```ascii art: Logo Texto-Plano
      _                    _
  ___/ \___             ___/ \___
 /__   ___/\_____  __ /__   ___/\__
   /  / __  /\  \/ /\  /  / ___  /\ Año 2025
  /  / _____/ /\    / / /  / /  / / / Nro. 001
 /  / /_/ /\/ /\  \/_/  / /__/ / /_________
/  /_____/___/ /\___/  /_______/ //_________/\
\  \_/ /\\___\/  \__\  \_/ /\_\/ \_________\/
 \_____/ /________,__  \_____/ /
  \___\// ___  /  /___\___\/      ________
       / /  / /  /___  /\_____    / ___  /\
      / /__/ /  /\_ / /  __  \  / /  / / /
     /  _____/  / __  /  /\_\  \/ /__/ / /
    /  /\__/  /_____/__/ / /__/_____/ /
   /  / /  \  \_____\__\/  \__\_____\/
  /__/ /    \_____/\ : : R e v i s t a : :
  \__\/      \___\/ : : D i g i t a l : :

```

íNDICE

---

00 - Tapa + Indice de artículos
01 - Editorial: La computación comunitaria ~ffuentes
02 - El alarido del Gaucho Vengador ARPACero
03 - CASO PRÁCTICO 1: Análisis de militancia para mejorar Anarcogordo
04 - Slashem ~anthk
05 - El grito del gritón ENtelER
06 - Nomad-Fuguita ~ffuentes
07 - SDF ENTelER
08 - Síntesis de audio ~hora_z
09 - La historia de DOS ~peron
10 - privacidad ~ffuentes
11 - Cielo Azul ~emilio
12 - Colofón ~ffuentes

---

01 | La computación comunitaria ~ffuentes

En el principio fue la mainframe

Cuando explico que levanté un servidor en la red para que cualquier persona lo utilizara siempre me preguntan si es arriesgado, y sí, lo es pero todo en la vida tiene riesgos.

Lo que muchas de esas personas no tienen en cuenta es que la computación comenzó de este mismo modo: Compartiendo una máquina en una época en que ni siquiera existían los computadores personales, ni los sistemas multiusuario. 'Hackers: Héroes de la revolución computacional' es un libro que lo explica muy bien.

A fines de la década del 50, cuando ni siquiera existía la carrera de ciencias de la computación ya había grandes computadores que se utilizaban para labores de investigación. Tal como cuenta Steven Levy en su libro, los ingenieros llegaban con series de tarjetas perforadas, las cuales debían contener código binario escrito con sumo cuidado para poder entregarlas al operador de la máquina (ej. un operador de IBM) quien procesaría la información dentro del sistema e informaría la salida del código. Si el código tenía errores, recibirías un mensaje de error y tendrías que reescribir todo y volver. Era un proceso muy tedioso. Por lo mismo, la escritura del código se volvió un proceso que requería bastante colaboración entre pares. Todos tienen proyectos distintos pero alrededor de un gran computador surge una gran comunidad en la que distintas personas crean y se apoyan mutuamente.

Esta comunidad de "hackers" no termina con las tarjetas perforadas y los mainframes, es sólo el comienzo de una forma de interactuar, no simplemente con máquinas sino con otras personas, desde la capacidad de hacer algo y luchar hasta volverlo una realidad. Cuando aparecen los sistemas operativos multiusuario, la colaboración se expande a otros lugares y ya no tienes que estar necesariamente en frente del mainframe para trabajar en él en proyectos con otras personas.

Hoy en día, existen cientos de hacklabs donde hobbistas trabajan en conjunto en proyectos compartidos, que pueden ser grandes como un nuevo programa de software libre mantenido en un control de versiones como git, o algo tan pequeño como un robot o un cartel artístico para una actividad.

Pero hay algo más. La colaboración a distancia existe hace mucho tiempo y aún existen servidores que llevan décadas de existencia dando acceso gratuito o a un precio accesible a sistemas compartidos donde además de conseguir, por ejemplo, acceso a una consola de Unix, puedes conversar con personas de todo el mundo. No te piden un certificado de antecedentes para entrar, sólo buen comportamiento. Es por eso quizás que he podido crear este espacio llamado Texto-Plano y, junto con otras personas que han surgido en el camino, atraer más personas a esta comunidad.

Lo que queremos hacer con esta revista no es promover texto-plano.xyz más allá de que es una comunidad que existe, que está viva. Lo que queremos hacer es demostrar que una computación comunitaria es posible, incluso en estos tiempos tan oscuros en algunos aspectos, plagado de intolerancia e individualismo extremo.

En resumen, compartir es algo bueno.

Texto-plano

==

Revista digital 01 | 02

----

02 | El alarido del Gaucho vengador ARPACero

"El alarido del Gaucho vengador"

Como muchos usuarios de PC de principios de los 90s en la Argentina, tuve mi copia trucha del TRUCO de los Hermanos Ariel y Enrique Arbiser. Por entonces tenía una AT 286 de las últimas que salieron, con una joyita que por entonces era el monitor VGA. Una amiga -la misma con la que nos metíamos de polizón en la red ARPAC- me lo pasó en un diskette de baja densidad.

Este TRUCO ARBISER, además de autóctono, era un juego buenísimo. De entrada salía un gaucho narigón en CGA que -al son de "la cumparsita"- despuntaba una tremenda mano de truco en la izquierda, y un mate en la derecha. Programado con rutinas de BASIC y Ensablador, el socarrón campesino sintético te desafiaba, gastaba, o sufría: al ir tirando las barajas respondía con toda una andanada de versos y cantitos de acuerdo a cómo venía la mano, todo ello al ritmo del parlantito que venía embutido en las misteriosas entrañas de la PC. Parecía que si te junaba mentiroso, te bolaceaba aún más. Incluso llegaría a tener una "voz" grabada, que reproducía carrasposamente desde el PC Speaker

````
``` Ascii art: Gaucho jugando al truco en una PC
 ________                        __________________
< TRUCO! >                      <  Quiero retruco! >
 --------                        ------------------
    \                                         /
     \   _________                           /
         |        |                   ________
     ____|________|_____      ~      |        \
        //// _____)              ~   ||        _____
        _|   (o)(o)       ~    ~ ||              \
       (9        \|              ~   ||               \
         |     (..)       ~    ~ ||                \
         |    /||||\             ~   ||                 |
         | \     \\       ~    ~ |                  |
        /|  -----___           ~   |___________----|--+
       /  __/_ )    ( ______ ~____      ======      wwwww8
      / +___   (/|    )______||_--\\--+  -                8
     /      +--- +- \/        /________|wwwwwwwwwwwwwwwwww8
    |________---____/
````
````

La cuestión que esta copia todavía no hablaba, era la 3.2, pero venía con un manual o un archivo LEEME.TXT donde los Arbiser explicaban los comandos que había que escribir a mano, o usar las teclas de función como atajos. Pero la cosa no quedaba ahí, pues se explayaban un poquito sobre las bondades y el funcionamiento del juego. Claro, desde lo técnico el TRUCO ARBISER ya no era uno de los juegos más avanzados que tenía en el disco rígido, pero sí el único programa netamente Argentino.

Un día, curioso de cómo funcionaba, se me ocurrió revisar el binario ejecutable TRUCO.COM con un editor hexadecimal que venía con el Norton Utilities (y que permitía observar el contenido del archivo). Para mi sorpresa, era un conjunto alucinante de simbolos ANSI raros y caritas. Fue en esta excursión por el mundo del "no te metás digital", donde descubrí que se podría cambiar el texto de la pantalla inicial y poner algo distinto (por ejemplo nombre y apellido).

No era difícil de hacer, toda vez que la cadena de caracteres originales "(c) 1982-86 Por Ariel Arbiser y Enrique Arbiser Version 3.2" figuraba en texto plano ahí nomás y a la vista casi al principio del archivo; solamente había que tener cuidado de usar la misma cantidad de letras y no "pisar" el resto del binario. Haciéndolo pude modificar el cartelito "y mandarme la parte" por la nerdeada.

Lo que por entonces no sabía era que esta práctica estaba mas bien extendida entre otros usuarios avanzados que -ni lerdos ni perezosos- se habían dado a la soez tarea de "vender" estas copias truchadas. Claro, sin el código fuente, era un crackeo más bien tramposo. Los Arbiser, por buenudos, no podían dejarlo pasar...

Y así, en secreto, idearon un método que hiciera de su gaucho un matrero de la "protección anticopia".

En efecto, reprogramaron su código fuente agregándole una subrutina llamada "CONTROLA_TODO_MAL". Esta -escrita en ensamblador- determinaba si se habían suprimido fragmentos del código, o si este difería del original. En efecto, detectada la alteración de los nombres de los autores, disparaba una cabeza de guerra que operaba sobre la modificación destructiva, apuntándola al intérprete de comandos del sistema operativo DOS -el inefable COMMAND.COM- así como los archivos ejecutables y recursos del juego en sí (archivos TRUCO.EXE, TRUCO.COM, MVYTRUC@, MWTRUC@.BAS y MXYTRUC@).

```
             CONTROLA_TODO_MAL
cd9b b9 01 00   MOV    AX.0x1
cd9e 50         PUSH   AX
cd9f 50         PUSH   AX
cda1 50         PUSH   AX
cda2 b8 04 00   MOV    AX,0x4
cda5 00         PUSH   AX
cda6 9a 13      CALLF  B$LOCT
     01 41 1d
             LOCATE 1, 1

cdab b8 f2 4b   MOV    AX,qbstr_and
cdae 50         PUSH   AX
cdaf 9a 61      CALLF  B$PSSD
     01 41 1d
             PRINT "&";

cdb4 b8 f8 4b   MOV    AX,qbstr_o
cdb7 50         PUSH   AX
cdb8 b8 01 00   MOV    AX,0x1
cdbb 50         PUSH   AX
cdbc b7 fe 4b   MOV    AX,qbstrr_atrucocom
```

```
cdbf 50          PUSH   AX
cdc0 b8 ff ff    MOV    AX,0xffff
cdc3 50          PUSH   AX
cdc4 9a 2e       CALLF  B$00PN
     01 41 1d
              OPEN "o", 1, "a:truco.com"

cdc9 b8 a2 00    MOV    AX,0x2
cdcc e9 56 00    JMP    LAB_1000_ce25
```

El DOS era un sistema operativo "a capot abierto", y no tenía mucho en forma de autoprotecciones. Ya con la primer parte de esta espoleta lógica hubiese sido suficiente para destruir el funcionamiento del programa, y de la copia en disco del sistema operativo, si estaba situado en la diskettera A:, pero el gaucho era muchísimo más sotreta, pues tenía una ojiva multiefecto: cual ebrio en baile se las agarraba con todos los archivos con extensiones .EXE, .COM, .BAT y .SYS que encontrase en las unidades A:, C: y D: y los pasaba a degüello.

En la pulpería que era el viejo DOS, esto significaba básicamente inutilizar además todos los programas los discos rígidos del usuario. Ya no sólo implicaba volver a copiar el COMMAND.COM – que por sana costumbre, se respaldaba siempre en algún diskette formateado y protegido a tal efecto –, sino que obligaba a tener que reinstalar todo el software de cero. Con su rutina los Arbiser habían echado un chancho muerto al jagüel, lo que hubiese afectado a todos los baguales que hubiesen querido recurrir a un editor hexadecimal para firmarles el binario.

Afortunadamente, la cabeza de guerra de esta rutina anticopia –que por poco mas que no ser autónoma no calificaba de virus troyano– no era "antitodo". El bucle de programación que hacía de espoleta se detiene e interrumpe ante cualquier error que pudiese existir entre sus procedimientos, y existen tantas posibilidades de ellos que en la práctica, resulta casi inerte.

Una de las causas podría ser que el juego no esté corriendo desde un diskette, sino desde un disco rígido (la unidad C:, algo cada vez más común a partir de finales de los 80s). Otro podría ser que el diskette estuviese protegido contra escritura (lo que se lograba poniendo una etiqueta autoadhesiva en la muesca que tenía al costado). Otro era que el juego se ejecutara desde la unidad B: (cada vez eran más corrientes las computadoras con dos disketteras, y no extrañaba que la segunda fuese una diskettera de 3 1/2", con diskettes provistos de potección de escritura "integral", con un pasador plástico que ya venía puesto y había que correr para poder grabarlos).

