ПОДДЕРЖКА ДЛИННЫХ ИМЕН ФАЙЛОВ ДЛЯ "BARE DOS" НА ОСНОВЕ НЕДОКУМЕНТИРОВАННЫХ ФУНКЦИЙ

Маргулис М.Б., Мацак С.А.
Кафедра ЭВМ ДГТУ

Abstract

Margulis M.B., Matsak S.A. Long file names for "bare DOS". *Since Windows95 come out users have had ability to use long file names. This article tells how Windows9x store long file names and contents a description of methods how write a driver to work with them in "bare DOS".*

Введение

В течение многих лет пользователи DOS и Windows испытывали неудобства, связанные с тем, что имена файлов не могли иметь длину более восьми символов плюс три символа на расширение. В операционных системах для Macintosh, в OS/2 и Windows NT длинные имена LFN (Long File Name) были предусмотрены гораздо раньше. С выходом Windows 95, такую возможность получила файловая система VFAT (Virtual File Allocation Table).

Структура записи для хранения на диске файлов с именами в формате 8.3 хорошо документирована. Об особенностях же хранения длинных имен сообщается очень мало.

Несмотря на то, что существует встроенная поддержка LFN в DOS 7 в виде набора функций, они доступны лишь при запущенном драйвере IFSMgr Другими словами: поддержка LFN обеспечивается только в оболочке Windows.

В практике работы с компьютером и, особенно, при ликвидации нестандартных ситуаций, возникает необходимость произвести операции над файлами или каталогами, без активизации Windows - в однозадачном режиме. Здесь предлагаются способы реализации такой поддержки LFN.

1. Формат хранения длинных имен файлов в Windows 9x

В FAT каждому файлу соответствует 32-байтный элемент каталога, содержащий имя файла, его атрибуты (скрытый, системный, только для чтения и т.д.), дату и время, а также прочую информацию. Прикладные программы обращаются к ОС за именами файлов и каталогов не путем прямого считывания с диска соответствующих записей, а через специальные, встроенные в ОС функции (за исключением специальных утилит, использующих для обращения к диску операции низкого уровня - типа Norton Disk Doctor).

Совместимость файловых систем VFAT и FAT обеспечивается тем, что для каждого файла и каталога имеются два имени: короткое, "понятное" всем прикладным программам, и длинное - для приложений Windows 9x и тех программ, в которых предусмотрена возможность работы с длинными именами. Для хранения коротких имен в формате 8.3 используются обычные 32-байтные записи. Короткие имена Windows создает из длинных имен, отсекая шесть старших символов и добавляя в конце этого базового имени "~1". Если же существует еще одно имя, состоящее из тех же шести символов, то этот номер увеличивается на единицу. Расширение файла сохраняется прежним.

Длинные имена хранятся в специально отформатированных 32-байтных записях, байт атрибутов у которых равен 0Fh (это нереальное сочетание атрибутов). Положение поля атрибутов в записях каталога как для LFN, так и формата 8.3 одинаково. Объясняется это тем, что, до тех пор пока файловая система не ознакомится с

содержимым байта атрибутов, она не "знает", с каким типом записи она имеет дело в данный момент.

Для конкретного файла или подкаталога непосредственно перед его единственной записью каталога с его именем в формате 8.3 находится группа из одной или нескольких записей, представляющих длинное имя. Каждая такая запись содержит часть длинного имени файла не более 13-ти символов, т.е. всего может существовать 20 порций имени (255 div 13).

LFN хранятся в формате Unicode, т. е. для каждого символа выделяется 2 байта. Если запись каталога с последней порцией длинного имени занята не полностью, сразу за последним символом имени должен следовать завершающий ноль, далее позиции заполняются кодом FFh.

Длинное и короткое имя файла связаны между собой двумя способами: во-первых, короткое имя следует непосредственно за длинным именем, во-вторых, в каждой порции длинного имени присутствует контрольная сумма короткого имени. В Windows 9x контрольная сумма используется для выявления "осиротевших" или испорченных записей в каталоге для длинных имен.

