

## Un vistazo atrás en la Historia,a los Orígenes del OS/400

*Este artículo revisa algunos de las decisiones de diseño que se tomaron tempranamente en el desarrollo del S/38. Esta historia del principio es importante porque gran parte de la arquitectura del OS/400 fue heredada del S/38.*

Hace unos 20 años, cuando el S/38 estaba siendo desarrollado, el mundo nunca había oído hablar de Microsoft o Netscape. Pero la lucha entre el gigante de las computadoras IBM y sus competidores para conseguir cuota de mercado era tan feroz como las batallas de hoy en día acerca del próximo navegador (browser). La Corporación IBM había pasado por la experiencia de la pérdida de ventas de hardware por “parecer igual ” a otros vendedores como Amdahl y otros. Estos clones de hardware podían ejecutar el sistema operativo de IBM y los clientes obtenían equivalente potencia de computadora a un menor coste. La venta de hardware representaba una gran fuente de ingresos para IBM por lo tanto esta pérdida de ventas de hardware a los vendedores de clónicos era un negocio muy serio para los ejecutivos de IBM.

En el gran frente de los mainframe , IBM acababa de cancelar un proyecto conocido como FS (Future Systems) que iba a ser un sustituto revolucionario para la línea existente de computadoras mainframe de IBM. Después de varios años de trabajo, reuniones y más reuniones y multitud de papeles con especificaciones, la dirección de IBM determinó que el proyecto era demasiado ambicioso incluso para IBM y abandonó el proyecto en 1975.

En el corazón de los campos de trigo de Rochester, Minnesota el S/3 y los siguientes sistemas S/32 y S/34 estaba casi al final de una vida exitosa. El momento para un recambio se acercaba. No existía ningún clon de hardware del S/34 pero el temor de un clónico que reemplazara el hardware existente era muy alto entre las preocupaciones de la dirección de IBM. Las noticias de la defunción del FS estaban llegando lentamente a Rochester así que el grupo de planificadores , ingenieros y programadores continuaron con su acercamiento independiente para el diseño de un sistema para el futuro , al menos para los clientes pequeños y medianos . Cada uno lo quería más grande, mejor, más rápido y más barato que el envejecido entonces S/34 pero desconocían como conseguirlo.

A principio de 1975, yo me incorporé al grupo base , con menos de 25 personas, que estaba desarrollando la visión del recambio del S/34. Este pequeño grupo tenía gente de talento con variados conocimientos. Algunos venían de los equipos de diseño del S/3, S/32 y S/34 mientras otros tenían un importante pasado con el S/360 y el S/370 .

---

Un vistazo atrás en la Historia,a los Orígenes del OS/400

© Wayne O.Evans , e-mail: WOEvans@aol.com

## Interface de máquina abstracto

Cuando me incorporé al proyecto, se estaba discutiendo una idea revolucionaria. Las palabras “objetos”, “encapsulación” y máquina abstracta eran nuevos para la mayoría de programadores de sistema, incluido yo. En vez de tener un conjunto de instrucciones de la computadora diseñadas por los ingenieros, los programadores (los usuarios reales del sistema) estaban definiendo instrucciones para una máquina abstracta. Estas instrucciones en vez de ser optimizadas para el hardware, estaban siendo diseñadas para escribir aplicaciones de software. Este interface de máquina abstracto sería conocido como MI (Machine Interface) o lo que IBM normalmente llama TIMI (technology Independent Machine Interface). Este Interface Abstracto (MI) definió algunos conceptos muy avanzados para hacer la programación más eficiente y menos propensa a errores. Los programadores escribirían programas para una máquina abstracta utilizando nuevos conceptos como la no carga de datos en los registros del hardware para su proceso. En esta nueva máquina abstracta, los programadores mejorarían la fiabilidad de los programas, al prevenir que los programas pudieran desde modificar el almacenamiento hasta manipular las direcciones de los registros. El uso de programas MI podía manipular los punteros, pero la máquina abstracta no permitiría un cambio que pudiera causar la corrupción de un puntero. Este interface abstracto estaba siendo diseñado para proteger a los programadores de ellos mismos. En el hardware previo, los programadores conocían la estructura de direcciones de los punteros y podían incluso alterar la memoria para cambiar las direcciones (lo cual a menudo era el motivo de los problemas de fiabilidad de los programas). En esta máquina abstracta, estos punteros estaban encapsulados (el contenido real del puntero no se presentaba en las aplicaciones) lo que significaba que los programas no podían crear un puntero al modificar un almacenamiento y sólo instrucciones de puntero podían usarse para modificar punteros. Cualquier intento del programador por alterar el almacenamiento que contenía un puntero podría causar que el puntero se volviera inválido. Estos punteros de la máquina abstracta eran mayores que cualquier registro de hardware y el microcódigo traduciría el uso de los punteros a los registros reales de hardware.

