También la puedes conseguir sin problema e instalarla en su ranura correspondiente. ¿El código es muy pesado? Eso no es problema mío, es culpa del compilador. Consigan una máquina con mayor capacidad. Vivimos en mundos diferentes. Espero haber aclarado el por qué.

Es como navegar en un hermoso crucero en el vasto océano. Puedes navegar hacia donde quieras. En cambio, intenta navegar entre los escollos finlandeses.¹² Sería como enhebrar una aguja en la oscuridad de la noche, usando un hilo negro.

Por supuesto, tenemos diferentes chips disponibles. Antes de iniciar el desarrollo, elegimos nuestro chip con mayor atención que a la pareja. Debemos adivinar con cafeomancia¹³ y otros medios improvisados, ¿será que este chip nos resultará suficiente? Generalmente, luego de hacer la elección no hay vuelta atrás.

Aclarado el contexto, continuaré el relato.

¹² Los escollos son arrecifes o islotes rocosos. En la navegación marítima se consideran obstáculos peligrosos.

¹³ Una práctica de adivinar el futuro analizando los residuos de la tasa de café.

El chip

El chip no me inspiró confianza desde el inicio: me pareció bastante apretado. Todo estaba al límite. No era posible contemplar los 20% de reserva reglamentarios para compensar posibles errores y ampliaciones. Sin embargo, los demás candidatos sobrepasaban por mucho los requerimientos, por lo que utilizarlos resultaba costoso e ineficiente. El ahorro de costos fue decisivo. Finalmente emití mi consentimiento, dejando por fuera a los poderosos últimos modelos de Intel y Motorola.

Todo iba bien al comienzo. Al cabo de un par de meses de trabajo los micros se llenaron de código, las pruebas funcionaron correctamente, los circuitos se diseñaron y validaron debidamente. Los engranajes comenzaron a girar, al comienzo lentamente, luego con mayor impulso, para poner en marcha la compleja maquinaria productiva. La voz de mi consciencia seguía preocupada, ya que todo se hizo muy a ras. Solo el tres, a lo sumo el cinco, por ciento de los recursos quedaban libres. Sabía muy bien que eso es demasiado poco.

El trabajo quedó bien hecho. Recibí mi justa recompensa en forma de satisfacción emocional por el trabajo realizado. Algunas partes del código eran mejorables, pero también hubo una que otra perla.

Entonces... Aquí es donde empieza la mejor parte. Faltó implementar una función muy importante para el dispositivo, la cual fue omitida en la etapa de diseño. No fue culpa de los diseñadores, sino del cliente, quien olvidó solicitar que se incluyera esta minucia. Era una sencillez absoluta, pero imprescindible para el producto final. Pues, olvidaron incluirla en el diseño. ¿A quién no le ha pasado? Debe de ser muy fácil agregarla, en comparación con las demás funcionalidades avanzadas, ¿no? Al menos esta vez no olvidaron incluir algún sensor. ¡Es solo código!

El karma de los diseñadores es entender lo que necesita el cliente, aunque para ello deban introducirse en su cráneo y filtrar todo su contenido.

Lamentarse no sirve de nada. Entonces, me encerré por dos semanas en casa, no sin antes advertir que no estaría disponible para nadie. Una pequeña alarma y un semáforo se encendieron en mi mente. ¡Ring! ¡Luz roja! El primer

banco de memoria del programa está agotado. Reempaquetamiento. Redistribución del código entre las páginas y bancos de memoria. ¡*Ring!* ¡Memoria volátil agotada! Revisión de funcionalidad de las subrutinas. Esta variable se elimina. También puedo prescindir de estas otras, si cambio un poco la lógica.

¡*Ring!* ¡Desbordamiento de pila! Tocando fondo. ¿¡Pero cómo!? ¿¡Acaso ya agoté todos los niveles de anidamiento!? ¡*Ring!* Saltos que exceden el alcance de la página. ¡*Ring!* Demoras incompatibles con los lineamientos de tiempo real. ¡Las subrutinas consumen demasiado tiempo! ¡*Ring!* Segundo banco de memoria agotado.