En fin, una protección sumamente agresiva, pero afortunadamente aún bastante dificil de "espoletear" como para que surtiera efecto.

En mi opinión, el TRUCO ARBISER no pudo ser superado. Claro que hubo otros que más o menos reproducían la idea base de este divertido juego de baraja, pero el hecho de programar tan bien las "mentiras", la IA básica, el "bolaceo", hizo que otros juegos de truco más modernos – incluyendo varios populares como el TRUCO SINTESOFT, o varios que han salido para ANDROID – carezcan de la "picardía" que mostraba el TRUCO ARBISER... ¡sin contar su protección anticopia matrera!.

Precisamente sobre esta técnica de protección del TRUCO ARBISER, existe un paper de Gustavo Del Dago (docente de programación en la Universidad Nacional de José C. Paz, Argentina), en la que emprendió la decompilación y análisis del mismo. Pueden encontrar este "Análisis y Estudio del Truco Arbiser, de la programación hogareña al primer videojuego comercial latinoamericano" en mi gopher, en esta URL:

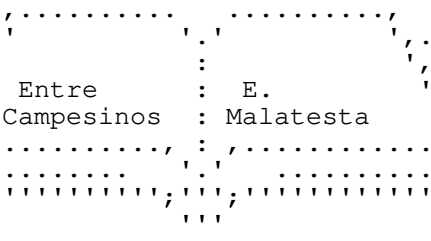
gopher://texto-plano.xyz:70/0/%7eperon/archivo/libros/hack/truco_arbiser.txt

Si desean descargar el juego pueden correrlo sin problemas en DOSBox en sistemas GNU o similares. Tras las décadas, los Hermanos Arbiser lo han puesto a disposición en una de las primeras páginas webs del país.

https://www-2.dc.uba.ar/charlas/lud/truco/truco.zip

CASO PRÁCTICO 1: Análisis de militancia para mejorar

#Catálogo de Biblioteca en Calibre

``` Ascii art: Libro abierto 'Entre Campesinos' de E. Malatesta


```

Una biblioteca tiene una hoja de cálculo en una computadora del 2010. Cada vez que alguien quiere chequear un libro, encienden la computadora con un Windows que se agita cuando sube un piso por escaleras. Cualquier error de tipeo en la hoja de cálculo puede borrar una celda y dios sabe cuando se verá el error. Ud., como militante, ¿cómo les ayuda?

Como decía el arquitecto Rodolfo Livingston, se comienza por la pregunta. Es necesario consultarles para qué quieren y usan el catálogo.

1. ¿Hacen préstamos?
2. ¿Se actualiza seguido el catálogo?
3. ¿Hay expertos en bibliotecología en el espacio?

En este caso, la respuesta a las tres fue no. No se tiene pensado hacer préstamos ni se actualiza seguido. Aquí que hay muchas opciones: Koha, AtoM (access to memory), Aguapey, la hoja de cálculo (total sigue sirviendo) o Calibre. La ausencia de préstamos en el presente y el futuro próximo hace innecesarios Koha, AtoM y Aguapey. De todas formas, dejo a la posteridad mi opinión de aficionado a la informática. Aguapey no lo usé, es de código cerrado y exclusivo de windows. Es un freeware de la Biblioteca Nacional Argentina. Sin embargo, con una interfaz anticuada a prueba de boludos, pensada seguro para que un señor bibliotecario pueda usarlo, un esqueumorfismo absoluto de principios del 2000. AtoM, el Utilizado por el Archivo General de la Nación (Argentina) me resultó muy pesado para una computadora Conectar Igualdad G5. Koha no es tan pesado y es mucho más fácil de instalar incluso en red local, a diferencia de AtoM. Dado que no se iban a realizar préstamos, el catálogo no se actualizaría en el futuro medio y no tienen expertos en bibliotecología, Aguapey, AtoM y Koha eran una bomba atómica para tirar abajo un arbusto. Lo importante era buscar y resguardar los datos.

En principio, sería cómodo quedarse con la hoja de cálculo. Entonces uno elige no hacer nada. Hasta que viene alguien a buscar un libro. Excel y Calc no tienen forma sencilla de buscar texto con múltiples criterios en paralelo. Las soluciones son abismalmente complejas y como militantes informáticxs debemos dar el pescado y enseñar a pescar a la vez. O, en su defecto, otorgar una herramienta que puedan manejar sin vos ahí. Calibre fue la solución a este caso.

Consejo: Hacé esto en una computadora potente, luego podés llevar la carpeta de Calibre en un pendrive y copiarlo a la PC del espacio.

El proceso comienza por sanitizar la hoja de cálculo. Que en la

columna de números solo haya números, corregir ortografía (categorías con tilde o sin, pero no tener FILOSOFIA y FILOSOFÍA), y evaluar qué contendrá cada columna. Si hay letras, será un string/cadena, años serán enteros, booleanos, etc. En este caso, la columna AÑO tenía texto para indicar años aproximados.

```Ascii tabla
+-------------------+----------------------+------+-------------+
|       Autor       |        Título        | Año  |  Editorial  |
+-------------------+----------------------+------+-------------+
| Malatesta, Errico | Entre Campesinos     | 1884 | Anarres     |
| Kropotkin, Pedro  | La Conquista del Pan | 1898 | Reconstruir |
|                   |                      |      |             |
+-------------------+----------------------+------+-------------+
```
Ejemplo: Un autor puede tener mas de una versión de nombre, como Pyotr y Pedro. Decidite por una Única versión.

Puede parecer mucho trabajo, pero te ahorrás de limpiar cagadas a futuro. Ver que las categorías estén bien escritas y no tener varias versiones de una categoría. Esto dependrá de tu biblioteca. Trabajá en una copia, no el original. Luego, transformá la hoja de cálculo CATALOGO a un archivo CVS de /texto/ separado por comillas. Utilizá el símbolo que más bronca te de, mientras no te de errores al importar. En Calibre, instalá el complemento "import list" que está incluído en la sección de plugins. Utilizaremos este plugin para importar un listado de "libros" vacíos. El plugin permite importar desde un archivo CSV y nos va a pedir que completemos y asignemos una etiqueta a cada columna.

Este proceso es muy particular. Calibre tiene sus propias etiquetas en ingles. YEAR es año y es un entero, lo mismo TITLE, AUTHOR, etc. Ya existen. Puede ser que estas etiquetas no sirvan para tu caso particular, por lo que vas a tener que crear las tuyas. Tené en cuenta solo acepta caracteres ASCII, nada de tildes o ñ.

Una vez asignadas las columnas a una etiqueta e importada la biblioteca vacía, vas a poder ver los "libros" en Calibre, que van a ser un cascarón de metadatos sin nada para mostrar. ¿Para qué hacer todo esto?

1) Separás la búsqueda del catálogo. Calibre puede morir que nunca vas a tocar la hoja de cálculos original.

2) La búsqueda se vuelve abismalmente más fácil. Podés buscar con múltiples criterios en simultáneo, como períodos de tiempo, más de un autor, etc.

3) El proceso es relativamente sencillo y, con la voluntad del espacio, podés enseñarles a hacerlo ellxs mismxs.

4) Es exportable a otros espacios de Calibre e incluso, en un futuro, a un gestor de bibliotecas como la gente.

No vale la pena adentrar en las contras porque en la militancia siempre las hay y nadie jamás te garantiza un resultado. Estás vos solx con una idea y podes hacerla o no. ¿Saldrá bien o mal? Nunca lo vas a saber. La experiencia no se puede transmitir. Podés hacer todo bien, con las mejores intenciones, con todo a favor, y fracasar de todas formas.

"Fue en España donde mi generación aprendió que uno puede tener razón y ser derrotado, que la fuerza puede destruir el alma, y que a veces el coraje no obtiene recompensa" -- Albert Camus

Este artículo solo busca mostrarte un caso práctico para que vos decidas si vale la pena o no arriesgarte. Es un buen parche que te permite cambiar de sistema de ser necesario y no encerrarte en una decisión del hoy. Lo que si, como todo, vas a tener que enseñarle a la gente a usarlo. Sentarte varios días a hacer

búsquedas, abrir el programa, practicar, corroborar que funcione, y por sobre todas las cosas, acompañar. La computación es un área donde la gente se suele empacar como una mula en la montaña. No fuerces nada, no se van a morir por usar una planilla si están negadxs. No te lo tomes a pecho. La informática permea toda la sociedad pero es tán compleja (y se le otorga un halo de misticismo no merecido) que solo lxs interesadxs comprenden el nivel de impacto que tiene un cambio tan pequeño como esforzarse en usar otro programa.

Slashem un roguelike, un juego de mazmorras descendiente directo del Nethack.
Nethack nace como un Hack creado a traves de Usenet, una especie de foro en los 80 hecho para discutir sobre tematicas de cualquier tipo de forma parecida a un sistema de email con sus respuestas a hilos.
Y, finalmente, Hack fue escrito como una copia libre y de codigo abierto de Rogue, el rogue original (de ahi viene el termino roguelike).

Este juego (Nethack/Slashem) ha sido jugado durante decadas y se sigue desarrollando. Es tal obra maestra que se exhibe en el MOMA de Nueva York.

Pero, que es un roguelike?
Un roguelike nace como un juego de mazmorras/rol en computador, es decir, como los juegos de mesa donde un personaje tiene varios atributos: (Ataque, Defensa, Destreza, Inteligencia...) y segun esos valores atacará a sus enemigos (y se defenderá por igual) segun los valores numericos de esas variables.
Un guerrero destacará en Ataque, un Ogro en Defensa, un Mago en Inteligencia (donde los hechizos son mas o menos efectivos segun ese valor).

En el caso de Nethack, tu cometido es rescatar el amuleto de Yendor robado por Moloch en el nivel mas profundo de la mazmorra, y volver arriba de todo y 'ascender' con ese amuleto para ofrecerlo a tu deidad.

Pero no es tan facil, ni por asomo.

Al contrario que muchos juegos de rol, los roguelike tienen un componente de aleatoriedad en el diseño de mazmorras de cada partida (salvo las mazmorras principales de la trama del personaje, donde no varian), incluso las pociones de diferente color son aleatorias en cada partida. Eso lo hace mil veces rejugable, y por supuesto, mueres.
Mueres porque en un roguelike las muertes son para siempre, y las mecánicas en esos juegos consisten en aceptar las derrotas y saber cómo has muerto para futuras partidas. Un poco como la vida.

Si se quiere ver una vista general de Slashem, se ve asi:

```Ascii art: Pantalla de juego de Slashem

  The bat bites!

              ------
              |....|     ----------
              |.<..|####...@...$.|
              |....-#    |...B....+
              |....|     |.d......|
              ------     -------|--


Player the Rambler     St:12 Dx:7 Co:18 In:11 Wi:9 Ch:15  Neutral
           Dlvl:1 $:0  HP:9(12) Pw:3(3) AC:10 Exp:1/19 T:257 Weak
```

Que es todo esto?

Antes que nada, esto es una vista aerea de la mazmorra.
Los rectangulos hechos a traves de | y --- son paredes de salas,
los # son caminos, < es para ir escaleras arriba, tu personaje
es una @, $ es oro, B un murcielago (Bat en ingles, B de Bat) y d
un perro. Finalmente, + es una puerta cerrada, y un . es suelo.

Abajo de todo sale el nombre de tu personaje, el rol, el valor de
fuerza, destreza, constitucion, inteligencia, sabiduria y carisma,
junto con la alineacion moral.
En el ultimo renglon sale el nivel de la mazmorra, la vida, el poder
para realizar hechizos, el tipo de armadura y la experiencia.
La T son el numero de turnos y 'Weak' indica debil, es tu estado de
inanicion.
Debes comer en el juego para no morirte de hambre, por cierto.

Tan simple como eso, pero el juego tiene cientos de ordenes.
Estas se realizan con un comando de teclado.
Por defecto, en nethack/slashem para moverse se usan las teclas hjkl
para ir izquierda, abajo, arriba, derecha, como en el editor vi.
Para ir en diagonal, yubn, exactamente las direcciones que apuntan
desde la tecla 'h' fisicamente en el teclado.
¿Obtuso? Puede, pero la mano descansa al recorrer menos el teclado.