La separación de la máquina real de la máquina abstracta es la responsable de la migración con éxito del hardware CISC al hardware RISC. Otras plataformas, están ocupadas reescribiendo los programas para conseguir direcciones de 32 o 64 bit (algunas plataformas están planificando completarlo en el 2005). Lo que es verdaderamente fascinante es que cada aplicación del AS/400 está capacitada para 64 bit sin ninguna modificación para el cliente ni los programas OS/400.

Un segundo concepto importante de este interface abstracto era el mover muchas de las funciones del sistema operativo (gestión de tareas, seguridad e incluso la base de datos) al microcódigo. Puesto que estas funciones son

Un vistazo atrás en la Historia, a los Orígenes del OS/400

© Wayne O.Evans, e-mail: WOEvans@aol.com

# Wayne O Evans Consulting, Inc.

AS/400 Security Consulting and Education

implementadas muy cerca del interface de hardware, deberían ser programadas eficientemente. Más importante para IBM era que esa integración de la función del sistema operativo en el microcódigo del hardware hiciera más difícil el copiar el hardware del S/38 . Este hecho por sí solo aseguraba que el proyecto del S/38 obtendría la financiación de la Dirección de IBM .

Hoy, nuestra experiencia con el AS/400 prueba que este concepto de máquina abstracta tiene mucho sentido , pero para los programadores de sistema de 1970s, esta idea era radical.

---

## EL GRAN DEBATE

Esta idea de un interface de máquina abstracto tenía también sus críticos. Había varios profesionales técnicos muy cualificados que argumentaban (con gran pasión ) que un concepto como el del interface abstracto podía ser académicamente posible pero podría provocar que la ejecución sería pobre. Mucho esfuerzo (y emociones fuertes) se estaban dedicando a discutir y aprobar y desaprobar la actuación del hardware directo o las implementaciones de microcódigo.

Recuerdo el día que el asunto se resolvió. Se celebró una reunión del equipo técnico clave una fría mañana de invierno en Rochester, MN. Los partidarios de la máquina abstracta y sus adversarios fueron convocados en la sala de conferencias de Glen Henry, el director de programación del sistema. Glen era un individuo extremadamente inteligente pero como otras personas verdaderamente dotadas carecía de habilidades para la interacción humana. Cuando Glen Henry veía la respuesta obvia te lo decía con pocas palabras y sin añadir ninguna dulzura. Glen proclamó el final del debate . Con gran emoción explicó que el interface de la máquina abstracta (después se conocería como MI) era la única posibilidad. Glen explicó como esto protegería a IBM de vendedores de hardware clónico. Concluyó con “o se acepta el programa o estaré encantado de firmar el traspaso de cualquiera a otra área.”

Podemos ahora mirar atrás y ver que este interface ha sido una de las ventajas más significativas de la arquitectura del AS/400. La separación del software del hardware permitió el cambio radical de hardware sin modificar el software. Los usuarios del AS/400 fueron capaces de realizar la transición desde CISC a RISC sin ninguna interrupción en sus aplicaciones , debido al interface de máquina abstracto.