Este ciclo se repitió muchas, muchas veces. Cuidadosamente, byte por byte, estuve haciendo caber el código de esta infeliz característica nueva, reempaquetando y redistribuyendo continuamente el código por páginas y bancos, optimizando el tamaño del código, acomodando las subrutinas según los requerimientos de tiempo real, uso de memoria volátil, ocasionalmente reescribiendo todo desde cero. Tal vez, crees que ocho kilos de código máquina no es mucho. ¡Ja! Las micro-instrucciones son de uno o dos bytes. Esto no es Zilog¹⁴ con sus micro-instrucciones de tres bytes, ni mucho menos el ochenta de Intel.¹⁵ A veces, me visitaba un amigo para preguntar cómo iba, pero no estuve de humor para conversaciones mundanas.

Dentro de una semana y media tuve que admitir que las cosas estaban mal. Había ensayado más de media docena de soluciones, pero no lograba avanzar. ¡Me aprendí de memoria cada subrutina y cada byte! Todos los esfuerzos eran en vano. La obra se congeló... El programa no daba signos de vida porque me faltaba un byte. Todas las soluciones alternas conducían a lo mismo. ¡Solo un byte!

¹⁴ Hace referencia al *Zilog Z-80*, un microprocesador legendario de 8 bits. (N. del T.)

¹⁵ Posiblemente el microprocesador *Intel 8080*. (N. del T.)

Una segunda opinión

De rápido fui a ver al amigo y le pasé todas las versiones de código con sus diagramas de tiempos y mapas de memoria. Luego regresé a mi guarida. Dormir. Necesitaba dormir como fuera. Necesitaba despejar la mente. Necesitaba una idea nueva. Me encontraba agotado. Durante los días siguientes no hubo ningún avance. Estuve componiendo y descomponiendo el código, ensartándolo como perlas en un collar, empaquetándolo de distintas maneras, tratando de encontrar la forma ideal, en la que todas las piezas del rompecabezas encajaran perfectamente, sin dejar un solo espacio. No había espacio suficiente... Un byte... Intenté amasar y moldear el código como si fuera arcilla, hice de todo, ¡pero siempre me quedaba faltando un byte! Probablemente, un compositor experimenta algo similar al escribir una sinfonía, tratando de encontrar la nota o el sonido que armonice la obra. También el artista busca esa pincelada única que le da vida a su cuadro. Sin el toque final la obra es inerte y todo el trabajo es un desperdicio.

*
**

Un día de esos, medio dormido, salí a la calle. La nieve crujía en tono chillón bajo mis pasos. Todo a mi alrededor parecía gris y opaco. Ya no necesitaba computador ni el código impreso. Todo el programa estaba dentro de mí... ¿o por fuera de mí? El programa me hacía ojitos y brillaba con destellos multicolores, tomando formas tridimensionales extrañas, pero en cierto modo hermosas, ¿o acaso eran diseños vectorizados?

Los marcadores de tiempo, entrelazados sutilmente en este diseño hechizante, emitían chirridos cortos y agudos. Todo esto se trasladaba y giraba a través de un canal polimorfo. ¿Acaso era una especie de tubo? A medida que avanzaba a través del canal, siempre quedaba a ras de sus superficies cambiantes. Era como si un copiloto invisible dirigiera su movimiento con gran habilidad, ordenando giros al borde del choque, siguiendo un instinto incomprensible, evitando que cada borde saliente se topara con algún obstáculo. ¿Quién era el arriesgado capitán o copiloto? ¿¡Acaso no se da cuenta de que no va a pasar por allí!? ¡Pero no! Mágicamente, toda la estructura logra esquivar el obstáculo, en el momento

preciso, deslizándose con una gracia admirable. Luego esquiva otro obstáculo. Sigue uno más... De repente, el borde de la estructura colisiona con estruendo. Sus bordes se deforman y estallan: todo explota en millones de fragmentos...

