

TempleOS

Alejandro Rubio Martínez, Luis Crespo Orti y Ángel Olmedo Navarro

1. Introducción

Construir un sistema operativo es una tarea complicada, que normalmente requiere de un gran equipo detrás y millones de dólares de presupuesto. Sin embargo, Terry A. Davis se propuso hacer uno por su cuenta tras sufrir una revelación divina en 2003, y tras 10 años de trabajo y unas 100.000 líneas de código, lo consiguió.

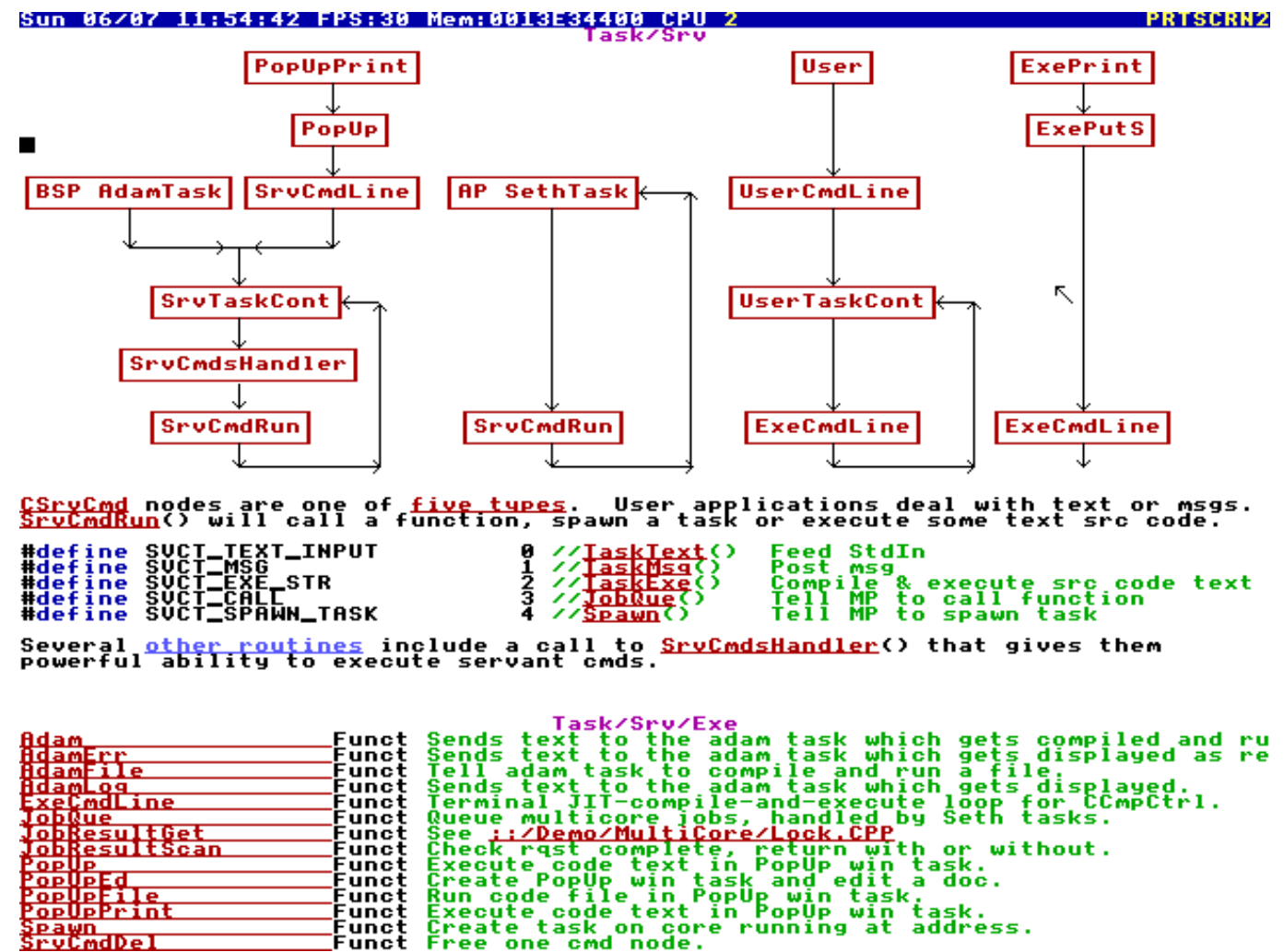
El sistema operativo que creó se llama TempleOS, y es, de hecho, una especie de simulación del Tercer Templo profetizado en La Biblia. A continuación vamos a proceder a explicar cómo funciona internamente.