Para ver qué realizan el resto de órdenes, se puede usar la tecla '?'
mientras se esta jugando una partida, pulsando despues 'b' y con los
cursores leer hacia donde pone 'commands:' donde se ve claramente su
función.
En ingles, pero nadie es perfecto.

Te preguntaras qué hace especial a Slashem/Nethack:
la interactividad y las situaciones emergentes.
Imagina que eres un Rol Turista (de Mundodisco, si, el de Pratchett)
mujer, te convierten en dragona al beber una pocion o con un hechizo:
al poco podras parir huevos de dragones hijo, y usar esos dragoncitos
para reventar a los enemigos de la mazmorra para completar tu 'Quest'
(mision) en el juego.

Tambien lo que lo hace especial es la rareza anacronica que posee
el juego y la posibilidad de razas y roles: tienes desde camaras
fotograficas a samurais yakuza con ametralladoras, personajes jedi
con sable laser en segun que ediciones, monjes doppleganger con ataques
a lo Dragon Ball lanzando rayos de energia, picaros con dagas envenadas
accediendo a hechizos de El Señor de los Anillos, arqueologos metidos
en mazmorras con un latigo para nada no inspirado en cierto personaje
de Hollywood...

La mejor forma de empezar a jugar a Slashem es, segun eliges tu raza y
rol y entras en la mazmorra como la 'imagen' de arriba, es pulsar '?' y
luego 'j' para leerte el Guidebook de cabo a rabo.
Es ingles pero no demasiado avanzado, no es una obra de Shakespeare, y
siempre puedes usar un traductor online para los terminos que no
conozcas, ya que el juego es por turnos y no te apresura en nada.
Ah, y se abre sobre cualquier chatarra de PC que tengas.

Happy Hacking!

Hubo un HACKER muy conocido que fue EL GRITÓN, el primer hacker
argentino que cayó en "una grande".

Era un estudiante de ciencias de la computación en la universidad John
F. Kennedy, y le gustaba pasear por los Estados Unidos... en módem.

Durante el verano boreal de 1995, el Departamento de Defensa gringo
detectó intrusiones en una variedad de sistemas de cómputo
universitarios y militares, que provenían de un mismo perfil.
El intruso buscaba información sobre investigación satelital, radiación
y energía.

Pudieron rastrear la actividad hasta un conjunto de cuentas apropiadas en un mainframe de la Universidad de Harvard.

Los servicios estadounidenses cayeron en su empleo sospechoso y lo perfilaron "a la americana": fantasioso, paranoide, sin mucho dinero, regulares conocimiento de telefonía, DOS y los UNIX "nuevos", pero genial en su meticulosidad. Un joven blanco, de familia militar, con acceso a la burocracia de las comunicaciones. Tras lograr una órden judicial, la primera en su tipo, usaron un programa llamado I-Watch (del cual dudamos todavía que existiese realmente, pero era lo que se dijo a la prensa). Y el caldo no tardó en espesarse: era de un país sudamericano.

En plena "Era de las Relaciones Carnales" entre los EE.UU. y la Argentina, sólo bastó un tubazo en mal castellano desde el FBI a la Policía Federal del país, para que "no le cuidaran ni una crin al pingo de la querencia". Los monos de azul le "cayeron" en el departamento; ¡hasta le entraron por la ventana del balcón en la exageradamente espectacular modalidad hombre-araña! Fue medio al pedo: el GRITÓN no estaba, los que se llevaron el susto terminaron siendo sus hermanitos, personalmente culpables de disfrutar de un muy soviético Tetris en una 286.

A las dos horas un natalia-natalia metió una llave en la puerta. Un cana flaco con anteojos rayban truchos color caramelo lo encañonó con una 9 milímetros Ballester Molina como si fuese un pirata del asfalto. Pero el GRITON tenía una orden de allanamiento más que nada por robo de pulso telefónico a TELECOM (ya después de la privatización de EnTel). Las facturas terminaban destinadas vaya a saber a quién. Pero eso fue dentro de las fronteras de la Argentina...

Fuera de ellas, cierto tufillo terminó atrayendo al FBI y a la justicia militar naval estadounidense, quienes lo usaron de ejemplo público en su "eterna interna" interservicios, en este caso con el servicio secreto.

En su momento no tardaron en tracender sospechas erizantes. El padre del GRITÓN era un mayor en el Batallón de Inteligencia 601 del Ejército Argentino, del que se decía "había dado que hablar en las mesas de reunión". En cualquier organización de ese estilo la mesa de reunión consiste en desglose analítico de un conflicto, pero en la Argentina de finales de los 70s un término así era ya mucho más jabonoso, y bien podía equivaler a la aplicación de tortura seguida de muerte y desaparición. Se aducía que la comunidad de los servicios del norte "conocían con qué bueyes araba".

No parecería extraño que un calurosísimo 28 de diciembre de 1995 se decidiera entonces a atender los cargos de felonía con computadoras, fraude telegráfico y pinchaduras ilegales. Si bien los había declarado en primera instancia obcenos - en vista de que no había extradición ya que tal crímen no estaba regulado en la Argentina - decidió jugar el papel de Mr. Nice Guy en un arreglo que consideró le sería más provechoso.

Julio César Ardita se declaró culpable y "cantó como Pavarotti en la ducha".

Viajó voluntariamente a los EE.UU. (de su bolsillo, dijo, y quedó bien) y cooperó. Y quedó mejor.

En esencia lo que hacía era comunicarse por módem desde su propio departamento robando líneas de la caja que Telecom de Argentina tenía en la terraza de su departamento. Usaba un número 0-800 (gratuito) para acceder a la central de telecom, y supuestamente accedía al CARRIER con una clave alfanumérica de 14 dígitos (lo que lleva a sospechar que tenía el dato de adentro). Usando el servicio de DDI (discado directo internacional) comunicaba con la universidad de Harvard en los EE.UU. y pudo acceder a su UNIX, un mainframe que contaba con 16 mil cuentas registradas. Y luego, con un sniffer de paquetes pudo levantar las antiguas conexiones Telnet y observar así usuarios y contraseñas de aquellos quienes usaban tal sistema, trampolinear a otros, así como también algo de su sistema de correo electrónico. De esta forma pudo lograr accesos a los outdials y servicios externos a Harvard

(incluyendo el de defensa).
Lo que no queda muy claro es cómo elevó su nivel para tener capacidad de uso para el sniffer, ahí está la verdadera "hackeada" del GRTIÓN...

En ultima instancia evitó la sentencia declarando que fue "un exceso de la juventud", prestando servicio comunitario "enseñando seguridad informática", y pagando una suma de 5.000 dólares americanos.

Lo usaron políticamente para implementar lo que la Fiscal General Janet Reno describió como "una telaraña global para detener cibercriminales" tanto dentro como fuera de las fronteras de los EE.UU.

Es muy interesante, aunque un poco complicado el tema de las distribuciones Live de BSD. Con cualquiera de ellas es más fácil probar BSD en hardware físico sin perder datos.

Ninguna es muy conocida y quizás las más destacadas son las del título de esta publicación:

- NomadBSD

NomadBSD es una distro de FreeBSD que tiene como objetivo entregarte un sistema de escritorio listo para usar. Si bien se puede usar como herramienta de rescate está pensada para ser persistente, es decir, para guardar cambios y todo eso. Viene con un manejador de ventanas Openbox, varias aplicaciones y por su persistencia de datos puedes instalar nuevas aplicaciones que se conservarán en el tiempo. Sólo necesitas un pendrive con buena velocidad de escritura (ojalá de una buena marca y USB 3.0 o superior aunque se puede usar el puerto USB normal).

La imagen pesa aprox 5 GB. En mi experiencia personal lo probé en mi ideapad z400 (lenovo) y tiene compatibilidad de wifi y resolución en el arranque pero se lagueaba mucho debido posiblemente a que lo probé en un pendrive barato que tenía por ahí. El sonido anda impecable, Firefox anda sin ningún problema. En el primer arranque exige una configuración y un reinicio. Esto es en todo pendrive nuevo. La estética del sistema es atractiva y liviana, incluso incluye un menú de aplicaciones tipo OSX abajo y un administrador de paquetes que recuerda a Synaptic. En la web de nomadbsd.org hay bastante información pero el sistema en sí es apropiado para personas con poca experiencia.

 - Fuguita

Fuguita es un sistema desarrollado por un usuario de OpenBSD japonés. Fugu significa "Pez globo" e Ita significa "disco". Es un sistema más complejo que el NomadBSD pero también es persistente. Permite probar OpenBSD como un sistema de prueba, con o sin entorno gráfico (que podemos elegir a partir de un menú). Es NECESARIO leerse la documentación antes de probarlo, ya que al instalar el sistema uno sólo cuenta con la consola de OpenBSD y pocas aplicaciones. Se puede ejecutar completamente en RAM o leer y escribir del pendrive.

Para realizar la persistencia uno debe hacer uso de una herramienta que copia el contenido como un respaldo en una unidad USB en una especie de copiado de todo el sistema actualizado. Esto permite también copiar la información a otros pendrives. La imagen pesa aprox. 2 GB. Lo probé en el mismo equipo y el rendimiento es impecable. Copiado en RAM es rapidísimo.

Al igual que OpenBSD normal, es mejor usarlo por primera vez en una conexión cableada o bien portar el firmware de la tarjeta de red WiFi. Al principio me costó usarlo porque no leí bien la documentación y eso era necesario. Copiado en RAM el volcado al pendrive era muy grande ya que quería copiar todo. Por lo demás pude instalar aplicaciones y usar cwm o instalar Mate. Temas como el sonido o la resolución de pantalla no fueron un problema ya que detectó todo perfectamente desde el

arranque. Por lo demás se debería esperar un comportamiento muy similar al de OpenBSD normal. Lo único que realmente cambia es cómo se maneja la persistencia (con usbfadm) y la facilidad para instalar entornos gráficos y configurar teclado, etc.

Un detalle es que la documentación más completa del sistema se encuentra en japonés. Recuerdo que Solene (la desarrolladora de OpenBSD) le hizo un artículo https://dataswamp.org/~solene/2020-11-18-fuguita.html Si bien es un Live USB/DVD no lo recomendaría en absoluto a novatos o gente que sólo se maneje bien con herramientas gráficas ya que OpenBSD requiere bastante investigación para manejarse bien. Fuguita no es (ni pretende ser) más intuitiva o más fácil que OpenBSD "normal".

Arte: Lopecito para Triple-ASCII

Principios de 1987. Ted Uhlemann ("Iczer"), de la ciudad de Dallas, aprovecha su ya veterana Apple //e de dos disketteras y un módem Hayes serial de 1.200 baudios, para armarse una BBS.
La llama MicroBBS Ciudad Mágica.

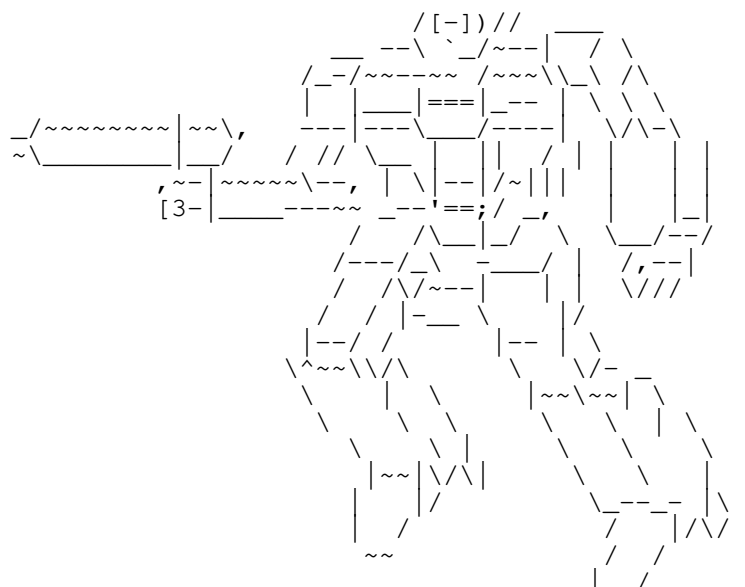
Durante la noche Iczer desenchufaba el teléfono, conectaba el módem, y lo dejaba en espera. El joven SysOp de 13 años empezó a fraguar en su máquina un Grupo de Interés sobre la temática de la animación japonesa (que por entonces tenía poca difusión en los EE.UU). Lo llamó "SDF-1" en honor a la gigantesca nave-robot transformable de la ópera espacial Macross, la que -además de surcar el espacio-tiempo- oficiaba de fortaleza y albergaba una ciudad entera en su interior.