Формирование записи для длинного имени происходит даже в том случае, если оно достаточно коротко и годится для формата 8.3. Структура элемента каталога для длинного имени показана в приложении 1.

2. Набор функций для работы DOS-приложений с LFN

Для работы с LFN в DOS 7 предназначены функции 71xxh прерывания 21h [1]:

```
INT 21 71-- - Windows95 - LONG FILENAME FUNCTIONS
   AH = 71h
   AL = function
      0Dh reset drive (see AX=710Dh)
      39h create directory (see AX=7139h)
      3Ah remove directory (see AX=713Ah)
и т.д.
```

Эти функции в основном являются аналогами “старых” функций прерывания 21h для работы с файлами (AH=39h, AH=3Ah, AH=3Bh ...).

3. Реализация поддержки LFN для “bare DOS”

Возможно несколько путей для решения задачи обеспечения поддержки LFN.

Первый и наиболее простой метод - это создание (использование) некой программы менеджера файлов или нескольких отдельных программ с собственной реализацией алгоритмов чтения информации с носителей. Недостатком такой реализации является ограниченность функций и невозможность использования других программ в том числе и внутренних команд DOS, с поддержкой длинных имен.

Исследования показали, что в MS DOS 7x при обработке каждой команды проверяется поддержка функций 71xxh прерывания 21h. В случае их наличия выполняются процедуры с поддержкой LFN, иначе вызываются “старые” функции. Таким образом, логично предположить, что если реализовать обработчики функций 71xxh прерывания 21h, которые отсутствуют в “bare DOS”, то таким способом обеспечивается работоспособность внутренних команд DOS и других программ с поддержкой LFN. В этом и заключается второй метод, основные идеи реализации которого представлена здесь.

Поставленная задача достаточно актуальна, т.к. несмотря на то, что имеются попытки разработки этой темы некоторыми авторами (Приложение 3), в настоящий момент не существует работоспособного алгоритмического и программного обеспечения этой задачи.

Для создания обработчиков функций 71xxh прерывания 21h существуют различные способы. Можно организовать функции работы с файлами (открытия, создания и т.д.) путем чтения "напрямую" с дисков, т.е. используя только прерывания 13h или 25h/26h (absolute disk read, absolute disk write). Однако, в этом случае необходимо правильно заполнять множество внутренних таблиц DOS, что затруднено отсутствием соответствующей документации.

Для того чтобы обойти эту проблему при вызове функций 71xxh прерывания 21h, можно вызывать "старые" функций прерывания 21h (работающие с короткими именами). При этом нужно осуществлять необходимые преобразования параметров.

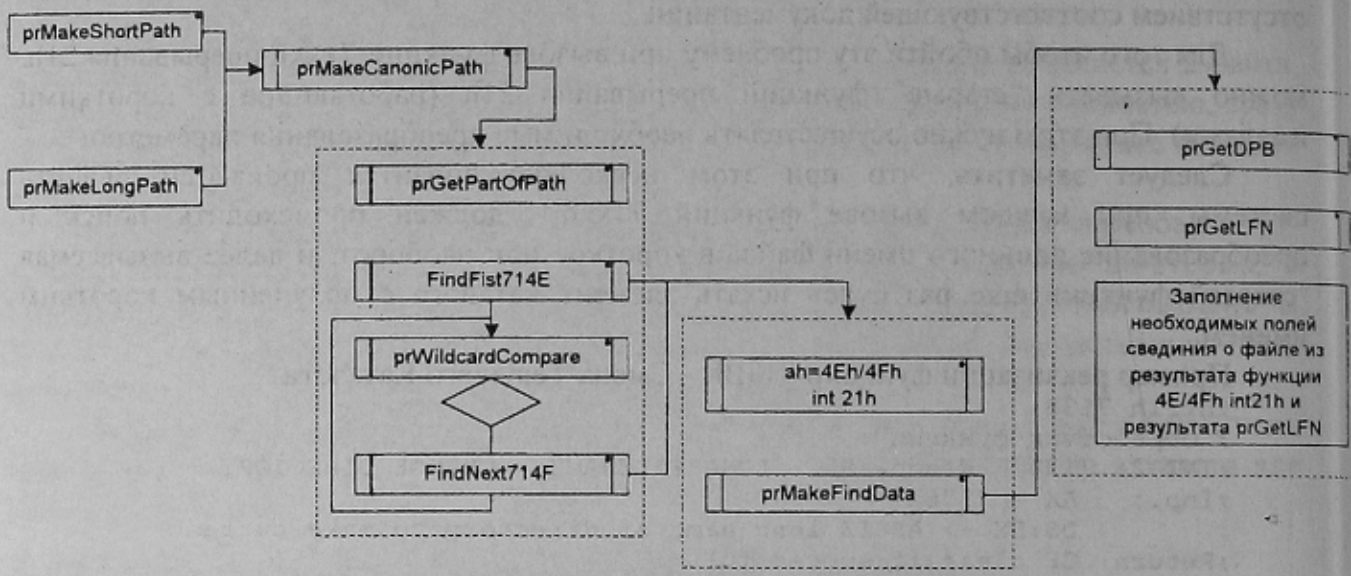
Следует заметить, что при этом несколько снизится производительность системы: при каждом вызове функций 71xxh должен происходить поиск и преобразование длинного имени файла в короткое или наоборот, и далее вызываемая "старая" функция еще раз будет искать элемент каталога с полученным коротким именем.