2. Características técnicas

- Se trata de un SO con arquitectura de 64 bits diseñado mediante un lenguaje de programación llamado HolyC, una mezcla entre C y C++, desarrollado por el propio Davis únicamente con este propósito. Es más, la Shell tan solo interpreta comandos en este lenguaje.
- El sistema entero es de código abierto ([Github de TempleOS](#)) y dominio público, es decir, quien quiera puede descargarlo y utilizarlo (es recomendable hacerlo en una máquina virtual).
- Utiliza la política de multitarea cooperativa, es decir, no es el propio sistema el que inicia los cambios de contexto, sino que son los propios procesos los que ceden el control a otros, ya sea periódicamente, cuando se bloquean, quedan inactivos, etc. Además es multinúcleo, luego permite ejecutar los procesos en varios procesadores.
- No utiliza espacios de usuario, pues está pensado para ser monousuario. No existen los permisos de archivo ni la protección de memoria. Davis pensaba que otros sistemas como Linux consumen demasiados recursos en estas tareas, las cuales no son necesarias para el uso común que se le da al sistema. Por tanto, el diseño del SO es de "anillo cero": da completa libertad al usuario para interactuar con el hardware (las llamadas al sistema se ejecutan directamente como funciones normales).
- En cuanto a paralelismo de procesos, no existen mecanismos como los threads de Linux. Si se crea un proceso, y se quieren compartir sus datos con otro, se tendrá que hacer manualmente, aunque resulta muy sencillo en este SO.
- En lo relativo a los gráficos, el sistema muestra ASCII de 8 bits con gráficos en el código fuente, además de una biblioteca de gráficos 2D y 3D basada en Scratch, que se ejecuta únicamente a 640x480 VGA con 16 colores.
- Implementa un sistema de hipertexto llamado DoIDoc (creado también por Davis) que sustituye a HTML, XML, JSON, shell scripts... Éste permite la inclusión de imágenes, mallas 3D e hipervínculos que activan macros, entre otros recursos, directamente en la shell, el editor de texto, la documentación, el explorador de archivos... Este aspecto es uno de los más innovadores.
- Soporta los sistemas de archivos ISO 9660, FAT32 y RedSea (creado también por Davis). De hecho, permite el montaje de discos, ya sean CDs, discos duros, etc.
- La única salida de audio soportada es el altavoz del PC (no se pueden conectar auriculares u otros elementos). Permite la entrada a través de ratón y teclado.
- Por último, cabe destacar que no admite conexión a la red.

3. Diagrama de funcionamiento interno

A continuación, exponemos el diagrama de la organización interna del SO, el cual es accesible introduciendo un comando desde la propia Shell:



Podemos ver 3 esquemas distintos:

1. El de la izquierda se refiere a la creación de ventanas. En resumidas cuentas, lo que nos dice es que primero se llama a `PopUpPrint`, que crea una instancia de `PopUp`, la cual genera un proceso que hereda de Adam (el equivalente al `init` de Linux). Posteriormente se llama la función `SrvCmdRun` para invocar a la función que crea la nueva ventana.
2. En el del medio podemos ver `SethTask`, que se refiere al proceso inicial de cada CPU. Por ejemplo, Adam es su propio Seth, en el `core0`. En este esquema se indica como el Seth en todo momento es el encargado de gestionar la CPU en la que se encuentra independientemente de sus hijos.
3. El de la derecha nos indica como el usuario realiza una acción en el sistema. Primero este le hace una petición al sistema (no confundir con llamadas al sistema), la cual se va a traducir a algo que entienda nuestro SO, y se comunicará cuando termine de procesarse.

Visto lo anterior, hemos concluido que TempleOS se asemeja a un microkernel, tal y como se ha señalado en clase. Esta conclusión la extraemos al observar cómo todos los procesos son planificados por el Seth y además, se traducen las peticiones del usuario a una capa más interna que entienda el sistema operativo, principalmente a través de la llamada `SrvCmdRun`.

4. Objetivos y utilidades

Como ya hemos comentado, según Terry, el objetivo principal del SO es el de crear el Tercer Templo que se refleja en las sagradas escrituras católicas, tal y como Dios le encomendó en su revelación. El sistema está lleno, por tanto, de herramientas y utilidades que ayudan a la extensión de la palabra de Dios por el mundo.