Al poco tiempo la BBS empezó a hacerse conocida gracias a unos volantes y una nota en la revista del colegio: la BBS comenzó a despegar. Era una de las primeras de Dallas con temática juvenil.

Operando de forma nocturna con una línea casera, comenzó a recibir llamadas locales, regionales y nacionales. El tablón de SDF-1 era exitoso: junto con algunos gráficos en ASCII se intercambian opiniones sobre la space ópera de Kawamori, su música, y se discutía haciendo caso al éxito de culto. Pero el hardware -bastante alejado del futurista hogar de la alelada Minmei y el valiente Rick Hunter- comenzó a quedar chico enseguida.

Fue por ello que otro de los tele-residentes más fanáticos, Stephen M. Jones ("smj"), un pelirrojo de 11 años que conocía al SysOp de la BBS "La Corte del Rey Arturo", ofició de relacionista para agenciarse un disco rígido de 5MB que aquél estaba por tirar a la basura. El Winchester MFM sonaba rasposo y trepidante, pero aún así le pudieron sacar unos 3MB útiles, lo que venía bien para almacenar mensajes. El cambio del módem por uno de 2.400 baudios hizo el resto. Podría decirse que ahora la SDF era una verdadera nave.

```Ascii art: Robot de Macross
```



BOOBYTRAP
```
Los pendejos iczer, smj y Daniel Finster ("df") también se hicieron adeptos a conectarse a otra BBS, aprovechando que sólo pagaban pulso local. A esta se conocía como "La Conexión UNIX", operada en la misma caracterísca por un tal Charles Boykin.

Boykin, programador de 50 años, se desempeñaba en el gran comercio de venta de harware Infor-Mart, y había adquirido y puesto en uso un mainframe Western Electric 3B2/500 que estaba en el local.

```Ascii art: Computador AT&T con Unix SystemV
                  .-----------.
      .---------. | == |      |
      |.-"""""-.| |----| AT&T |
      ||  UNIX  || | == |      |
      ||SystemV|| |-----------|
      |'-.....-'| |:::::::::::|
      `"")---(""` |.__________.|
     /:::::::::::\"           "
    /:::=======:::\
    `"""""""""""""`
```

El hardware era una apuesta tecnológica fuerte de la telefónica en el negocio de las computadoras. La habían pensado para dominar, pero no sabían a qué... A Boykin le daba pena. Provisto del UNIX System V de AT&T, su tarea era dar acceso a foros técnicos, pero no se quedó en eso. Tanta potencia le permitía ofrecer acceso público a una shell UNIX en el propio sistema. Este intérprete de comandos permitía usar la suite UUCP, y con ello se operaba el nodo "attctc" de la USENET. En resúmen, tras loguearse a la BBS "Conexión UNIX", y a tiro de un par de comandos, cualquiera podía lograr acceso a la ARPANET.

Los adolescentes aprovechaban con desparpajo el System V. Pero no estaban solos: provenientes de toda Texas y partes del sur estadounidense, Boykin acumuló una foja de más de dos mil usuarios de redes de datos de área amplia y telecomunicaciones, que anteriormente sólo habían estado al alcance de los académicos.

El nodo attdct era "conocido", incluso en la Argentina, por la funcionalidad que proveía como outdial telefónico en suelo norteamericano. ¡Y como para no serlo!

Cualquier hacker argentino con el buen tino "de no pagar de mas" podía enlanzar desde una simple línea telefónica de ENTel a la red de datos nacional de paquetes conmutados ARPAC. Gracias al descalabrado empleo de seguridad de esta red, podía entrar a la DEC VAX que tenían en la cerealera de "Goyo", o a la veterana TWENEX del Centro Atómico (cualquiera de ellas provista de un enlace x.25 24/7). Y así, con la cuidada facilidad intermediadora de attctc en Dallas, o alguna mas en Atlanta, sumadada a la proverbial vista gorda del gremio de Esquivel y Barone, podía salirse a las redes de datos estadounidenses. Eso si, a la friolera de 1.200 baudios y -en el peor de los casos- abonando el Plan MegaTel en hiperdevaluadísimos australes...

En los Estados Unidos, la crisis vendría por otro lado. El burocrático hostname de ARPANET attctc refería nominalmente al "Centro de Tecnología del Cliente de AT&T", relación que ofrecía Boykin en tareas soporte. Era lógico: el técnico se había hecho con la 3B2 aprovechando las facilidades del plan de leasing que esta compañía reservaba a los empleados que le prestaban servicios por fuera de la nómina.

```Ascii art: Logo AT&T
  ___  _______   ____   _______      _____
 / __ \|__   __| /    \ |__   __|   _________
| |__| |  | |     \ [__|  | |     ___________
|  _  |  | |     /    _  | |     ___________
| | | |  | |    |  []|  |  | |      _________
|_| |_|  |_|     ____¬_|  |_|       _____
```
```

GRITO DE GUERRA

Pero este hostname no guardaba la misma aprensión de negocios que el rimbombante URL reservado en la red TCP que se estaba comenzando a armar y que formaría el cimiento de la nueva internet. Boykin eligió para este killer.dallas.tx.us, o "Asesino".

A finales de los 80s, estaba de moda en los EE.UU. decir que algo era "asesino" para denotar grandes capacidades o una inusitada potencia. Y Killer estaba expandido al máximo con varias plaquetas controladoras atadas a unos discos de 525 megabytes, tan grandes que al mismo fabricante les costaba darles salida.

Pero esa explicación fanfarrona puede también sonar poco sensata a una persona incorrecta.

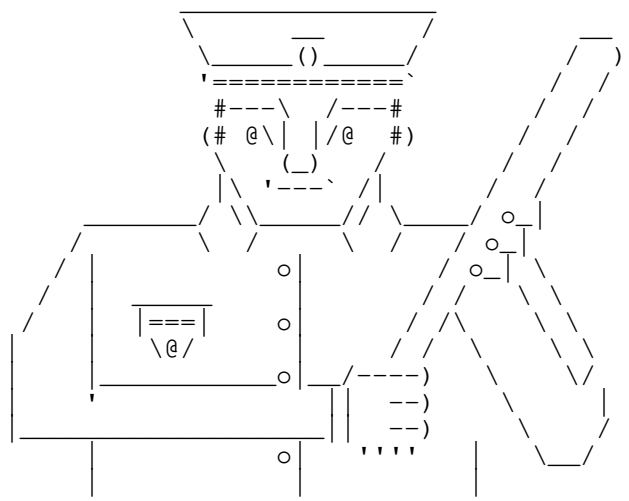
CUENTA REGRESIVA

Por entonces la Fuerza de Tareas contra delitos informáticos (que agrupaba tanto gente del FBI y del Servicio Secreto de los EE.UU. como a representantes policiales y legales de nivel estatal), estaban a la caza. El día de Martin Luther King de 1990 los había desesperado: tanto las líneas telefónicas de media como las de larga distancia habían sufrido una interrupción masiva. La realidad era aburrida: el corte había sido provocado por un error de programación localizado en el software UNIX de sus conmutadores de última serie.

Pero el horno no estaba para bollos. La opinión pública no entendía y demandaba respuestas. No tardó en armarse el circo de los hackers informáticos, inadaptados cucos de cómputo capaces de subvertir los sistemas telefónicos de América.

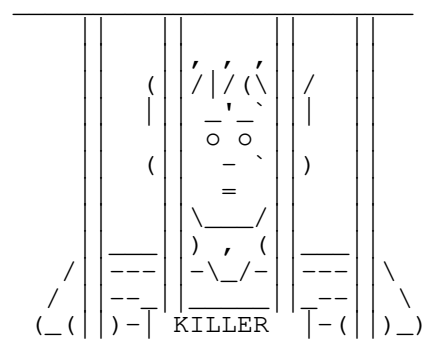
Durante la serie de redadas espectaculares conocidas como la "Operación Sundevil", dieron con todo aquello que -imaginaron- podía constituir delito telemático (desde uso ilegal de líneas hasta un uso totalmente legítimo de computadoras).

```Ascii art: Oficial de policía



```

Pero al margen de esto, "Killer" cayó en la volteada, le pegaron un tijeretazo a sus cables, unos 3,2 GB de información comunitaria que se encontraban en los rígidos ESDI se perdieron para siempre, y a Boykin lo mandaron a tostarse en barras para que se ablandara "por las dudas".

```Ascii art: Killer tras las rejas


```

De buenas a primeras, Iczer, Smj y Df dejaron de escuchar el canto de jilguero del módem a 1.200 baudios, ya no pudieron consultar sobre UNIX, y desde la Argentina, como en todos lados, previeron que el salto a la tapia para acceder a la ARPANET había tocado el arpa por última vez.

DESOLACIÓN

Las aguas tardaron muchos meses en calmarse. Los tres decidieron armar ellos mismos otro sistema que pudiese operar con la calidad que ofrecía la antigua Killer. Optaron por entonces por una 386 sin gabinete y discos rígidos MFM ST506, corriendo el System V Publicación 3.2 1.0 de la Kodak Interactive. Para entonces Richard Losey (rlosey) les proveyó amablemente un AST de cuatro puertos, que permitían usar cuatro módems y así lograr las ansiadas cuatro líneas telefónicas. Lograron así dar acceso a la ARPANET, que ya cursaba su ocaso.

La salida de Boykin de sus problemas legales no significó que le devolvieran su viejo mainframe (que terminó ilegalmente para AT&T). La aparición de documentación sobre el sistema de cómputo del 911 de emergencias en "Killer" y el hecho de que fuese un trabajador tercerizado de la telefónica había complicado todo. Resuelto el problema con las reservas del caso, al menos pudo ayudar a smj y varios más en montar un AT&T UNIX System V r.3.1 puro en distinto hardware, compuesto por mainframes 3b2/400 con CPU WE32100 de la misma marca. Con varias líneas integradas, ya la BBS SDF comenzó a formar parte de un verdadero cluster de cómputo accesible a distancia.

TRANSFORMACIÓN

SDF ahora podía tener conexiones telefónicas, prestar acceso a UUCP en ARPA, y noticias USENET, todo con ficheros compartidos remotamente sobre red STARLAN local. Y para todo el estado de Texas.

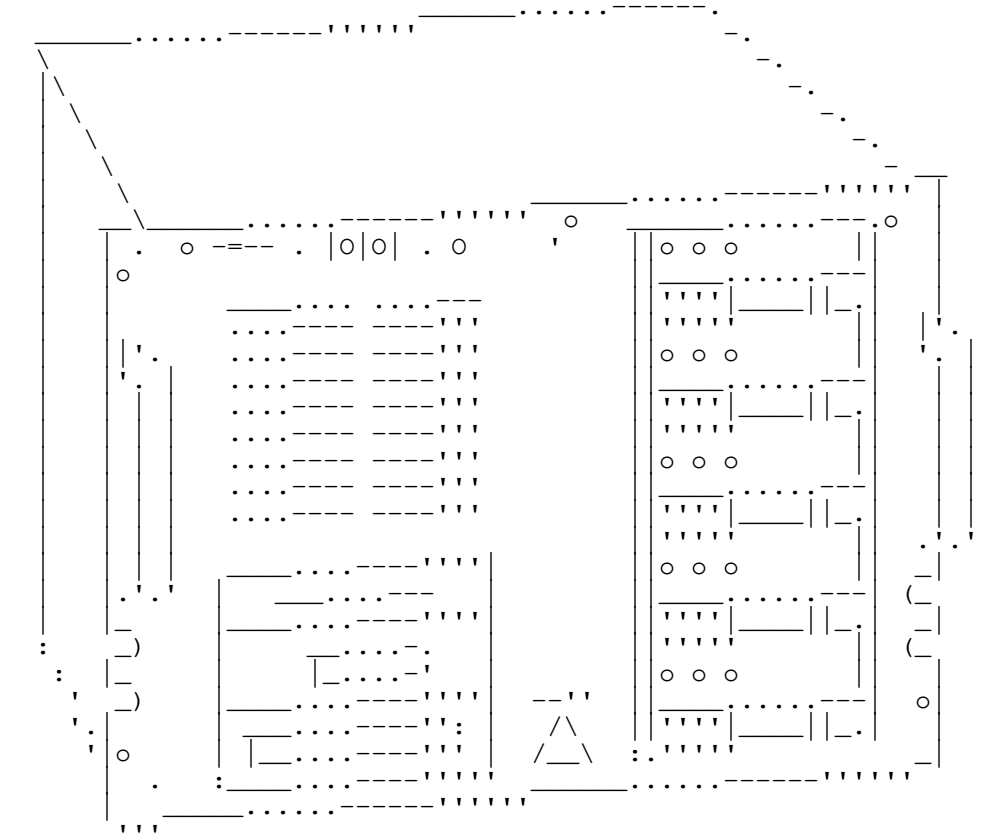
A partir de 1996 se comienza a proveer acceso a internet sobre TCP, bajo el concepto de "Red Social Ética", y coincidiendo con la incorporación de equipamiento de arquitectura DEC Alpha, comienza el uso libre de la red.