¡Es la pila! La estructura maniobraba para no sobrepasar la profundidad de la pila. Entonces, así es como se visualiza gráficamente este proceso. Cuando se desborda la pila, toda la estructura se derrumba estrepitosamente.

Mientras caminaba, noté a una anciana solitaria sentada en un banco. Decidí tomar asiento también... Enfrente vi como una vendedora de flores se agitaba de frío. No me agradan las ancianas... En particular, ésta se ve bastante desagradable, casi vil. ¿Por qué me está mirando fijamente? ¿Quién es ella? ¿Por qué está sentada allí? Probablemente no tiene nada que hacer. Es de las que forman filas en las tiendas, obstruyendo el paso en las cajas, por lo que nunca me alcanza el tiempo para comprar lo que necesito. Luego acosan en el trolebús, exigiendo que se les ceda el puesto. Tal vez, se la pasa viendo telenovelas tontas, de las que están de moda. ¿Cómo era que se llamaba? Acerca de una tal María, «*Los ricos también lloran*» o algo así.

¿Quién sabe cuántos conocimientos y esfuerzo se necesitan para construir estos programas? ¿Cuántas noches sin dormir? Además, tienes que estudiar y leer todos los días, pero no precisamente novelas románticas.

¿¡Quién puede entender esto!? ¿Será que lo entienden estos novatos que se hacen llamar programadores? Ellos escriben a la ligera en *FoxPro*, *Clipper* y *Basic*.¹⁶ Se la pasan preguntando:

- ¿Para qué es este comando?
- ¿En dónde consigo el archivo de ayuda?
- ¿Cuándo traducirán el manual?

¿Y qué se puede decir acerca de sus obras de “software”? Son toneladas de bibliotecas y módulos contables. ¡Creaciones monumentales, pero todas iguales! Las únicas diferencias son el grado de mediocridad y la cantidad de bugs. Estos bugs se amontonan uno sobre otro.

¹⁶ Algunos de estos lenguajes de programación de alto nivel estuvieron de moda en los años 90 e inicios de los 2000, especialmente en el ámbito de aplicaciones de negocios y bases de datos. (N. del T.)

Deberían obligarlos a tallar el código en piedra, como los antiguos canteros, para que piensen un poco en lo que escriben. Además, están las ratas que escriben virus. Cometan maldades en busca de fama. ¡Puah!

No, no quiero permanecer sentado en este banco, en compañía de esta... ¿A dónde me dirigía? Ya lo recordé: iba a ver al amigo. Tal vez, me de algún consejo. ¡Maldito byte!

¿Por qué siento tanta agresividad? Todos se dedican a sus trabajos, no necesariamente menos importantes que el mío. En todos los oficios hay personas talentosas. Algo me está irritando demasiado.

El amigo abrió la puerta y se quedó mirándome en silencio:

- ¿Y bien?
- Cuéntame solo una cosa — dije rápidamente, al ingresar en el apartamento. — ¿Me puedes ayudar con ese byte? Te daré lo que sea a cambio. Me hace falta un único byte, ¿entiendes? ¡No cabe en el chip!

El amigo permaneció en silencio durante un rato.

- Estuve viendo tus códigos.
- ¿Y? ¿Encontraste algo?
- No — respondió en voz baja y agregó luego de una pausa — Es un tejido perfecto. No hay nada que se pueda mejorar. Cada línea de código merece sello de calidad. Este código se puede tallar en mármol. Es homogéneo, sin solapamientos ni vacíos. Es como un monolito, pero elástico. Pero...
- ¿Pero qué? ¡Dilo de una vez! ¡No me tortures más! — exploté desesperadamente.
- No logro asimilar la mayor parte del código... No lo entiendo... Solamente percibo su belleza intangible. Mi conciencia no logra captar su significado. Parece que el sentido está allí, pero se me escapa en el último instante. Es como un copo de nieve que se desvanece en la mano, o el sonido de una música conocida, pero que no logras recordar.
- ¡¿De qué hablas?! ¡¿Nieve, música!?! ¿¡Puedes ayudarme o no!?! — comencé a gritar. — Eres mi amigo. Ayúdame. Solo dime en dónde consigo ese byte — lo miro con esperanza. — Dale una mirada fresca.