---

## El Porque los nombre del AS/400 son de longitud 10 caracteres

Una vez tomada la decisión sobre la máquina abstracta, se pudieron tomar las decisiones de diseño menos críticas . Como esta era una nueva máquina, todos los parámetros de diseño fueron cuestionados. Una de las decisiones que tuvieron que tomarse fue la longitud de los nombre de objeto. Una propuesta era un nombre de 8-caracteres que sería compatible con el S/34. Otros propusieron

Un vistazo atrás en la Historia, a los Orígenes del OS/400

© Wayne O.Evans , e-mail: WOEvans@aol.com

## Wayne O Evans Consulting, Inc. AS/400 Security Consulting and Education

“nombres de documentación propios” de hasta 32 caracteres soportados por el MI. La decisión se tomó cuando el equipo de diseño de la base de datos determinó que necesitaban poder almacenar el fichero , la biblioteca y el nombre del miembro en los 32 caracteres que permitía el MI. El nombre de 10-caracteres se determinó al dividir la longitud de nombre máxima (32) por 3 así que los nombres del S/38 (y los nombres AS/400 ) están limitados a 10 caracteres de longitud.

---

### ***La Selección de la denominación de los comandos.***

La consistencia y estructura del lenguaje de comandos CL es bien reconocida. La estructura de objetos verbo de los nombres de comando resultó de problemas que se descubrieron durante la primera fase del desarrollo. Los comandos eran desarrollados por equipos individualizados de desarrolladores. Ellos llamaban sus comandos de forma inconsistente y esta “política de terminología” necesitaba unas convenciones muy estrictas respecto a los nombres de objetos verbo y utilizar entonces abreviaturas comunes por todo el sistema operativo. Estas convenciones de nomenclatura ya consistentes fueron extendidas desde el sistema operativo para incluir también a los Licensed Program Products. Una estructura de objetos verbo no fue suficiente porque aparentemente había una dificultad en reconocer las separaciones entre abreviaturas , así que los estándares se fueron extendiendo a abreviaturas de 3 caracteres, con la excepción de la última palabra en el nombre del comando. Miles de usuarios del AS/400 encuentran que el lenguaje CL es fácil de aprender y usar, gracias al esfuerzo en la política de terminología para la selección e imposición de estándares en el nombre de los comandos.

### **Nace el Prompter**

Los test de usabilidad mostraban que los usuarios no podían recordar todos los nombre de comando y palabras clave y el nombre de los valores. Los esfuerzos iniciales eran para construir pantallas individuales para cada comando y hacerlas consistentes . Como el número de comandos crecía, era claro que IBM no podía permitirse el coste de interfaces creados individualmente para cada pantalla, así que entonces los planificadores de IBM intentaron priorizar los comandos, de forma que había pantallas para algunos , y no para otros. Con el número de comandos que iba creciendo cada día, el departamento de interface de usuario no podía desarrollar todas estas pantallas o incluso mantenerlas, así que propusieron automatizar la generación de esas pantallas basada en los objetos de definición de comandos almacenados.

El resultado final fue el revolucionario comando prompter, el cual se ha probado inestimable para más de 1500 comandos de IBM pero también disponible para los comandos del cliente. Utilizar el símbolo de interrogación delante de los

Un vistazo atrás en la Historia, a los Orígenes del OS/400

© Wayne O.Evans , e-mail: WOEvans@aol.com

# Wayne O Evans Consulting, Inc.

## AS/400 Security Consulting and Education

nombres de comando provee una forma sencilla para los programas CL de buscar el input.

---

### **Revisión del Sistema.**

Pronto el proyecto FS se canceló, y la dirección de IBM empezó a centrarse en los esfuerzos de desarrollo de Rochester. La pregunta del día era cómo podía este pequeño grupo en los campos de Minnesota ( conocido en código como Pacific) tener la esperanza de cumplir lo que los equipos de diseño masivos del este y del oeste no podían. Se llevó a cabo una revisión del sistema para determinar el estatus del proyecto Pacific . El futuro del proyecto Pacific dependía de los resultados de esta revisión del sistema. Los técnicos que lideraban el proyecto FS revisaron el diseño del hardware y del software . Todos los esfuerzos en Rochester se centraron en contar una buena historia sobre nuestro progreso. Se prepararon y explicaron planes detallados del hardware y del software para el equipo de revisión. En alguna ocasión la tinta no estaba seca en el último diseño cuando se presentaba de hecho al equipo de revisión. Fue un tiempo de gran stress porque el equipo de diseño conocía que el destino del sistema estaba en las manos de ese equipo. El proyecto sobrevivió a la revisión y el S/38 y AS/400 son el resultado.