Como puede imaginarse, la era de ciertos sistemas operativos privativos, y la comercialización de internet preocupó a estos veteranos Operadores de Sistemas. Por tal motivo decidieron conformar una asociación que hiciera de su objetivo aquel credo en común de proveer cómputo gratuito accesible a distancia. El manejo más directo, responsable, respetuoso y artístico - decían - era lo que movería a sus componentes.

```Ascii art: La SDF Hoy

```

Luego de su sinuoso viaje en el ciberespacio, la Fortaleza Super Dimensional SDF es hoy un clúster moderno basado en múltiples procesadores AMD Opteron, pero también equipos retro como DEC Alpha, TOPS-20, Symbolics Genera y varios más.

```Ascii art: Equipo informático - Servidor

```

Es base de la conmunidad Pública UNIX más antigua, compuesta por autores de software libre, profesores, bibliotecarios, estudiantes, investigadores, hobbistas y entusiastas informáticos en general. Funciona como una Organización No Gubernamental sin fines de lucro, y está mantenida por sus propios miembros.

Su misión principal es la de proveer infraestructura computada remotamente accesible con el objeto de promover el avance de la educación pública, el enriquecimiento cultural, la investigación científica y la recreación. Otros propósitos declarados incluyen el intercambio recreacional de información concerniente a las artes liberales.

La Comunidad SDF está conformada por individuos amables y altamente calificados que obran desde las sombras para mantener en pié una INTERNET no comercial.

Para los usuarios del Sistema UNIX de Acceso Público SDF, la "SDF Social" está basada en los conceptos y principios de estas Redes Sociales Primigenias de las que siempre han sido parte, y no de aquellas Redes Sociales basadas en empresaas con fines de lucro, altamente comercializadas y éticamente cuestionables de finales de la década del 90 y del año 2000.

El acceso es libre y gratuito, si bien el entorno puede recibir donaciones y segmentar su alcance con servicios de costo reducido.

Sin importar su localización, los Miembros de la Comunidad son capaces de utilizar terminales de computadora para interactuar entre sí por medio de foros pasivos o interactivos. Tras establecer enlace

telemático con el centro de cómputo SDF, se operan distintas Shells o intérpretes de comandos UNIX históricos con el fin de acceder a una miríada de servicios informáticos. Entre las aplicaciones remotas se cuentan las casillas de correo electrónico, USENET, chat, sistema de cartelera electrónica BBoard, páginas Web, Gopherholes, utilidades de programación, archivadores, navegadores, pero también servicio de VoIP, casilla de voz Asterix, radio en línea con programación y música 24x7, y muchos más.

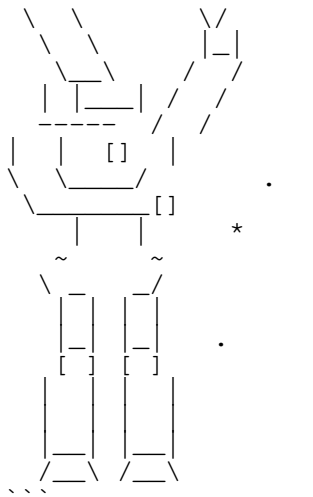
PRIMER CONTACTO

Quien desee puede obtener un acceso (en inglés) desde un navegador moderno con HTML5 puede acceder a https://ssh.sdf.org. O idealmente crear una cuenta desde un shell seguro. En tal caso, desde cualquier terminal con acceso a internet, se puede acceder a crear una cuenta con:

ssh new@sdf.org

(usuario "new")

``` Ascii art: Robot saludando con fondo de estrellas



```

Texto-plano

Revista digital 01 | 08

---

08 | SÍNTESIS DE AUDIO ~hora_z

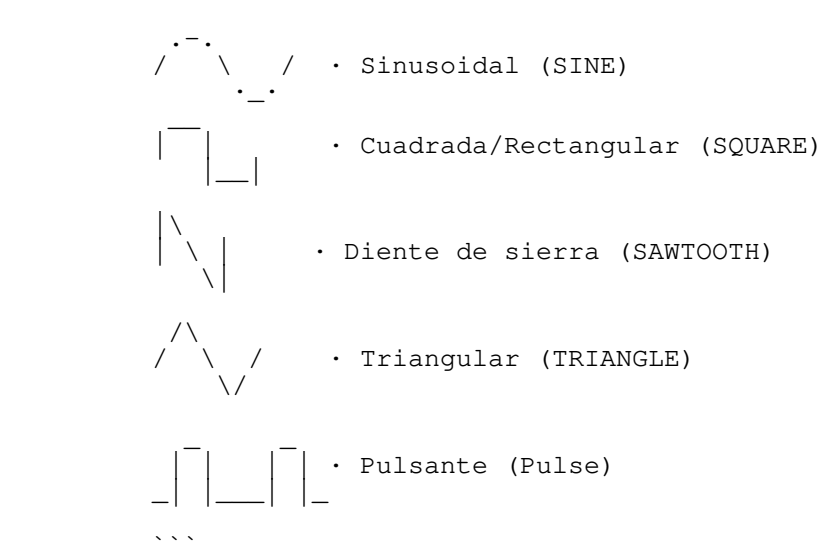
Vamos a hablar un poco sobre las distintas técnicas de síntesis de sonido y a nombrar algunos de los sintetizadores más comunes.
¿Qué geranios es la síntesis granular? ¿Y la sustractiva? ¿Qué es la tabla de ondas? ¿Es como una tabla de quesos, pero ondulante?
En este opúsculo, pretendo echar algo de luz sobre estos temas.

Síntesis de audio: Sonido creado por medio de circuitos.
Los sintetizadores pueden ser:
Analógicos: utilizan circuitos electrónicos que producen variaciones en el voltaje.
Digitales: Utilizan DSP (Digital Signal Processing) para producir los sonidos. Pueden emular a los analógicos e ir aún más allá.

¿Pero cómo es que se producen los sonidos?
Por medio de un componente electrónico llamado oscilador.

EL OSCILADOR
El oscilador envía impulsos eléctricos de forma periódica a través de los circuitos que, al final de la linea, se traducirán en vibraciones de la membrana de un parlante o unos y ceros en un archivo de audio.
El VCO (Voltage Control Oscilator) es un tipo de oscilador que, según el voltaje que se le aplique, puede alterar su frecuencia (altura del sonido) dando lugar a las diferentes notas. Estas notas suelen estar mapeadas a las teclas del sintetizador.

Los osciladores pueden generar ondas de distinto tipo. Las más básicas son Sinusoidal, cuadrada y diente de sierra.



Algunos pueden crear ondas de mayor complejidad -incluso pueden generar ruido- pero la mayoría del sonido sintetizado parte de estas ondas básicas que luego se pueden modificar por medio de módulos.
Es por eso que esta manera de generar sonidos se denomina SíNTESIS MODULAR.

ENVOLVENTE
Una de las modificaciones básicas que podemos hacerle a nuestra onda -y la más directa- consiste en ir cambiando la amplitud del sonido. Esto es, toquetear el volumen. Un módulo que se encargue de esto dividirá la onda según su evolución en el tiempo, en 4 etapas que son conocidas por sus siglas en inglés:

|A|- ATTACK - ataque: el inicio del sonido
|D|- DECAY   - un decaimiento momentáneo hasta el punto de sustain
|S|- SUSTAIN - el sonido se sostiene mientras se mantiene activo
|R|- RELEASE - finalización del sonido

Por ejemplo, pensemos en una campana que tiene un ataque repentino y un apagado lento. A diferencia de un cello, que puede tomar volumen de a poco o un piano, que puede cortar su sonido de golpe.

FILTRO
Un filtro es un módulo que permite, justamente, filtrar armónicos, dando mayor riqueza y variedad al sonido que sale del oscilador.
La aplicación de un filtro para generar sonidos a partir de una onda diente de sierra o cuadrada, es un tipo de creación sonora denominada SíNTESIS SUSTRACTIVA
Se quitan armónicos para cambiar la sonoridad de la onda fundamental. Es como esculpir una imagen partiendo de un bloque de arcilla e ir desvastándolo hasta conseguir la forma deseada.
También podemos aplicarle una envolvente (ADSR) al filtro, logrando sonidos que pueden ir cambiando a lo largo del tiempo.

El sintetizador sustractivo más famoso es el MOOG
Pueden encontrar una versión para linux en el siguiente enlace:
https://linuxsynths.com/RaffosynthPatchesDemos/raffosynth.html
Hay versiones para windows, como el FreeMoog o el MiniMogue VA.
Pero mi preferido es el Sylenth 1, que pueden encontrar en su web:
https://www.lennardigital.com/sylenth1/

SíNTESIS ADITIVA.
La idea de que una onda puede modificar a otra es muy antigua. Si pensamos el sonido como un estanque de agua podremos entender cómo funciona. Si tiramos una piedra al agua se genera una serie de ondas concéntricas. Tirando una segunda piedra, a cierta distancia de la

anterior, crearemos una nueva onda. La onda original se verá alterada por la nueva. Y así, se irán expandiendo unidas en una nueva onda que será la suma de ambas.
De la misma manera, la síntesis aditiva parte de una onda fundamental a la que se van sumando osciladores de ondas sinusoidales que, cada uno con su envolvente, su control de amplitud y frecuencia, aportan riqueza al sonido, agregando armónicos a la fundamental.
Esto va aumentando el consumo de recursos a medida que se agregan osciladores a la mezcla sonora pero el grado preciso de control y la meticulosidad con la que se puede moldear un sonido hacen que valga la pena el costo de procesos.

El sintetizador aditivo más conocido -después del Hammond y de los órganos de tubo, claro- es el K5, de Kawai.
El K5 permite manejar ¡hasta 128 armónicos a la vez!
Algunos VSTi interesantes que utilizan esta técnica:
RE-AXT - de KrakLip
https://www.krakliplugins.com/re-axt
Zebralette - de U-He
https://u-he.com/products/zebralette/
TuneFish - (Usa wavetable de aditivos - what???)
https://www.tunefish-synth.com/

SíNTESIS GRANULAR
La llegada del audio digital permitió no sólo reducir los enormes y aparatosos muebles que eran los viejos sintetizadores analógicos, sino también experimentar con nuevas maneras de trabajar con el audio.
Tomemos por caso un sonido, al que podemos descomponer en partes muy chiquitas, llamadas 'granos'. De ahí que se llama síntesis granular.
Los granos pueden tener distintas duraciones, pueden ser modificados y vueltos a poner en secuencia, no necesariamente en el orden original.
Este rearmado de los granos en una linea sonora puede ser sincrónico o asincrónico, dependiendo del tiempo en el que caen las muestras.
Tendríamos como resultado un sonido nuevo. Distinto. A cada grano se le pueden modificar los parámetros: duración, frecuencia, dirección, envolvente, etc. Además, variando la cantidad de granos por segundo, podemos modificar la densidad del sonido. ¡Una locura!