Sin embargo, es obvio que este proyecto también tiene una gran componente recreativa, lúdica e incluso didáctica. TempleOS es, en muchos sentidos, un terreno de pruebas e innovación en el ámbito de los sistemas operativos. Prueba de ello es el hecho de que no esté basado en ningún otro: está íntegramente construido desde cero y todos los mecanismos propios del sistema son originales.

La interfaz de TempleOS es, básicamente, una shell desde la cual podemos acceder a los elementos del sistema, ejecutar aplicaciones, activar funcionalidades... Está organizada de una forma algo anti-intuitiva para el usuario medio. Su estética recuerda a la de los ordenadores Commodore 64 de la década de los 80.

Tenemos disponibles atajos de teclado básicos que encontraríamos en cualquier sistema, tales como:

- **CTRL+M** para ir al menú personal.
- **F1** para acceder al manual de ayuda.
- **SHIFT+ESC** para abortar tarea y salir.

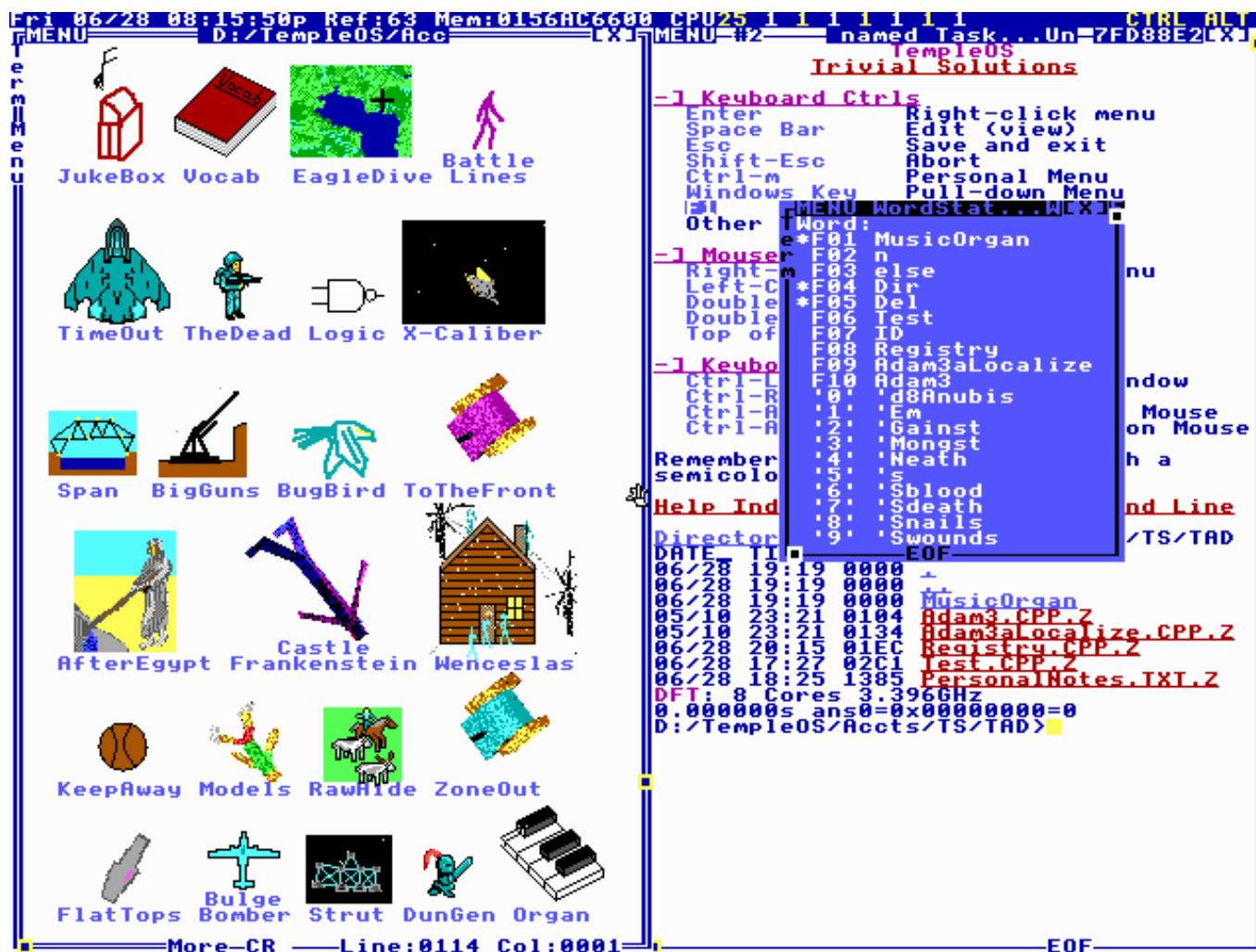
Y otros algo más peculiares:

