

F3S: Sistema de ficheros para Windows CE a prueba de fallos

Artículo cedido por Venco Electrónica



xxvª Aniversario 1983-2008

Para más información:
www.vencoel.com
info@vencoel.com
Oficinas
Barcelona: Tel. +34 93 2633354
Fax. +34 93 2633323
Madrid: Tel. +34 91 3295500
Fax. +34 91 3295925
Durango: Tel. +34 94 6232655
Fax. +34 946202185
Porto: Tel. +351 225191386
Fax. +351 225191389



¿Por qué este sistema de ficheros?

El manejo de datos de forma consistente y segura es indispensable en muchas de las áreas donde se utilizan los sistemas embedded, como por ejemplo en las aplicaciones médicas. El sistema operativo de Microsoft Windows CE incluye el sistema de archivos FAT para el almacenamiento de datos de forma persistente. Sin embargo, se ha demostrado que en muchos de los casos el sistema FAT no puede gestionar este requerimiento de forma suficiente. Con bastante frecuencia el responsable de un fallo en el sistema o malfuncionamiento es producido por datos corruptos.

Por esta razón, F&S ha decidido desarrollar un nuevo sistema de ficheros a prueba de fallos Failsafe Flash File System (F3S). Este sistema está especialmente diseñado para utilizarlo en sistemas embedded. Utiliza el concepto de transacción, especialmente ante interrupciones eléctricas. F3S es capaz no sólo de garantizar que la información no será dañada, tal y como lo hace el sistema FAT, sino que garantiza la seguridad de transacciones a nivel de fichero. La modificación de los datos en el fichero no se llevará a cabo hasta que la operación haya finalizado completamente. El punto de finalización puede ser definido en la aplicación. El caso más sencillo es utilizar un manejador de fichero. Si la operación es abortada, el estado anterior a la modificación se restaurará de forma automática. Esto ofrece la posibilidad de guardar los datos de una forma persistente y razonablemente consistente.

Contrariamente a los sistemas de ficheros convencionales del Windows CE, F3S está dirigida a aprovechar las características de memoria flash NAND. Por lo tanto, el número de operaciones de escri-

tura física se reducen al mínimo. Esto principalmente mejora la tasa de transferencia en operaciones de escritura de manera espectacular. Al mismo tiempo, esto reduce el uso de memoria flash y, por tanto, aumenta su vida útil, la cual es limitada. Además, F3S está equipado con un recolector de basura muy eficiente que resuelve el problema de la fragmentación. F3S está implementado en C++. Como resultado de la misma arquitectura modular es portable a otras plataformas muy fácilmente. Por lo tanto, está disponible por el momento para todos las Single Board Computer de la familia NetDCU (WindowsCE 4.2/5/6) de F & S. El sistema de ficheros F3S se mantiene y mejora continuamente. Además F & S ofrece un soporte cualificado y orientado al cliente.

FAT para el almacenamiento de datos de forma persistente. Especialmente en entornos donde el sistema es sensible a la información, es indispensable un manejo de datos seguro y consistente. Sin embargo, ha sido demostrado en muchos casos, que el sistema de ficheros FAT no puede asegurar este requerimiento. Con bastante frecuencia datos corruptos son los responsables de un malfuncionamiento y fallos en el sistema. Debido a esto F&S ha decidido desarrollar un nuevo sistema de ficheros basado en transacciones que sea aplicado en diversos sistemas, esto es lo que se conoce como Failsafe Flash File System (F3S)

El sistema de ficheros FAT utiliza una tabla de asignación de datos o contenido de los directorios a su posición en el medio



Descripción

La compañía F&S Elektronik Systeme GmbH, distribuida en España por VENCO ELECTRONICA, se ha especializado en desarrollar PCs Embedded utilizados para controlar displays gráficos.

Estos SBS están equipados con un Windows CE y por lo tanto sólo incluyen el sistema de archivos

de almacenamiento. Esto permite una sencilla búsqueda de datos. Pero después de que se produzca un fallo, mientras una operación está en curso, siempre existe el peligro de que los datos sean dañados. Incluso si la escritura es un bloque de datos atómico, la mayoría de las operaciones, como la creación o eliminación de un archivo, están consumiendo varias

operaciones de escritura física. Si el fallo de energía eléctrica se produce mientras realiza una operación de escritura, los datos del archivo se dañarán casi seguro. Si mientras se modifica la tabla de asignación se produce un fallo, archivos completos o estructuras de directorios pueden perderse. Por lo tanto, la meta del desarrollo fue sistema de ficheros que aseguren la integridad de los datos y garanticen también una protección eficaz a nivel de archivo.