¡Puede estar sobrando un operador, solo uno! ¡Y problema resuelto!
Te lo agradeceré la vida entera... — susurré, agarrando su manga. —
Son muchísimas líneas de código. Pude haber cometido algún error,
como cualquier humano. Pude haber pasado por alto algún detalle...
Ya no necesito ni fama ni reconocimiento ni nada. Lo único que quiero
es dejar atrás esta tarea, desprenderme de este tormento. Ya no
tengo fuerzas para continuar. Vamos, échale otro vistazo a los
códigos...

- Desiste — dice él.
- ¿Desistir? No entiendo.
- Cambia de chip.
- ¡¿Estás loco?! ¿Cómo que cambie de chip? ¡Montones de circuitos impresos listos para usar, técnicos, salarios, partes! ¡No es broma! Esto no es como agregar disimuladamente un par de líneas de código en el computador, sin impactos ni sobrecostos. ¿Qué hago con la gente? ¡Ellos confían en mí! Les dije que sí. Me comprometí, aunque sabía que no tenía suficientes recursos de reserva. ¡Ellos ya tienen todo listo! Chasis, fierros, alimentación. Solo están esperando el código. ¿Sabes lo que implica cambiar de chip? ¡¿Sabes lo que terminaría costando ese byte?! — quedé aturdido por mi propio grito.
- Idiota — las palabras caen al piso como un fajo de ropa mojada. — ¡Desiste! ¡Olvídalo! Los muchachos y yo hemos estudiado tu código durante tres días. Reuní a todos los que pude. Tu problema es que NO tienes ningún error. ¡Ninguno! Ni siquiera entendemos cómo lograste compactarlo tanto.

Monólogo

Mis piernas se doblaron y terminé sentado o reclinado sobre una silla. En el fondo lo sabía. Entonces, comencé a hablar... Fue un monólogo extraño... Como si alguien gritara, susurrara y volviera a gritar con mi voz:

«... He estado pensando todo este tiempo. Me di cuenta de que el alcance de mi responsabilidad en este proyecto no es tan importante. Bueno, si no se soluciona, se tendrá que rehacer todo. ¡No será el fin del mundo! No es tan grave. Más allá de avergonzarme... El problema es otro, está dentro de mí. Sabes que hace tiempo dejé de ser un esclavo que cumple ordenes, justificándolo con la necesidad de ganar dinero para la familia. Hace tiempo me convertí en mercenario. Mis servicios, mis programas cuestan mucho. Sabes que nunca acepto trabajos simples, aunque me ofrezcan el cielo y la tierra. Trabajo por mí y para mí. ¡Hace tiempo dejó de importarme el dinero! Ya no tiene mayor poder sobre mí. ¡Es otra cosa lo que necesito! No lo admito abiertamente, pero estoy dispuesto a tomar tareas desafiantes de forma gratuita, hasta pagaría por hacerlo. Tú también lo sabes, ¿no? Sé que tampoco lo haces por dinero. Hay algo que nunca le he dicho a nadie. Cuando programo, vuelo... ¡No te hagas el que no sabe de qué se trata! Todos lo saben, aunque prefieren no recordar o no creer. ¡Tú también conoces esa sensación! Es como en los sueños de niños. ¿Recuerdas? Despegamos y nos elevamos muy alto, luego aceleramos y gritamos de emoción. ¡Debajo vemos pasar bosques, montañas y mares! ¿Crees que fue un sueño? ¡No! Hace tiempo lo comprendí, pero no me atrevía a admitirlo. Me sentía avergonzado, ¡pero ya no me importa! ¡Es el alma volando! ¡De veras! La mente nos hacía creer que estábamos durmiendo. ¿Sabes por qué perdemos esta habilidad cuando nos volvemos adultos? ¿Sabes por qué dejamos de tener este sueño? ¿No sabes? Es porque nuestra alma se vuelve pesada por las cosas terrenales, como el dinero o la condición social. Ya no es capaz de elevarse con esta carga. ¡Nos encanta tendernos como un tapete limpiapiés! Se nos ocurren muchas razones, realmente convincentes, para justificar nuestra humillación, pues lo hacemos por el bien de alguien más. Nos engañamos todos los días, convenciéndonos de que hacemos lo correcto. ¡No quiero, no puedo engañarme más! ¡¿Qué tiene que ver esto con mi byte?! Es muy simple. El