Los sintes más conocidos son Texture Lab, de SonicWare y GR-1 de Tasty Chips Electronics.
En formato virtual, propongo un par que encontré con versión Linux:
Emission Control 2
https://github.com/EmissionControl2/EmissionControl2
GRainbow - de Strange loops
https://strangeloopsaudio.gumroad.com/l/gRainbow

WAVETABLE - TABLA DE ONDAS
En la síntesis por tabla de ondas, el oscilador cicla a través de una serie de formas de onda previamente almacenadas en una tabla.
Se crea así un sonido que evoluciona en el tiempo mientras muta de una forma de onda a la siguiente, creando un sonido dinámico y complejo.
Luego, como en los demás tipos de síntesis, podés aplicar envolventes, filtros y efectos para darle un toque final.
Algunos sintes con esta técnica:
WaveTable - de SocaLabs
https://socalabs.com/synths/wavetable/
Overdose - de NoizeField
https://www.noizefield.com/products/overdose

FM (Frecuencia Modulada)
En este tipo de síntesis, una onda moduladora influye sobre una onda portadora modificando su frecuencia lo que da lugar a una variedad de sonidos complejos con poco uso de recursos. A cambio de esa economía, los sonidos son más difíciles de dominar y pueden volverse inestables.
Otra característica importante en todas las síntesis que involucren la intervención de una onda sobre otra, es que los armónicos que las componen pueden ser múltiplos de la portadora o pueden no serlo. Esto da lugar a sonidos armónicos (con una nota definida) o inarmónicos (de nota idefinida, como un platillo por ejemplo).
El sintetizador más famoso de este tipo es el DX7 de Yamaha.
Podés encontrar un sintetizador virtual basado en el DX7 en este link:
https://github.com/asb2m10/dexed
Y para tus otras necesidades de FM tengo este Oxe FM Synth:

https://github.com/oxesoft/oxefmsynth

Los sintetizadores nos acompañan desde finales de la década de 1970. De ahí en más, todo ha sido experimentar y probar maneras de generar sonido (¡y mucho ruido!) por diferentes métodos. La llegada del audio digital supuso la simplificación de muchos obstáculos -tamaño y costo de los equipos, mantenimiento, cableado, complejidad e inestabilidad-. Las computadoras permitieron un modelado del audio mucho más minucioso y con la posibilidad de guardar parámetros, automatizar tareas y tener varios sintetizadores en un solo aparato.
Hoy día, muchas de las técnicas descriptas a lo largo de este artículo se pueden combinar en una única pieza de software. Podemos encontrar sintetizadores híbridos en la mayoría de los instrumentos VST.
Reconozco que debo haber dejado en el tintero algunas técnicas menos ortodoxas en pos de la brevedad del formato. Espero haber cumplido con la misión de acercar las más usadas y que al abrir un instrumento VST en tu próximo proyecto, puedas entender un poco mejor cómo funciona y qué parámetros se cruzan para lograr los sonidos que estás buscando.
¡Buena suerte y buenos patches!

hora_z

Texto-plano

==========

A:>

Una letra. Dos Puntos. Una rayita parpadeante. Esa era la presentación del Sistema Operativo de Disco, el mítico DOS.

DOS surge de la necesidad de poder interactuar con la computadora. Nos llama la atención y lo recalcamos no tanto por su similitud con UNIX o BSD, o por su pretensión de compatibilidad, ni por ser software libre, sino por contar con una CLI, una interfaz de línea de comandos.

En este sentido, uno escribe comandos u órdenes para interactuar con la computadora.

Podemos decir que la tipología de este tipo de sistema de control cobra su sentido si lo entendemos como parte de una EVOLUCION, una progresión mas o menos cierta en la manera de manejar la computadora. Me gusta decir que DOS es una familia de sistemas operativos, de la cual la más difundida es el DOS de Microsoft, o MS-DOS de 1982 compatible con la PC de IBM. Pero con anterioridad a este gran difusor, existieron desarrollos previos similares que le dieron inspiración. Uno de ellos, tan fundacional como olvidado, es el RT-11 de las minicomputadoras DEC PDP-11 de los 70s. Además de su logevidad, presentó un intérprete de comandos bastante capacitado (con archivos manejables, con nombres de 8 caracteres y 3 de extensión, etcétera). Y en las postrimerías de los 70s, un sistema operativo para microcomputadoras hobbistas evolucionadas: el CP/M, de Digital Research.

La aparición de los microprocesadores a principios de los 70s confiere una disminución muy marcada en el tamaño y costos de los equipamientos de cómputo. De mano de estas disminuciones, como si hubiesen sido el agua que necesitaba la semilla, se produce el progresivo despunte del CÓMPUTO HOBBISTA. Este giraba alrededor de microcomputadoras, máquinas de escasa potencia pero de un tamaño conveniente, similar al de una computadora de escritorio actual. Quienes se lanzaron a cultivar la novedad fueron los técnicos en radioelectrónica, aficionados talleristas, y todo tipo de personajes raros con ganas de "meter mano". Y debían tenerlas: los primeros microprocesadores programables carecían inicialmente de sistema de operación, y debían programarse accionando tediosamente interruptores para recibir información en base a ristras de luces de status. Sin embargo, ellos mismos se dieron a adaptarle conectores seriales que les pudieran permitir recurrir a los terminales impresores, las teletipos. Y con estas viejas veteranas máquinas electromecánicas -usadas anteriormente para transmisiones de télex- se

podía usar su teclado mecánico de entrada. Su medio de almacenamiento: ningún otro que las cintas de registro perforado.

En este contexto de máquinas de microcómputo armadas en base a plaquetas –con un microprocesador central, y muy escasa memoria– aparecen las primeras propuestas de software centralizado.
El objetivo era contar con un ambiente más o menos común y facilitado para operarlas.
Micro-Soft, una empresa recién fundada tras este objetivo, se propuso desarrollar versiones del lenguaje de programación BASIC para este tipo de tareas, habiendo desarrollado un intérprete capaz de correr en tan solo 4K de memoria. Una microcomputadora hobbista provista de un teletipo y el software del intérprete de lenguaje BASIC de Microsoft en cinta de papel se convertía en una máquina bastante más útil y operativa. Por lo pronto permitía escribir otros programas de manera mucho más concisa y simple que recurriendo al tedioso y dificultoso ensamblador.

Además de lo expuesto, la férrea intención de Micro-Soft era vender licencias de uso de las copias de su intérprete BASIC. La práctica en el ambiente hobbista había sido compartir los logros –incluidos las copias del código fuente de los programas– algo denostado por el presidente de Micro-Soft, William "Bill" Gates III.

Un nuevo y conveniente medio de almacenamiento aparece entonces, que se demuestra ideal para el micro-cómputo: el magnético. Este incluía la adaptación de interfaces de audio para utilizar cintas de casette para grabar información (de manera lenta y secuencial), y si bien su escaso costo las popularizó, otra tecnología magnética cobraría aún mayor relevancia para la computadora: se trató de los discos flexibles o "diskettes". Estos se iban a convertir en poco tiempo en los medios de almacenamiento favoritos de la época. Consistían en discos magnéticos que podían girar libremente en un estuche flexible cuadrado. Con un tamaño inicial de 8 pulgadas, serían luego reducidos a un formato tradicional apodado "mini-floppy" de 5 1/4".

El diskette permitía un acceso no secuencial mucho más cómodo y veloz a la información, a la vez que agilizar las funcionalidades de lectoescritura. Era obvio que el nuevo medio permitiría –a una microcomputadora dotada de diskettera– alcanzar incluso mayor desempeño que un ambiente BASIC.

La empresa Digital Research de Gary Killdal fue la que percibió este contexto. Se abocó entonces a desarrollar un "programa de Monitoreo y Control" (CP/M) portado en diskette, para microcomputadoras.
A pesar de su gran potencia comparativa, CP/M debía atender las distintas excentricidades que contaban los numerosos modelos de microcomputadoras. CP/M recurrió a un modelo de carga en memoria "personalizado" según el tipo de equipo, y un sistema de operación basado en disco inspirado –de alguna manera– en el efectivo DEC RT-11.

De esta manera, permitía contar con un sistema que se cargaba en memoria, y cargaba y ejecutaba software de terceros, algo notablemente inventivo y poderoso, pero sin requerir una máquina del porte de una minicomputadora PDP-11 sino una mucho más asequible microcomputadora (unas siete o cinco veces más barata).

Por controlar y difundirse en diskettes, podemos decir que CP/M es –en el sentido estricto del término– un sistema operativo de disco.
Incluia herramientas de BASIC, pero también editores y compiladores, junto con control de videoterminales.
Brilló por sus pequeñas aplicaciones de uso personal y de negocios.

Una revolución que germinó desde lo pequeño: microcomputadoras, microprocesadores, mini-floppyes y armándose bit a bit, byte a byte, kilobyte a kilobyte. El tejido social cooperativo era contagioso, y las compañías de software, pequeños emprendimientos personales o de equipos poco numerosos, se afianzaban mutuamente.

La industria de la electrónica y la microelectrónica apuesta, a veces a ciegas, a estos pequeños equipos que raramente sobrepasan los pocos miles de dólares. El rumbo a veces es nebuloso, pero se va definiendo a las microcomputadoras con pantallas y teclados, lectores de casette y

BASIC – en los modelos más básicos – y disketteras y CP/M con software de aplicación en aquellos de avanzada.

La IBM PC

IBM es "el Gigante Azul". Una compañía ya por entonces centenaria, dedicada de lleno al mundo de la informática corporativa, militar y de gobierno.

Se dice que no puede haber un trueno si nadie lo escucha, e IBM detecta esta nueva necesidad de usuarios de pequeños comercios, cuentapropistas, que cada vez más se acercan seducidos por las posibilidades de las maquinarias de marca monguito, y con CP/M. Si bien los considera por fuera de su marco natural de grandes empresas y emporios bancarios, lo cierto es que percibe que un cómputo de bajo nivel podría tener sentido, siempre y cuando sea el Gigante Azul quien hegemonice este nuevo tipo de informática comparativamente frugal.

Ante el cada vez mayor revuelo del microcómputo, IBM decide que también puede hacer una computadora de cariz personal sin que se le mueva demasiado un pelo. El único inconveniente es la necesidad de que esta cuente también con un Sistema Operativo de Disco. Mas vale recurrir a uno establecido, o que siga dicho esquema ya probado y favorecido por la pequeña clientela de estas pequeñas máquinas..

El verdadero énfasis en el mundo de los DOS viene traído de la mano de IBM y su intención de montar el pingo del cómputo compatible.

Es que las ahora llamadas "home computer" en tímida boga a principios de los 80s, estaban impulsando un negocio cada vez más apetitoso. Centenares de modelos, casi todos de 8 bits e incompatibles entre sí, constituian el firmamento a explotar. La lógica de la baja potencia impulsaba la disminución de precios.

Para inicios de 1981, IBM comprendió la seguridad de apostar a un terreno ya pisado por los hobbistas y los adeptos al cómputo de garage. En su apreciación, estos grupos variopintos jamás habían alcanzado una arquitectura informática compatible: cada uno de ellos – o bien las pequeñas empresas que habían despuntado – en realidad carecían del empuje necesario para imponer condiciones técnicas, políticas e industriales a los demás.

Curiosamente, el Gigante Azul – establecido como el alto clero de las Mainframes – también había sufrido del mismo problema. Sus distintas líneas de computadoras centrales estaban fuertemente compartimentadas. Si iba a dedicar entonces una minúscula parte de su Capital para dominar esta nueva religión de la compatibilidad en el microcómputo –que para ellos no era más que un pequeño sector del mercado– bien podría usarla como nueva práctica corporativa.

La apuesta segura de IBM involucró entonces desarrollar una estándar de arquitectura abierta al que los hobbistas y las pequeñas empresas de desarrollo tuviesen que atenerse de facto. Esto les permitiría retener el control de la máquina – por cuestiones de compatibilidad – junto con socios estratégicos. A esta arquitectura la denominiarían "IBM PC", la "Computadora Personal compatible de IBM".

Desde el punto de vista del software, IBM no puso demasiados recaudos. Consideró que existían ya estas pequeñas compañías con capacidad de organizar y montar el desarrollo de un ambiente de operación para la pequeña máquina. Se dirigiría a las renombradas en el "bajo clero" del cómputo: Micro-soft y Digital Research.

Creación de DOS

Existen varias versiones que giran sobre la manera en la que se fue dando la creación del sistema operativo de la IBM PC, pero para homogeneizar podemos sostener conclusiones en base al deseo de IBM de colocar en mercado una computadora de bajo costo destinada al hobbista y a los pequeños negocios, y quería hacerlo por fuera de los órganos de desarrollo principales de IBM (acostumbrados a grandes máquinas

corporativas).