- **CTRL+R** nos permite realizar dibujos que podremos insertar más tarde en documentos.
- **F6** reproduce un sonido celestial.
- **F7** escribe en la interfaz una palabra del diccionario del SO.
- **CTRL+ALT+B** nos da acceso a una Biblia interactiva.

El SO incluye por defecto algunos juegos y aplicaciones. No todos, pero muchos de ellos incluyen elementos de temática religiosa. Por ejemplo, tenemos el simulador de vuelo **Titanium** y una aplicación llamada **GodBoodle**, que genera dibujos sencillos, con el objetivo de poder ilustrar el mensaje de Dios (especialmente dirigido a los niños).

El juego estrella implementado por el creador se llama **AfterEgypt**. Este no se encuentra instalado por defecto en TempleOS, pero sí se puede ejecutar a través del montaje de un disco, al estar el SO preparado para ello. Se trata de un simulador de Moisés en el Monte Sinaí, que recibe mensajes de Dios generados automáticamente mediante una función pseudo-aleatoria que mezcla frases y palabras tomadas de la Biblia, entre otros diccionarios. El algoritmo para esta función también lo creó Terry, y en muchas ocasiones los mensajes tienen sentido, lo cual denota su complejidad.

Por supuesto, además de todo esto, se pueden realizar tareas básicas como escritura de código, lectura de archivos, ejecución de programas implementados por el propio usuario, etc, desde una o varias ventanas. Todo ello utilizando la shell como centro de operaciones.



5. Conclusión

Para empezar, queremos aclarar que hemos elegido TempleOS en nuestro trabajo para mostrar hasta donde podría llegar el propósito de un Sistema Operativo, y probar que no es necesario basarse en sistemas comunes como Windows, MacOS o Linux para construir uno de ellos.

Si bien es cierto que hay otros sistemas que son más funcionales y tienen mayor utilidad, TempleOS construye un entorno de programación muy sencillo para principiantes. De hecho, Terry anima a los usuarios a desarrollar sus juegos dentro del propio sistema. Para él, esa tarea resulta más divertida que jugarlos.

Muchas personas no prestan atención a este sistema porque lo ven como un delirio de un loco, pero la realidad va mucho más allá. Hay bastante trasfondo y es interesante realizar un estudio sobre este SO.

6. Curiosidades

- La motivación de su hazaña fue, según el autor, una "revelación" que tuvo con Dios. Sin embargo, antes de ello, se consideraba ateo.
- En un ámbito más personal, mencionar que Terry fue diagnosticado con trastorno bipolar y posteriormente con esquizofrenia tras frecuentes ataques psicóticos. Incluso, estuvo internado en hospitales psiquiátricos. Esto no le impidió culminar su proyecto.
- Terry se ganó el respeto entre muchos usuarios de la comunidad informática tras lanzar TempleOS. A día de hoy, es un ídolo y prácticamente una leyenda para muchos aficionados a este mundo.

- La muerte de Davis en 2018 asestó un duro golpe a la comunidad. Su familia pidió a los contribuyentes del proyecto que redirigiesen sus donaciones a asociaciones que luchen contra los trastornos mentales. Aun así, la comunidad sigue dando soporte al sistema de forma altruista.



7. Bibliografía

- <https://templeos.org>
- <https://www.diariotecnologia.es/templeos-una-rareza-de-sistema-operativo-tan-denostado-como-revolucionario>
- <https://www.youtube.com/watch?v=LtlyeDAJR7A>
- <https://www.youtube.com/watch?v=qstjWeW6UEU>
- <https://templeos.holyc.xyz/Wb/Doc/>
- <http://www.codersnotes.com/notes/a-constructive-look-at-templeos/>
- <https://es.wikipedia.org/wiki/TempleOS>
- <https://www.adslzone.net/2015/07/23/templeos-el-sistema-operativo-escrito-por-una-solo-persona-porque-dios-se-lo-pidio/>