Esa revisión tuvo un papel importante en solidificar el diseño del sistema operativo. Antes de la revisión, los diseñadores cambiaban sus diseños “mejorándolos” por semanas. La revisión del sistema forzó al equipo de diseño a solidificar y documentar las decisiones de diseño.

---

### **Retos de la Implementación**

El equipo de ingeniería y el grupo de programación empezó a crecer rápidamente después del éxito de la revisión de sistema. Los recursos de IBM en el emplazamiento de Rochester no podían soportar a todas esas personas, así que el proyecto Pacific al completo ( ingeniería y programación) se trasladó a un gran almacén abandonado en Rochester. Esto mantuvo a los grupos de diseño de hardware y de software juntos y permitía a los desarrolladores con sólo caminar un pasillo hablar con los diseñadores de áreas específicas. Yo creo que esta facilidad de comunicación fue una de las razones por las que el proyecto en Rochester tuvo éxito donde los intentos de multi-centros para implementar FS habían fallado.

Como muestra la Figura 1, había 3 equipos de diseño (hardware, microcódigo, y sistema operativo) todos centrados en el diseño y la implementación. Estos equipos de diseño casi representaban la arquitectura de la máquina. La comunicación entre estos 3 grupos diferentes era excelente aunque a menudo pensabas que estabas en una torre de babel. Los distintos grupos utilizaban el mismo término de proceso de datos ( como “cola”) para describir un concepto similar pero cuyos detalles de función era enormemente diferentes. Este conflicto de terminología podía ser muy confuso porque cuando los diseñadores

Un vistazo atrás en la Historia, a los Orígenes del OS/400

© Wayne O.Evans , e-mail: WOEvans@aol.com

CPF Sistema Operativo

MI – Machine Interface  
Abstract Machine

VMC Vertical MicroCódigo

HMC Hardware MicroCódigo

Figura 1. Capas de Implementación

de diferentes áreas hablaban, cada uno pensaba que lo había entendido , pero a menudo había una gran diferencia en lo que cada uno había comprendido. Cuando los programadores del sistema operativo e ingenieros hablaban, necesitaban a menudo algún individuo que pudiera servir como traductor y explicar las sutiles diferencias.

## Simulador

Se procedía al desarrollo en paralelo del hardware, del microcódigo y del sistema operativo .En las fases iniciales del proyecto, no había hardware disponible para el equipo de programación. Se desarrolló en el S/370 un simulador de las instrucciones de hardware de la máquina .Éste permitía a los desarrolladores de microcódigo diseñar y probar sus funciones en ausencia de un hardware físico. La simulación de las instrucciones de hardware era lenta al principio. Yo era el que lideraba el desarrollo del analizador de comandos y para la simple comprobación de la sintaxis, un comando CL utilizando el simulador tardaba entre 30-45 minutos si las computadoras no estaban sobrecargadas. Si había múltiples simulaciones ejecutándose, el mismo test requería hasta 2 horas. Como resultado, muchos otros y yo mismo, empezamos a trabajar fuera de horas para obtener mejores tiempos de respuesta (Si puedes llamar una mejor respuesta a una comprobación de sintaxis que tarda 30 minutos ).Con frecuencia los programas fallaban y entonces el reto era determinar si era tu código, un bug de microcódigo o un problema con el simulador. El lenguaje de programación que se usaba para desarrollar el sistema operativo era el PL/MI, un derivado del PL/I. Había un lenguaje similar al PL/S que desarrollaba código para el S/370 así que nuestro equipo de desarrollo utilizaba el S/370 para hacer gran parte del programa de debug del analizador de comandos . Esto era esencial puesto que el analizador de comandos CL tenía que hacerse pronto para que otros equipos de desarrollo pudieran crear comandos individuales en el sistema operativo S/38 llamado Control Program Facility (CPF).