trabajo me permite volar nuevamente. ¡Si! En los momentos de mayor dificultad, al resolver los problemas, despego. ¡Es indescriptible! No puedo volar todo el tiempo. Desciendo... Hasta que un nuevo desafío de código me permite despegar una vez más. Invierto una parte de mí en el código. Finalmente, esta vez quedé atrapado. El orgullo fue la causa de mi perdición. Me acostumbré a ser el gurú que trata a todos como principiantes, resuelve fácilmente sus problemas de manera condescendiente. Me encanta que me vean como a un dios. Todos somos engreídos, también yo. Ahora todo cambió. ¿Crees que el problema es ese byte que no logro reducir? ¡No! No lo puedo resolver, pero sé que lo puede resolver **otro ser**, ¿entiendes? La solución existe, la percibo, pero no la puedo encontrar. Debo dejar de ser yo, convertirme en otro, para encontrarla. Alguien o algo logró atraparme con ese único byte. Sabes que soy astuto. Si el problema no tuviera solución, habría evadido esta trampa. Creí que si existe solución. ¡Por eso quedé enganchado! Quedé atrapado en este byte, como en una red. Este byte tiene el diámetro del orificio, por el cual no me puedo escabullir. Llegué demasiado lejos... ¡Por eso ya no me puedo ir y ser libre de nuevo, a menos que cambie algo dentro de mí! Debo cambiar por completo, ¿sabes? ¡Debo convertirme en otra persona! Entonces, tal vez, se abrirá una nueva puerta ante mí... No sé qué habrá detrás de esa puerta... No sé qué y cómo debo hacer para que esto suceda... Además, tendré que pagar un precio. También lo he comprendido. El precio es que no volveré a ser como antes y no podré volar más. Todo habrá terminado...»

Me levanté lentamente, agaché la cabeza y me dirigí a la puerta. Adiós — me despedí con total resignación.

¡Cretino! — escuché detrás.

Las flores

Morirás con este programa. ¡Morirás! ¡Mírate en el espejo! Terminaras en un manicomio. ¡Haz algo ya! Mujeres, trago, cualquier cosa...

Ya no lo puedo escuchar. Dios, si existes, ayúdame... En la calle me doy cuenta de que olvidé los guantes y el gorro. ¿Para qué los necesito? ¿Acaso importa? ¿Qué es lo que importa? ¿Para qué todo esto? ¿Quién lo necesita? ¿La gente? ¡A la gente no le importa! ¡Soy **yo** quien lo necesita! Me he conducido a mí mismo a una trampa y no puedo salir de ella. ¿Será que es un castigo? Es una lección de humildad para mi ego hipertrofiado. Un momento... ¿Qué fue lo que me dijo? ¿Mujeres?



La oficina es cálida y acogedora. Calefacción. Todo se ve muy limpio. Se escucha el zumbido de los computadores.¹⁷ Carpetas, repisas. Las chicas me ofrecen café con panecillos. Les unto chocolate y los devoro rápidamente.

Me sirven más y más. Ven a su colega caído con miradas compasivas. Las chicas también son programadoras, hacen programas de oficina para contabilidad y cosas afines. Están bien arregladas, maquilladas. Se ven realmente genial, especialmente en contraste conmigo y mi barba de varios días. Respondo brevemente y sin mucho entusiasmo sus preguntas acerca de los problemas que me hicieron caer tan bajo — «Ensamblador, problemas con la pila, falta un byte».