Пример реализации функции 713Bh – Смена Текущего Каталога :

```
int21h_713B:
; Обработчик функции :
;INT 21 713B - Windows95 - LONG FILENAME - CHANGE DIRECTORY
;Inp.:   AX = 713Bh
;        DS:DX -> ASCIZ long name of directory to make current
;Return: CF clear if successful
;        CF set on error
;            AX = error code
        push    bp
        mov     bp,sp                   ; сохраняем указатель стека
;-Создание короткого пути
        push    ds                      ; сохраняем используемые
        push    dx                      ; регистры
        push    es                      ;
        push    si                      ;
        push    di                      ;
; Вызов процедуры преобразования в путь с коротким именем
;prMakeShortPath
;Inp.:   DS:SI -> ASCIZ long filename or path
;        ES:DI -> (possibly 128-byte) buffer for short filename
        mov     si,dx                   ; Формирование указателей
        les     di,dword ptr pFspec     ;
        call    prMakeShortPath         ; Вызов процедуры
        pop     di
        pop     si
        pop     es
;-Вызов "старой" функции
;INT 21 3B-- - DOS 2+ - "CHDIR" - SET CURRENT DIRECTORY
;Inp.:   AH = 3Bh
;        DS:DX -> ASCIZ pathname to become current directory
;Return: CF clear if successful
;            AX destroyed
;        CF set on error
;            AX = error code
        lds     dx,dword ptr pFspec     ; Формирование
        mov     ah,3Bh
        int     21h
        pop     dx
        pop     ds
        jnc     iRet713B
iError3B:or     word ptr [bp+6],00001h ; установить CF
        jmp     iExt713B
iRet713B:and    word ptr [bp+6],0FFFEh ; сбросить CF
iExt713B:pop    bp
        iret
```

Таким образом, для организации работы функции смены текущего каталога необходимо лишь осуществить преобразование пути с длинным именем в путь с коротким.

Упрощенный алгоритм преобразования пути с длинным именем в путь с



коротким или наоборот выглядит следующим образом: Рис.1.

Рис. 1 Структурная схема алгоритма преобразования пути с длинным именем в путь с коротким или наоборот

где:

`prGetPartOfPath` – процедура выделения порции пути и формирования маски для поиска короткого имени

`prWildcardCompare` – процедура сравнения двух строк (строки и маски)

`prGetDPB` – процедура получения Drive Parameter Block (DPB)

`prGetLFN` – процедура вычисления и чтения номера сектора с коротким именем - по номеру кластера из недокументированного поля Disk Transfer Area заполненного функцией 4Eh/4Fh int21h (см. Приложение2), а также по номеру элемента каталога; далее, если это необходимо, происходит формирование LFN.