Convencionalmente los sistemas de ficheros modifican datos almacenados en su ubicación original. Si un bloque de datos que contiene el archivo o meta-datos se ha modificado, esta acción se llevará a cabo en la misma ubicación física. Debido a la restricción del acceso de las memorias flash, esto no es posible directamente. He aquí la necesidad de una capa de software adicional, implementada entre el controlador de flash y el sistema de ficheros – a esto se le denomina Flash Abstraction Layer (FAL) o Flash Translation Layer (FTL) (véase Figura 1). Esto permite que el sistema de archivos para acceder a la memoria flash como un dispositivo de bloque. La FTL realiza la llamada mapeada en bloques. Las operaciones de lectura y escritura se basan en direccionamiento lógico de bloques. El tamaño de un bloque de datos normalmente se corresponde con el tamaño de una página flash. La escritura de un bloque de datos modificados no cambiará la celda actual de almacenamiento físico. En lugar de eso, el bloque de datos modificados se escribe en una ubicación física libre. Posteriormente, se realiza el mapeo interior entre la dirección lógica y física.

El hecho de que cada página de flash no se pueda sobrescribir directamente, apoya el desarrollo de la operación de transacción a nivel de fichero, los archivos de datos antiguos y los nuevos se almacenan por separado en flash para un corto período de tiempo. Además, cada página de una memoria flash incluye una zona especial (spare-page) que ofrece la posibilidad de almacenar la información de datos en la memoria flash directamente. Para establecer un nivel de archivo seguro, es muy útil guardar el archivo de cada una de las páginas en flash directamente. Basado en estas notas se ha desarrollado el sistema de ficheros F3S.

- Todos los búferes de ficheros deberían ser vaciados: **FlushFileBuffers()**
- Si el flag **WRITE_THROUGH** se ha activado y la operación se ha finalizado

Si el sistema falla antes de llegar a alguna de estas condiciones, modificación de datos, pérdidas o el original del contenido de fichero original recuperado. F3S es sistema de ficheros estructurado en Log. Lleva el diario, conocido a partir de NTFS o Ext3 / 4 hasta el extremo. Bloques de datos almacenados en la memoria flash no se asigna a las direcciones de bloques lógicos, sino que se asignan directamente al

correspondiente sistema de ficheros objeto (archivo o directorio). Esto lleva a dos ventajas fundamentales frente a los convencionales sistemas de archivos:

- Las características físicas de la memoria flash pueden ser utilizadas directamente.
- Las pérdidas de datos o incluso de estructuras de directorios pueden ser evitados si no se asignan tablas o estructuras en memoria flash.

F3S no almacena la estructura de directorios

en una estructura central, como los sistemas de archivos para los discos duros, ya que esto generalmente se traduce en problemas de incoherencia. En lugar de ello, utiliza el spare-area de la página en particular de la flash para almacenar toda información necesaria. La estructura de directorio exacta se copia a la estructura de datos relacionados en la memoria principal. Hay un objeto para cada objeto del sistema de archivos, lo cual indica que el directorio en el que reside actualmente. Varios objetos

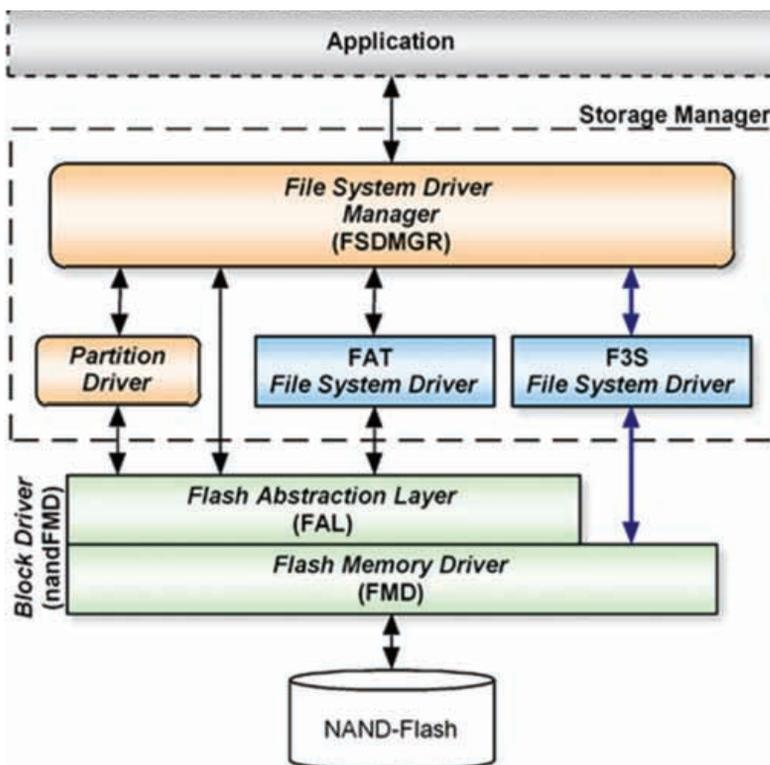


Figura 1. Acceso a la memoria flash en WindowsCE

en el mismo nivel jerárquico están organizados como una lista. Con la ayuda de esta estructura, cada objeto del sistema de archivos se puede encontrar mediante la ruta. Como la relación de un objeto, o más precisamente una referencia al mayor directorio clasificado objeto, se almacena en la spare-page, la estructura correcta del directorio puede ser reconstruida en cualquier momento. Y debido a la atomicidad de escribir una página, la integridad de la estructura de directorios se puede garantizar en cada situación.