Una de ellas (creo que se llama Oksana) dice con acento agradable — «¿Cuál pila? ¿Para qué la necesitas? Olya y yo programamos en Clipper y FoxPro. No usamos ninguna pila. ¿No será que tú tampoco la necesitas? Ustedes, los hombres, siempre inventan algo para complicarse la vida con tonterías. Se atormentan ustedes mismos y luego nos atormentan a nosotras. ¿Cierto, Olya?»

¹⁷ Los computadores de la época producían un ruido característico, principalmente emitido por los ventiladores, discos duros y pantallas CRT. (N. del T.)

¡¿Cómo así?! — solo la silla evita que caiga al suelo. Mi voz se vuelve ronca y desagradable — ¿¡Ustedes programan **sin pila!**?

Lo que siguió fue un ataque de histeria...

*
**

Nuevamente estoy en la calle. Lindas chicas, son afortunadas al no tener estos problemas. Qué bueno que no lo sepan. No todos ni todas necesitan el “privilegio” de enloquecer resolviendo problemas filosóficos de programación. En los lenguajes de alto nivel hay muchas damas, pero hasta ensamblador llegan pocas. En el nivel más bajo, donde me desenvuelvo, no hay casi ninguna.

Los pasos me conducen en dirección desconocida. Los pensamientos fluyen solos. Ya no les pongo cuidado, estoy condenado. No podré vivir con esto. Me dedicaré a cualquier otra cosa, me mantendré a tiro de cañón de los computadores. ¡Bien merecido! Esto me pasa por creerme superior. Disque todos saben que lo puedo todo. ¡*Finita la commedia*, imbécil! Deambulo en este mundo extraño, en el que nadie es capaz de comprenderme. Mi amigo tampoco pudo. A nadie le importo...

¿Pero qué es esto? Ya estuve aquí. Flores. El banquito. La ancianita sigue sentada allí. Parece que no se ha ido en todo este tiempo y con este frío... Indeciso, me siento en el borde del banco. ¡Ay! Mis manos están heladas. Como si algo me empujara, me levanto y me acerco al quiosco de flores. Me gasté el poco dinero que quedaba (igual no lo necesito más) en un ramo de rosas. No dejaba de pensar ¿qué estoy haciendo? ¿Para qué? La abuelita me estaba esperando, dirigiendo su mirada hacia mí. Finalmente pude decirle — Disculpe, ¿puedo regalarle estas flores? ... Pensé mal de usted ... entonces...

Ella no estaba sorprendida, dijo — ¿En dónde has estado tanto tiempo, mijo? Estoy congelada esperándote. Sorpresa no es la palabra correcta para expresar lo que sentí al escucharla... ¡Estaba impactado, aturdido! Ella continuó — ¿Te sientes mal, mijo? — me miró con preocupación. Sabiduría, bondad y... amor brillaban en su mirada. Amor hacia mí... Entonces dije — ¡Sí! Me siento mal. Me siento muy mal... — Lo decía sin miedo ni pena. Era como si algo se desprendiera de mí, como una cáscara. Entonces, comencé a contarle todo... con prisa, atragantándome con mis propias palabras.

Le conté a la abuelita cómo programo chips...

Ella me escuchaba atentamente, no interrumpía. ¡Lo entendió todo! ¡Cada palabra! Lo vi en sus ojos. Hablé y hablé. Ella me llevó a algún lugar donde comí algo delicioso, luego tomamos té con una mermelada espectacular, en un espacio pequeño pero muy acogedor.

Debe haber sido una imagen muy extraña: un programador medio loco y una abuela escuchándolo atentamente... Luego ella habló. No recuerdo bien qué me dijo. Solo recuerdo que decía cosas muy importantes y necesarias para mí. Era como beber de un pozo de sabiduría. Solamente al perder tanto se puede obtener esta recompensa.