El analizador de comandos del S/38 tiene la característica única de que tanto el sistema operativo como las instalaciones de usuario usan las mismas

Un vistazo atrás en la Historia, a los Orígenes del OS/400

© Wayne O.Evans , e-mail: WOEvans@aol.com

# Wayne O Evans Consulting, Inc.

## AS/400 Security Consulting and Education

características para crear comandos para el sistema operativo S/38 y ahora para el AS/400. Como se usaron las mismas características para tanto el sistema operativo como comandos definidos por el usuario , esto significa que estos comandos de usuario tienen las mismas características que los comandos del sistema operativo.

---

### **Probando el Hardware**

Finalmente, la ingeniería ofreció algún hardware disponible y empezó un nuevo reto. Las herramientas de código paso a paso en el simulador , no existían en el hardware así que tenías que aprender un nuevo método completo de “debug” de un programa. Las herramientas de debug no existían durante las fases iniciales así que la configuración de las paradas en las direcciones de hardware se convirtió en el método para testear tus programas.

El programa más utilizado en el sistema era una utilidad llamada la copia 2 a 1 . Todo el microcódigo y el sistema operativo cabía en un simple disco (Piccolo drive). El primer paso era una copia de 15-30 minutos del sistema completo de la unidad 2 a la unidad 1. Entonces añadías tus programas y si tenías suerte podías añadir tus programas en la unidad uno y reiniciar el sistema de forma que entonces podías copiar la unidad 1 a la unidad 2. A menudo el sistema se “colgaba” y la única alternativa era empezar de nuevo ejecutando la copia de 2 a 1 otra vez. La copia de unidades de disco estaba funcionando el 80 % del tiempo y sólo el 20 % se utilizaba para un test real.

Cada semana los desarrolladores conseguían un nuevo driver con las últimas reparaciones de la función del microcódigo. Las funciones de restauración no funcionaban así que el equipo de construcción del driver unía una unidad de disco a tu sistema, conectaba los cables y ejecutaba la famosa copia 2 a 1 aunque eso requería una nueva carga del sistema completo. Entonces necesitabas añadir todo tipo de programas de test al sistema. Cada semana el sistema se volvía más estable.

---

### **ITSWITCH salva el día**

El programa más util aparte de la copia 2 a 1 era una herramienta llamada ITSWITCH. Si no hubiera sido por este programa, el sistema operativo S/38, CPF nunca se hubiera completado. El programa tenía muchas funciones, la configuración de la tabla de puntos de entrada y determinaba la fecha y hora de compilación de los programas. La función más crítica era la capacidad para atrapar un error inesperado y permitir al operador actuar manualmente para resolverlo. Había dos terminales adjuntados a cada sistema, uno se usaba para testing, y el otro para correr el ITSWITCH.

Como se hizo pronto el código del analizador de comandos, yo pude cambiar al desarrollo de una demostración del funcionamiento del sistema. Esta se convirtió en la demostración de la función del S/38 que se utilizó en el primer anuncio en

Un vistazo atrás en la Historia, a los Orígenes del OS/400

© Wayne O.Evans , e-mail: WOEvans@aol.com

## Wayne O Evans Consulting, Inc. AS/400 Security Consulting and Education

1978. La demostración era una aplicación simple de entrada de pedidos que actualizaba la base de datos y mostraba los resultados. Para los estándares de hoy, era una aplicación muy simple pero que durante el desarrollo del CPF era una gran aplicación. Imagine ejecutando múltiples terminales todos accediendo al mismo fichero y permitiendo la actualización de los datos de ese fichero. Todo estaba cuidadosamente programado para asegurar que la cantidad del pedido no podía reducir nunca la cantidad en stock a menos de 0 , pues podrían ocurrir circunstancias inesperadas. La demostración parecía muy impresionante pero era muy frágil. El sistema tenía una impresora asociada pero si intentabas imprimir y realizar el pedido a la vez el sistema se podía colgar. La demostración fue así convenientemente ensayada para mostrar a los usuarios introducir el pedido y entonces mover a la audiencia a la impresora para que pudieran mirar el output (mientras ningún pedido se introducía). Todo ese tiempo, el ITSWITCH se monitorizaba detrás del escenario para controlar un fallo potencial del sistema.