A este fin, en finales de 1981 dispusieron en Boca Ratón (Florida) el Proyecto CHESS, un equipo de desarrollo con un presupuesto y staff comparativamente limitado, y rienda suelta. El sistema operativo que iba a servir a este proyecto no tenían intención de desarrollarlo in situ, sino que querían tercerizarlo, idealmente a Microsoft –y establecida como creadora de BASIC y compiladores para microcomputadoras– u otros.

Esto se motivaba en dificultades legales que IBM había experimentado con respecto al desarrollo de las ideas informáticas que hacían a sistemas operativos: a principios de los 80s y cada vez con mayor frecuencia, el desarrollo y publicación de programas de software corporativo se veía litigado por programadores hobbistas independientes que aducían ser los verdaderos idearios de código que IBM implementaba. Es que la trazabilidad de los proyectos de software –en la medida que estos se vuelven más complejos– se dificulta aún más. Esto implica mayores riesgos y dificultades al trazar su atribución.

Desde el punto de vista del hardware del equipo, sus especificaciones fueron concisas y realistamente planteadas por el Proyecto Chess, en base a las limitaciones y críticas de otras microcomputadoras:

* Microprocesador Intel 8088 de 16 bits.

* Placa madre con conjunto de chips limitado y de terceros. Ranuras de expansión estándar para placas de terceros.

*Chips de memoria RAM fabricados por terceros.

* Dos modelos distintos de Controlador de video a elección, con chip de terceros (Motorola 6845 en las placas de video monocromática MDA de sólo texto, y gráficos a color CGA.

* Dos disketteras de 5 1/4" Tandor, con controlador de terceros (NEC).

* Teclado IBM de alta calidad con microprocesador propio.

* BIOS propietaria de IBM de 8KB de ROM.

* Intérprete de BASIC M80 de Microsoft integrado en ROM de 32K.

Al consultar con Microsoft, la empresa adujo contar con un sistema operativo preparado y listo para un sistema de estas características, con lo que pudieron cerrar el trato de provisión en base a royalties por máquina vendida. La realidad es que Microsoft no tenía ningún sistema operativo.

Contactaron a una empresa de su misma ciudad, la Seattle Computer Products, disponía de un sistema operativo de circunstancias para una placa de desarrollo provista de microprocesador de 16 bits, el Intel 8086. La empresa consultó con Tim Paterson – su programador – y terminaron vendiéndole a Microsoft una licencia de uso no exclusiva por 50.000 dólares. El QDOS (Quick and Dirty Operating System, sistema operativo rápido y sucio) era capaz de hacer funcionar la placa de desarrollo, una diskettera, junto con una interfaz de terminal de video y monitor. Naturalmente, no pusieron al tanto que el verdadero cliente final era IBM. Contratando los servicios de consultoría de software de Paterson, Microsoft se dio a expandir las escasas capacidades del ahora llamado 86DOS, dando forma inicial a lo que sería el primer MS-DOS (Sistema Operativo de Disco de Microsoft).

EL MS-DOS tampoco tendría la santidad de sistema "exclusivo".
IBM apreciaba que la cantidad de paquetes de software semiprofesional existente ya para el sistema CP/M valía la pena, y si bien este operaba fundamentalmente en máquinas hobbistas con procesadores Zilog Z80, o Intel 8080 de 8 bits, lo cierto es que se esperaba que Digital Research pudiese adaptar y mejorar su CP/M al muy esperado CP/M-86 para el Intel 8086 de 16 bits que usaría la IBM PC. Con esto en vista iniciaron gestiones directas, pero por distintos motivos las mismas no llegaron a buen puerto. En primer lugar, el costo: Digital Research vendía CP/M prácticamente como un único producto integrado, y lo adaptaba a las

necesidades de múltiples fabricantes. Por tal motivo, los de Killdal desconfiaban ceder licenciamiento de modificación a IBM del tipo "licencia en blanco". Tampoco estuvieron de acuerdo en el precio propuesto, ya que el CP/M de 8 bits nunca se vendía a menos de 360 dólares por usuario, mientras que IBM pedía menos de un tercio de dicho valor "por máquina" (lo que supuestamente elevaría los costos nunca menos de una cifra por encima de los 400 dólares).

En cualquier caso, IBM quería lanzar su producto en góndola, lo cual denostaba cualquier derecho de exclusividad. Microsoft aproximó la idea de producir "su" sistema operativo a un costo de 60 dólares, para lo cual licenció 86DOS bajo la visa de "MS-DOS", junto a poderes de integración. Surge entonces el sistema operativo "oficial" de IBM PC, el IBM PC-DOS. La diferencia entre PC-DOS y MS-DOS venía dada por su lanzamiento seis meses adelantado, así como programación específica. Pero a pesar de estas ventajas, el de Microsoft terminó siendo el sistema operativo favorito de una clientela no tan dominada por IBM: el MS-DOS era 120 dólares más barato que el PC-DOS de IBM.

En sí, el DOS para la IBM PC era un sistema monousuario, monotarea, de 16 bits, que fue apareciendo con versiones progresivas en la medida que las PC se iban volviendo las máquinas favoritas de la industria.

En el caso de Microsoft, las versiones iniciales no revestían un sistema potente en lo absoluto. Las primeras versiones 1.x carecían de subdirectorios, y se limitaban a pedir un archivo binario y cargarlo del diskette a la memoria RAM y darle ejecución.

Aún así la PC fue un éxito en ventas entre hobbistas. La plataforma de hardware era muy buena, de alta calidad, y pronto las máquinas fueron dando lugar a programadores aficionados y profesionales por igual. Los programadores independientes hacían buen software y de bajo costo, y esto fue provocando paulatinamente un espiral de retroalimentación positiva: mayor oferta de software significaba mayor interés en la máquina, y esto, mayores ventas. Para 1983 la IBM PC era exitosa, y surge una versión expandida y más cuidada, la IBM PC XT "Tecnología Extendida", que fue realmente el producto estrella de la línea, destinado a la pequeña empresa.

La importancia de los discos rígidos (inicialmente, tecnología MFM de 20MB, luego 40MB y 80MB), hizo necesaria la integración de la primera versión importante de DOS, una que prácticamente hizo "borrón y cuenta nueva" definiendo parámetros mínimos de compatibilidad con el software anterior.
Se trató de la primer versión "moderna" de DOS, la MS-DOS 3.2.
Esta contaba con una shell COMMAND.COM que avalaba la posibilidad de usar subdirectorios arbolados, a la vez que hacía frente a nuevos desarrollos de hardware. El tope máximo de memoria que podía gestionar era de 640K, divididos en diez sectores de 64K de memoria (este era, originalmente, el tamaño máximo que se consideraba que podía ocupar un programa de DOS).

Para ese momento, surgen versiones de Digital Research que abandonan el paradigma CP/M-86, y mutan al DOS, conociéndoselos como DR-DOS.

El MS-DOS 3.2 es el arquetipo del viejo DOS, donde la máquina es un XT de IBM o una de sus ya cada vez más comunes "clones", provistos de dos disketteras de baja densidad (360K), un disco rígido de 40MB, y monitor monocromático y placa de video "Hércules", que permite gráficos y textos de "alta resolución". Un sistema potente como este, podía realmente oficiar de asistente en una oficina y pequeño negocio, como auxiliar de laboratorio, etcétera. Incorpora – además – cierta cosa UNIX, al permitir redireccionar salida y entrada con pipes.

La máquina tampoco fue sacrosanta. Su BIOS fue sometida a ingeniería inversa, y esto permitió la creación de clones. En 1986, máquinas como estas fueron la que inspiraron a centenares de programadores y empresas a dedicarse de lleno a la venta de software.

DOS ofrecía un "marco común" para el cual desarrollar –y una clientela interesante y normalmente ávida de novedades– y la PC una arquitectura

ideal. Los utilitarios potenciaban la capacidad de la PC con nuevas operaciones escapadas del estado de diseño original, pero no solo eso. Aparecían nuevos programas que llevaban las posibilidades de uso de la computadora a parámetros nunca antes vistos sobre máquinas de cómputo para el escritorio.

El procesador de texto WordPerfect fue ganando adeptos entre periodistas, escritores y abogados, lo mismo que el ambiente de base de datos DBase, un fijo para control de stock y aplicaciones de bancos de datos de todo tipo. También las planillas de cálculo, como la Lotus 1-2-3, que cambian rápidamente la manera de operar para todo tipo de acción científica y contable. Los nuevos ambientes de programación, como el Turbo Pascal, permitían generar código fuente y compilarlo para desarrollar utilidades de negocios destinadas a múltiples ramas del comercio.

Los paquetes de software se vendían con licencias de usuario final "para una sola PC", lo que hacía necesario comprar varias copias para distintas máquinas (algo raro, la mayoría de la gente solo tenía una PC).
En vistas de esta imposición estalla en grande la "piratería" del software, en cantidades inusitadas (el mismo sistema operativo era pirateado sin asco). También cobra popularidad el modelo del "Shareware": para evitar los costos logísticos de poner software en góndola, el autor y programador permitía la libre copia y difusión "mano en mano, diskette a diskette" de su programa.
Simplemente solicitaba "por honor", pagar un precio normalmente asequible si el programa "era del gusto del usuario que lo probó".

Los juegos fueron expandiéndose conforme se mejoraba el hardware. Para finales de la década surge la máquina AT, con procesadores más potentes, del 286 primero, y luego del modelo 386, de 32 bits.
Si bien DOS siempre fue un sistema de 16 bits, comenzó a expandir sus capacidades en torno a estos microprocesadores y sus memorias ampliadas. En primer lugar, para la versión MS-DOS 5.0 comenzó a aceptar nuevos esquemas de direccionamiento de memoria que iban por encima del miope límite de 640K impuesto por Bill Gates a principios de los 80s. En su lugar, se podían adaptar hasta 16MB de memoria utilizando controladores de software o software/hardware. Para finales de los 80s aparecen discos rígidos de 100MB, y las versiones de BIOS posibilitaron gestionar estas unidades y mayores (hasta un máximo de 2GB).
También permite integrar una pseudomultitarea gracias a los TRS (programas residentes, que quedan en memoria RAM hasta llamarlos nuevamente).

Para principios del 90 llega la "edad de oro" del DOS. Se transforma con mucho en el sistema operativo dominante, incluyendo las versiones PC-DOS y DR-DOS, que también gozan de cierto uso. Para este momento el DOS se encuentra sumamente depurado, y casi no presenta errores. Además, domina de tal manera el mercado que prácticamente ha eliminado el uso de terminales en el ambiente de negocios y de pequeñas empresas, puesto que la existencia generalizada de los clones hacen que las PC reduzcan su precio hasta un cuarto de los solicitados anteriormente, a la vez que potencian sus prestaciones, volviendo innecesarias a las minicomputadoras de antaño.

Los juegos de DOS extraen cada último resquicio de potencia del sistema operativo: recurren a la explotación de las placas de sonido, provistas tanto con sintetizadores de frecuencia modulada, como conversores DAC/ADC para sonido digitalizado de alta calidad. Los trucos de programación más ingeniosos afrontan las limitaciones originales del sistema. Los desarrolladores más imaginativos dan cuenta de innumerables éxitos en venta, la era de los "juegos clásicos del DOS".

Se distribuyen, se piratean, se crackean (rompen sus protecciones anticopia). Los programas van y vienen en diskettes, y por módem, gracias a las BBS, aquellas queridas redes hobbistas basadas en el teléfono de línea. Y con ellos los invitados de nadie: los virus informáticos.

MS-DOS sigue siendo un sistema operativo de interfaz de línea de comandos, pero a partir de 1992 Microsoft publica su primer éxito de su

anterior línea de interfases gráficas de usuario: Windows 3.1. Esta incorpora elementos faltantes que habían reducido las versiones anteriores a meros ambientes sin importancia. El uso de tipografías vectoriales escalables, ventanas rápidas a color y de resolución Super VGA, uso generalizado del mouse, además de la implementación multimedial, lo convierten rápidamente en el ambiente favorito para los programas de utilidad y oficina.

MS-DOS y sus competidores siguen lanzando versiones, integrando el uso de redes locales, o unidades ópticas de CD-ROM, de muy alta capacidad. Sin embargo, en el caso de Microsoft estas cesan en 1994 con MS-DOS 6.22, su última versión publicada. Los días de DOS están contados: con la aparición de procesadores de 32 bits cada vez más potentes - como el 486 y el Pentium - el sistema operativo de disco ha quedado desfasado.