Процедуры `prFindFist714E,FindNext714F` используются также в обработчике функции 714Eh/714Fh прерывания 21h.

Таким образом, процедура `prMakeFindData` будет иметь вид:

```
prMakeFindData proc near
        mov         byte ptr [LFNOk_MFD],01h ; set lfn error flag =ok
;--get "current" drive from FindFirst data block(int 21 4Eh/4Fh)
        lds         si,dword ptr pCurFindFileHandle   ; указатель на data b
        mov         al,byte ptr ds:[si]               ; drive
        and         al,7Fh
;--if new current drive == old drive then do not read DPB
        cmp         al,[CurDrive]           ; текущий диск
        je          l21_MFD
        mov         [CurDrive],al
l2_MFD:
        call        prGetDPB                ; чтение Drive Parameter Block (DPB
        jc          ErrorMFD
l21_MFD: call       prGetLFN                ; чтение длинного имени
        jnc         l4_MFD                  ; carry=1 если была ошибка или
l3_MFD:                                     ; это короткое имя
        mov         byte ptr [LFNOk_MFD],00h ; error geting LFN
l4_MFD:
;Заполнение FindData record для 714Eh/714Fh int21h(приведено в сводной
таблице1)
```

Таблица 1 Сводная таблица форматов FindData record для 714Eh/714Fh int21h и FindFirst data block для 4Eh/4Fh int21h.

Format of LFN FindData record		Format of FindFirst data block		Description = идентичны ~ условно равны * новое поле	
Offset	Size	Offset	Size		
00h	DWORD	15h	BYTE	=	file attributes bits 0-6 standard DOS attributes bit 8: temporary file
04h	QWORD	16h 18h	WORD WORD	~	file creation time/date (if SI=0 number of 100ns intervals since 1/1/1601)
0Ch	QWORD			*	last access time
14h	QWORD			*	last modification time
1Ch	DWORD			*	file size (high 32 bits)
20h	DWORD	1Ah	DWORD	=	file size (low 32 bits)
24h	8 BYTEs			?	Reserved
2Ch	260 BYTEs			*	ASCIZ full filename =long or short
130h	14 BYTEs	1Eh	13 BYTEs	~	ASCIZ short filename if long ^ or 0h if short

```
        lds         si,dword ptr pCurFindFileHandle
        add         si,15h
        les         di,dword ptr pFindData
; далее идет копирование одинаковых реквизитов из FindFirst data block
; (int 21 4Eh/4Fh) (даты, времени, размера)
.............
;--Если длинное имя не было сформировано то копируется короткое
        cmp         byte ptr [LFNOk_MFD],01h ;lfn error flag
        je          lOk_MFD
        mov         cx,13
        rep movsb       ; store short file name
        add         di,0F7h
        stosb   ;0h
        jmp         RetMFD
lOk_MFD:                        ;--store short filename to end if lfn=ok
        add         di,260      ; skip lfn
        mov         cx,13
        rep movsb               ; store short file name
        jmp         RetMFD
ErrorMFD:stc
        jmp         ExtGetLFN
RetMFD:  clc
ExtMFD:  ret
LFNOk_MFD               db  00h ; set lfn error flag clear
        endp        prMakeFindData
```

Приведенный здесь алгоритм чтения LFN, позволяет избавится от работы с таблицей FAT, и поиска кластера каталога. В случае любой ошибки, результат представляется в виде короткого имени.

Заключение

Современные методы работы файловых систем невозможны без использования расширенного способа записи имен файлов и каталогов – LFN. Эти требования долны выполнятся в любых современных операционных системах.

Несмотря на то, что в наиболее распространненой ОС Windows9x реализована поддержка LFN, в некоторых критическ ситуациях возникает необходимость обрабатывать файлы с LFN под управлением MS DOS – без участия Windows9x (режим "bare DOS").