De repente... Nuevamente comenzó a desplegarse una estructura efímera dentro de mí. Se desplegó con fuerza y elegancia, rodeada de luces espléndidas. ¡Cada faceta, cada elemento de esta estructura eran perfectos y únicos! La estructura vibraba levemente, generando música. ¡Todo este conjunto me llenó de admiración y asombro! ¡Fui yo quien creó esta magnífica obra! ¡He pulido todas sus facetas! ¿¡Por qué nadie más ve esto!?! ¡Mírenla! ¡Compartan conmigo esta felicidad! Ya no tenía miedo. ¡Sabía que la estructura volará! Me ayudará a despegar también. ¡Con ella, volaré hacia las estrellas!

*
**

Volví a caminar por la calle, pero en otro lugar, mejor dicho, en otro mundo. ¡Este mundo era genial! La nieve salpicaba con chispas multicolores, sonaba como música bajo mis pasos. Es como si un viejo computador con gráficas CGA¹⁸ comenzara a mostrar millones de colores. ¿Qué tontería estoy diciendo? Esto era mucho, muchísimo mejor.

Creen sus propias naves. Naveguen hacia mundos extraordinarios. Contrario a lo que creía antes, cada uno de nosotros puede viajar y vivir en estos mundos. No importa cómo o en qué lenguaje programes. Creamos programas en la misma medida que los programas nos crean a nosotros. Los programas de verdad no se crean con astucia, no requieren solamente

¹⁸ **CGA** (en inglés, *Color Graphics Adapter*) fue el primer estándar de gráficas para IBM PC, introducido en 1981. Soportaba hasta 16 colores.

conocimientos e ingenio. Deben salir del corazón porque solamente pueden ser creados con amor.

Los programas que creamos son obras de arte puras. Por eso nos fascinan tanto. No necesitamos martillo, cincel, pinceles ni pinturas para expresar lo esencial en nuestras obras: ¡nosotros mismos! No importa que, tal vez, pocas otras personas puedan entender la belleza del código. Si nos superamos al escribir el programa, ¡hicimos lo correcto, creamos un buen programa! Quienes creen que pueden ahorrar esfuerzos, se equivocan. Si no están dispuestos a colocar una parte de su alma en la creación, si no están listos para cambiar, entonces lo mejor sería... no programar. Existen muchos otros campos de acción...

Acerca del autor

Incluso el hacker que trabaja solo, colabora y se comunica constantemente con otros, para que su trabajo no se confunda y no se pierda.

«Maestro Foo y el programador prodigio»

Eric Steven Raymond, 2003

Dmitry Galuscenko es físico de profesión, oriundo de la península de Kamchatka (Rusia), uno de los autores más reconocidos en *FidoNet*,¹⁹ la red mundial que se hizo popular antes de Internet, particularmente en los años 90. El maestro Galuscenko solía publicar en el segmento de programación para sistemas embebidos «RU.EMBEDDED».

También ha publicado en la revista ucraniana *RadioHobby*,²⁰ una de las más prestigiosas en materia de electrónica y programación en el espacio post-soviético. Esta revista ha circulado desde 1998 con el sello oficial de la Liga de Radioaficionados de Ucrania (UARL).

Hasta la primera década de los años 2000, el autor ha trabajado desarrollando instrumentos electrónicos para la industria médica en Letonia.

Las publicaciones de Dmitry Galuscenko han contribuido a formar varias generaciones de programadores Zen. Se trata de la escuela oriental Zen²¹ aplicada a la programación, especialmente en lenguaje ensamblador. El programador Zen no busca fama, sino iluminación. Por eso, no es mucho lo que se conoce acerca del maestro Galuscenko.

¹⁹ <https://es.wikipedia.org/wiki/FidoNet>

²⁰ <http://www.radioman-portal.ru/magazin/radiohobby/>

²¹ <https://es.wikipedia.org/wiki/Zen>