Windows 95 incorpora una interfaz visual independiente como parte del mismo sistema operativo, y su éxito hace que de un día para otro prácticamente cese el desarrollo de utilitarios y programas para el ya obsoleto sistema operativo de disco.

El caso FreeDOS

Todo parecería concluido para este noble ambiente. Pero donde hubo llama, ceniza queda. Y con esto surge un nuevo DOS.
El sistema operativo de Microsoft y sus licenciatarios nunca fue "libre", en el sentido de que su código fuente siempre se guardó como "secreto industrial" por parte de una empresa que -durante su desarrollo- pasó de ser un galpón con veinte nerds difíciles de bañar a una de las compañías más poderosas del momento.

Sin embargo, para mediados de los 90s comienza a hacerse patente en el ambiente universitario estadounidense un nuevo sistema operativo, el GNU provisto de kernel Linux. Este tenía la característica de tener el código fuente de desarrollo publicado, y licenciado de manera tal que podía alterárselo, copiárselo, y usárselo como se quisiera, con la salvedad de tener que hacer públicas las modificaciones (siempre que se publicara a terceros).

Siguiendo estos principios de la Libertad del Software, el estudiante de física Jim Hall lanza en 1996 la primera parte de su Proyecto FreeDOS, cuyo objetivo es contar con un DOS de código libre y abierto.

En un principio el proyecto es unipersonal, pero gracias a Internet al poco tiempo va recibiendo colaboraciones y comienza a lanzar nuevas versiones.

El desarrollo de FreeDOS ha sido ejemplar, y si bien ha sido lento, Jim cuenta que no tiene ningún apuro, simplemente quiere seguir usando un ambiente DOS. La nueva propuesta de libre descarga puede conseguirse en freedos.org, y permite correr DOS incluso en una máquina moderna, con Internet (algo dificultoso de hacer con el antiguo DOS original, claro).

Hoy FreeDOS va por la versión 1.4, cuenta con gestor de paquetes gratuitos para DOS, permite correr la enorme mayoría de los antiguos programas de DOS (muchos de los cuales son funcionales perfectamente).

Pero FreeDOS no es la única reencarnación moderna de los Sistemas Operativos de Disco.

SvarDOS.org es un proyecto derivado de la tardía liberación del código fuente de MS-DOS 4.0, tal vez la versión menos exitosa de DOS. Aún así, este código fuente permitió hacer una distribución de DOS de dominio público, también con instalador de paquetes.

Conclusión

Los Sistemas Operativos de Disco fueron una parte importante de la historia del cómputo personal, y base importante de desarrollo. Como tal, contaron con buenos e ingeniosos programas que -partiendo del ambiente hobbista- llegó a profesionalizar la manera de escribir y

usar aplicaciones de software.

Muchísima paquetería constituye hoy "software abandonado", pero no son pocos los ejemplos que han pasado a manos del software libre, puesto que sus autores han comprendido la necesidad –y diversión– de liberar su viejo código fuente para permitir usarlos nuevamente, ya sea en ambientes similares, o portados a otros diferentes.

El hecho de contar hoy con versiones de software libre de este ambiente es un caso paradigmático e importante que subraya la importancia del software libre no sólo para el uso de aplicaciones, sino también para el arte de la conservación informática.

---

10 | PRIVACIDAD

Imagina que sales a la calle para ir al trabajo o simplemente salir a comprar. Hay una anciana intentando cruzar la calle y la ayudas, pero ella en vez de agradecerte, te empieza a hacer preguntas todo el rato, súper personales, como preguntarte si te has casado, si tienes un departamento en propiedad, a dónde fuiste el finde. Luego en el almacén o el kiosco saludas al encargado y el tipo en vez de saludarte te dice "ah ya sabía que venías, la vieja me conto ¿qué compraste en la farmacia?".

Claramente acá hay dos personas que se están desubicando mucho al meterse en tu vida personal intentando y consiguiendo meterse en los detalles de tu día a día. Esto es lo mismo que pasa continuamente al usar tu teléfono móvil o cada vez que compras en el súper, el problema es que no nos damos cuenta, pero como dice la ensayista y periodista española Marta Peirano, vivimos en un capitalismo de vigilancia donde generamos mucha información y esta tiene mucho valor, sobre todo agregada o individualizada y no porque haya un secreto muy delicado para la seguridad interior de algún país.

El ser humano tiene una necesidad básica de privacidad que va más allá de hacer algo malo. No le puedes contar toda tu vida a tu tía, tampoco lo tiene que saber el verdulero o el supermercado de la otra cuadra. Pareciera sin embargo, que eso es más sencillo de entender porque es visible y no en el caso de Facebook o Instagram donde todo va a una base de datos que no entendemos cómo funciona. Algo que me llama poderosamente la atención es que ni siquiera aceptan un pago a cambio de la publicidad que te muestran. No podría asegurarlo sin datos, pero probablemente valga más tener un conjunto de usuarios expuestos a publicidad que lo que cada uno pudiera pagar por eliminarlos.

No sólo las GAFAM (Google, Amazon, Facebook, Apple, Microsoft) acumulan datos personales para su beneficio, también lo hacen empresas locales más pequeñas, tanto para mejorar sus ventas como para obtener un ingreso extra vendiendo esa información a terceros.

¿Y ahora qué hacemos?
¿Nos vamos de todas las redes y dejamos el teléfono móvil en casa?
No sólo es preocupante que las empresas acumulen datos para ellas sino que esa información (que muchas veces no sabes exactamente cuál es) podría caer en malas manos, ser cruzada con otras fuentes o como dice Peirano, identificar a terceros a través de tu información.
Por eso es importante ser precavidos.

---

11 | Cielo Azul ~emilio

Nos comprometemos a luchar contra la obsesión por
por los cielos azules allí donde los encontremos.

– Lema de la Sociedad Internacional

de Observadores de Nubes.

Esa tiene forma de Oso Hormiguero. ¡Mira! esa otra parece un Camaleón. A Raquel le encantaba observar las nubes, subía a la azotea y aprovechaba el poco tiempo libre antes de irse a la fábrica, era una agradable rutina que diriamente practiba, era de esas cosas que la consolaba, ver nubes le recordaba su pueblo y su niñez.

Un día Raquel bajó de la azotea sumamente contrariada, las nubes habían desaparecido por completo. Toda esa semana y la otra y la otra intentó observar alguna nube pero ¡nada!, todo el tiempo un cielo azul.

Las nubes habían desaparecido igual que su niñez, sus hermanos y su pueblo, de un día a otro habían desaparecido dejando un enorme vacío.

Raquel descubrió a través de su teléfono que en distintos países también estaban procupados por la total desaparición de las nubes, nadie sabía qué ocurría.

En algunos medios se anunciaba el suicidio colectivo de pilotos de Globos Aerostáticos que no soportaban el insípido vuelo en un cielo completamente azul.

"Han dejado ver el cielo, ya no buscan figuras entre las nubes, nadie anhela o sueña con volar", afirmaba uno de los especialistas en pedagogía en su artículo "Los niños con un cielo azul".

"¿Dónde están ahora los niños? atrapados en el automóvil... ¿dónde están sus ojos? encerrados en la pantalla del celular, ya no se les decir... ¡miren! esa nube parece un dragón", agregaba el especialista en el artículo.

El gran edificio de cristal creaba un efecto de transparencia al reflejar un inmenso cielo azul, dentro se encontraban inversionistas y representantes del gobierno que escuchaban atentos la conferencia del CEO de Cloud Inc.

"Nuestro principal objetivo es el bienestar de la sociedad, por ello gracias a esta nueva concesión podremos llevar alimentos a los más necesitados.

En el siguiente video pueden apreciar nuestra tecnología de extracción del agua de las Nubes. Con tecnología satelital localizamos la formación de masas de Nubes, nuestros drones recolectan información detallada in situ y con el uso de Inteligencia Artificial seleccionamos las mejores zonas de recolecta y extracción de Nubes.

Transportamos la maquinaria extractora que ubicamos debajo de los cúmulos de nubes seleccionadas previamente y absorbemos todo su volumen en agua y demás gases ricos en nutrientes.

Podemos absorver más de 10,000 nubes diariamente y comprimirlas en contenedores del tamaño de pelotas de fútbol que podemos enviar a las granjas agrícolas y sustentables de Cloud Inc. donde se liberará el contenido de la Nube para producir lluvia y con ello abastecer a la población de alimentos. También distribumos las Nubes a clientes que requieren agua para procesos de refrigeración de sus granjas computacionales.

Agradezco al gobierno por permitirnos ser parte de su progreso y desarrollo, aprovecho para mencionar que la Fundación Cloud Inc. apoyará la educación de los nños instalando pantallas informativas sobre los distintos tipos

de nubes que conforman nuestro catálogo: Cúmulus, Stratos, Cirros y otras varientes. Y por último hemos realizado un acuerdo con las principales empresas de videojuegos y producir el juego pedagógico y gratuito "La Nube Exploradora" donde las niñas y niños fomentarán su creatividad buscando formas de animales o cosas en las nubes que se van a ir presentando en las pantallas de sus celulares o tabletas electrónicas.

Nuestro interés es participar en la construcción de un país más fuerte y democrático".

...

"¡Compañeros hoy nos reunimos aquí ante esta megacorporación y ante ese gobierno corrupto para mostrar nuestra indignación, porque no puede ser posible y es injusto que se roben las nubes, esas nubes pertenecen a todos los seres vivos que habitamos el planeta y no a unos cuantos humanos que las roban y las concentran en su propio beneficio!"

Gritaba una de las personas que participaba en la manifestación multitudinaria frente al gran edificio de Cloud Inc.

Raquel se enteró de la convocatoria y decidió no ir a la fábrica y presentarse en la manifestación de más de 5,000 personas entre ancianos, niños, adolescentes que exigían la no extración de nubes en cualquier parte del mundo.

Como protesta final se elevaron miles de papalotes y pancartas preguntando ¿Dónde están las Nubes?.

Preocupados por la seguridad nacional y la estabilidad económica el gobierno envío una solución inmediata y contundente...

Un grupo de drónes rodearon a los manifestantes y dispararon densas nubes de un color azul intenso.

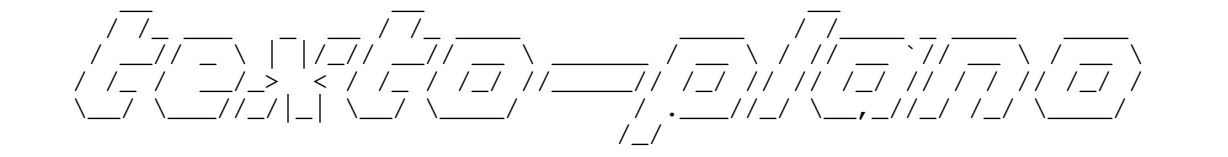
Raquel como pudo apenas y llegó a su casa donde perdió el conocimiento, al otro día al despertarse descubrió que había perdido la vista.

Al fondo se escuchaba el reporte meteorológico del noticiero: "Tendremos otro fin de semana con un clima agradablemente soleado y completamente despejado un perfecto cielo azul".

FIN

Texto-plano

Revista digital 01 | 12



Texto-Plano es una comunidad en un servidor en la que puedes participar si te interesa utilizar la consola en un servidor compartido. Tal y como decía, Ken Thompson, Unix es un sistema alrededor del cual se puede formar una comunidad. Algo así como un hacklab pero totalmente digital. Aquí creamos contenido, probamos scripts y aprendemos a usar la consola, conversamos de la vida... No somos un hosting de uso privado.

Tenemos seis principios

1. Minimalismo
2. Fomento de la curiosidad
3. Comunidad
4. Prioridad a las actividades en el servidor
5. No comercial
6. En español

Si te interesa, puedes enviarnos tu clave SSH ed25519 a admin@texto-plano.xyz explicándonos qué te motiva. (Lee el código de conducta).

Si deseas hablar con nosotros y conocernos antes de participar o quieres contactarnos si eres parte de otra similar, estamos en:

Jabber:
general@conference.texto-plano.xyz
IRC: #texto-plano en irc.tilde.chat

En una futura edición hablaremos más en detalle de qué es un pubnix, qué es un tilde, su historia, etc. Así que esperen nuestra siguiente edición.

~ffuentes