Incluso para garantizar la coherencia de los datos del archivo, F3S tiene características adicionales. En una petición de transacción, un bloque especial de datos se escribirá el archivo que valida todas las modificaciones anteriores. Por lo tanto, una transición de los datos del archivo modificado consiste en una operación atómica. Cualquier operación

interrumpida será detectada la próxima vez que monte la partición, y el último estado válido se recuperarán automáticamente.

Se evita deliberadamente el uso de una tabla para almacenar en la flash. Cada bloque por tanto se almacena en una estructura de árbol binario con el correspondiente objeto del sistema de archivos en memoria principal y se establecerá al mismo tiempo a la estructura de directorios durante el proceso de montaje. A diferencia de la FAT, el mapeado de bloques de datos no se basa en direccionamiento de bloques lógico. En lugar de página en flash se relaciona con el contenido del fichero directamente. Por lo tanto el mapeado en una parte inherente del sistema de archivos. Refiriéndose a la transacción segura, esto es esencialmente un beneficio

ya que permite sistemáticamente una cache a nivel de fichero. Los bloques de datos en cache son identificados directamente.

Como resultado de la arquitectura de WindowsCE, el sistema de archivos que son especialmente desarrollados para memorias flash no puede ser adoptados directamente. El FAL es una parte inherente del driver del bloque de memoria flash para restringir el acceso físico a la memoria. Algunas pequeñas adaptaciones son inevitables. Ejecución de la FAL,

parte, se analiza periódicamente la flash para detectar fragmentación. Por otra parte, la GC se iniciará si el espacio libre disponible no es suficiente para terminar la actual operación de escritura. Sólo hay en liberación de espacio de almacenamiento cuando es realmente necesario mantener la latencia lo más corto posible. Este proceso no afecta a la seguridad de las transacciones de datos de no ser cambiado.

Debido al proceso de envejecimiento de las células de memoria flash, la duración de estas memorias es limitada. El fabricante especifica un número máximo de ciclos de borrado de bloque de flash. Para utilizar eficazmente la vida, se aplica un mecanismo denominado Wear-leveling. Por ejemplo F3S utiliza el bloque que menos veces se ha utilizado.

Como se mencionó anteriormente, las características

de la memoria flash son abordadas. En contraste con los sistemas de ficheros convencionales, el número de accesos de escritura se reduce al mínimo, lo cual principalmente mejora considerablemente la tasa de transferencia de lectura y escritura, además esto reduce el uso de la memoria flash y en consecuencia aumenta la vida útil de la flash.

F3S está implementado en C++. Como resultado de la misma arquitectura modular es portable a otras plataformas muy fácilmente. Por lo tanto, está disponible por el momento para todos las Single Board Computer de la familia NetDCU (WindowsCE 4.2/5/6) de F & S. El sistema de ficheros F3S se mantiene y mejora continuamente. Además VENCO ofrece un soporte cualificado y orientado a las necesidades del cliente. ■

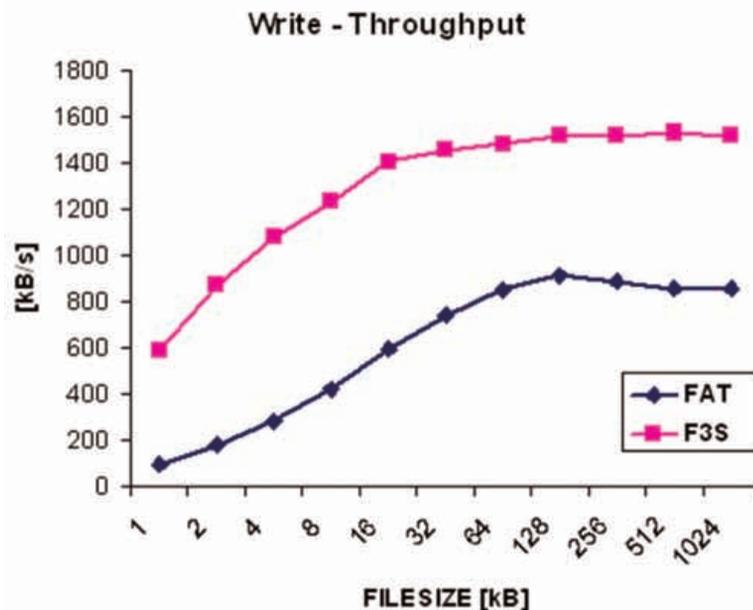


Figura 2. Operaciones de escritura

además de un sistema de archivos flash se consigue mediante un flag especial en la spare-page, lo que hace que la FAL ignore la relación con las páginas de uso. Cuando se declara la información almacenada en esta área, esta compatibilidad debe tenerse en cuenta.

En el transcurso del tiempo y como consecuencia de las características de las memorias flash NAND, surgen algunos efectos de fragmentación en varios bloques de flash. Para trabajar en contra de estos efectos, se utilizar lo que se denomina recolector de basura (Garbage-Collector GC) que realiza una desfragmentación en caso de de ser necesario. En este proceso, los datos válidos se copian a zonas libres creando bloques que después pueden ser reescritos. El CG utilizado en el F3S tiene dos modos de funcionamiento. Por una