---

### **S/38 Perdió**

Esta aplicación ayudó a convencer a la Dirección de que el S/38 estaba preparado para presentarse antes de que estuviera preparado realmente. Como resultado, se decidió que el sistema estaba listo para su presentación y entonces la fecha de entrega de un primer cliente se cambió porque la fiabilidad del sistema no tenía unos estándares mínimos. Fue un “black eye” para el equipo de desarrollo pero un retraso esencial porque los clientes no disponían de ITSWITCH para mantener sus sistemas funcionando.

Mirando atrás en el día del anuncio del S/38 , había muchas utilidades elegantes como la base de datos integrada, subficheros y CL comandos a definir por el usuario, y un lenguaje de CL que podía compilarse en un programa. Pero también había otras características que faltaban como por ejemplo: no se añadió la capacidad para las comunicaciones hasta la release 2.

---

### **Defectos del diseño.**

#### CL

Uno de los errores de diseño que había cometido el equipo de CL , apareció durante el desarrollo de la CPF release 2. Si IBM añadía una clave a un comando CL que ya existía, cada programa que usaba ese comando debía recompilarse . Esto motivó un rediseño del método que utilizaban los programas CL compilados, que requería que los clientes recompilaran todos los programas CL cuando la release 2 estuviera disponible. Había un gran interés en el S/38 pero muchos clientes no habían materializado su decisión de compra sólo porque algunos “early adopters” habían sufrido el impacto de ser obligados a la recompilación de programas.

---

Un vistazo atrás en la Historia, a los Orígenes del OS/400  
© Wayne O.Evans , e-mail: WOEvans@aol.com



# Wayne O Evans Consulting, Inc.

## AS/400 Security Consulting and Education

### Copy File Interactive

En la primera release del CPF, IBM empaquetó lo que no sería llamado un asistente para simplificar la función de copia de ficheros. La función interactiva de copia de ficheros instaba al usuario a una copia de los ficheros de comandos y basada en la respuesta de los usuarios a estos avisos, le presentaba otros avisos. Fue implementado como múltiples comandos CL que incitarían al usuario a respuestas y simplificarían el comando CPFY . Como el número de opciones crecía , esta alternativa se hizo imposible , así que CPYFINT (copy file interactive) fue retirada en la versión 2.

---

### Memoria Principal.

El coste de la memoria principal era tan importante que el S/38 no tenía los tamaños tan grandes de la memoria como soporta hoy el AS/400. Los primeros sistemas de hardware tenían 512 K memoria y existía un esfuerzo muy importante de diseño para conseguir el punto de entrada de diseño por debajo de 256 K de memoria. La actividad del S/38 y AS/400 se mejoró significativamente con la adición del almacenamiento principal. Es duro considerar ahora que los primeros sistemas podían correr (lentamente) en 512K.

### El Porqué los Mensaje empiezan con CPF

Los usuarios del AS/400 pueden no reconocer que el nombre del Sistema Operativo del S/38 era Control Program Facility (CPF). Esta abreviatura se utilizó en todos los mensajes cuyos programas de usuario incluían la monitorización de los mensajes. Resultó que el número de mensajes requeridos era más grande que los que se habían previsto, así que se añadieron variaciones utilizando las 2 letras iniciales CP de la abreviatura. La próxima vez que tenga un mensaje CPF desde el AS/400 ,reconocerá los esfuerzos iniciales de un equipo de diseño del S/38 con talento que dejó al AS/400 una valiosa herencia.

### CONCLUSION

Hubo mucha prensa alrededor del éxito en la conversión del S/36 y S/38 en un tiempo récord para conseguir el AS/400. En mi punto de vista, los retos técnicos encarados por los primeros desarrolladores del S/38 fueron mayores que aquellos que se enfrentaron con la conversión desde el S/38 al AS/400. La mayoría de reglas de diseño fundamentales para la arquitectura fueron ya establecidas por el S/38 y el funcionamiento del AS/400 pudo ser desarrollado y probado en el hardware del S/38 existente.

---

This is a work in progress and I hope to expand this with more stories from others that worked on the S/38 development. June 26, 2000

Please contact me if you have a story to share. Wayne O. Evans WOEvans@aol.com

Un vistazo atrás en la Historia, a los Orígenes del OS/400

© Wayne O.Evans , e-mail: WOEvans@aol.com

Traducción autorizada al Español realizada por <http://www.recursos-as400.com>