Литература

1. Interrupt List (c) by Ralf Brown Release 57 (http://www.pobox.com/~ralf)
2. Джефф Просис, "Как Windows 95 хранит длинные имена файлов", PC Magazine, June 25, 1996, p. 217

Приложение 1

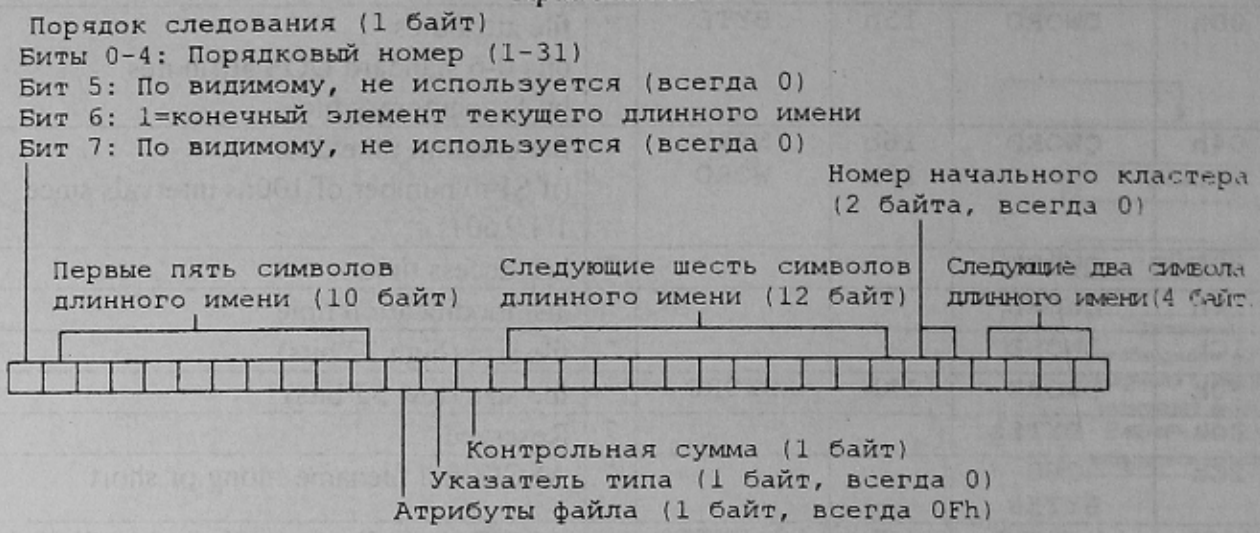


Рис. 2 Элемент каталога для длинного имени

Приложение 2

Таблица 2 Формат FindFirst data block для 4Eh/4Fh int21h с недокументированными полями

Offset	Size	Description
---PC-DOS 3.10, PC-DOS 4.01, MS-DOS 3.2/3.3/5.0---		
00h	BYTE	drive letter (bits 0-6), remote if bit 7 set
01h	11 BYTEs	search template
0Ch	BYTE	search attributes
0Dh	WORD	entry count within directory
0Fh	WORD	cluster number of start of parent directory
--- MS-DOS 7.0---		
11h	WORD	hi cluster number of start of parent directory
13h	WORD	=3D08 ???
---all versions, documented fields---		
15h	BYTE	attribute of file found
16h	WORD	file time
18h	WORD	file date
1Ah	DWORD	file size
1Eh	13 BYTEs	ASCIZ filename+extension

Приложение 3

Ссылки в Internet:

Odi's LFN Tools 1.43 by Ortwin Glueck (with source)
http://odi.webjump.com/

ADIR v1.05 (c) 1998 Chris Jones
http://members.xoom.com/dosuser/dosutils.htm

LFNDir v1.0 (c) 1998 Ziff-Davis Publishing Company by Rick Knoblaugh (with source)
http://www.zdnet.com/pcmag/pctech/content/17/09/ut1709.001.html

OpenDOS Long File Name (LFN) Support Beta 3 (c) 1997 Caldera, Inc.

LFNDOS: DOS LFN driver (c) 1998, 1999 Chris Jones
http://members.xoom.com/dosuser

Поступила в редакційну колегію 20.04.2000 р.