

Linux Conference

October 26- 28, 2011

Prague , Czech Republic



The Extended FAT file system

Differentiating with FAT32 file system

Keshava Munegowda , Venkatraman S

Texas Instruments (India) Pvt Ltd

Bangalore.

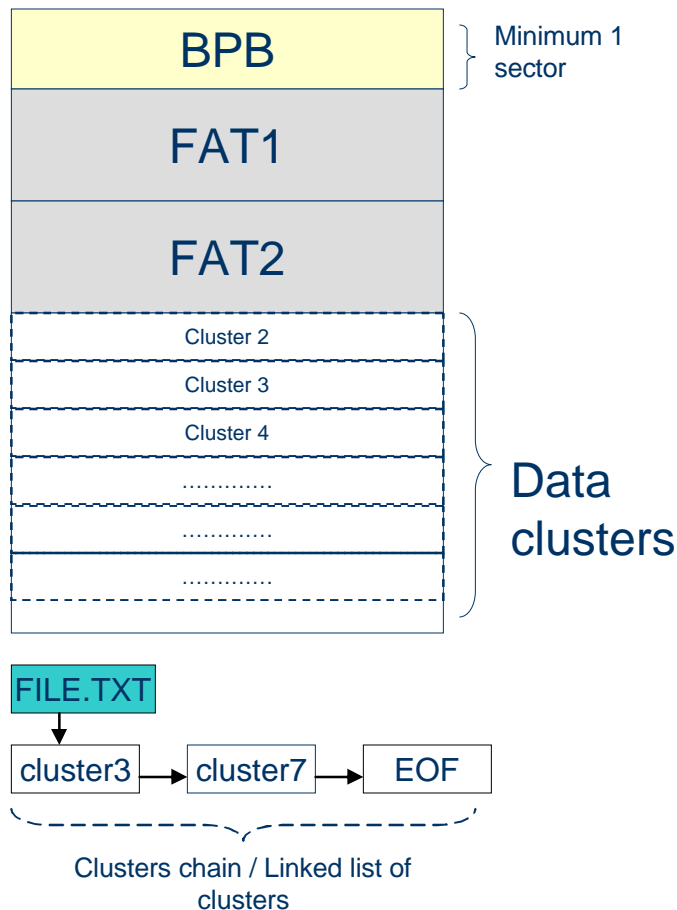
Dr. G T Raju

Professor and Head, Computer Science and Engineering Department,
R N S Institute of Technology, Bangalore, India.

Agenda

- ∅ FAT file system
- ∅ Need for ExFAT file system
- ∅ ExFAT file system organization
- ∅ Boot Sector of FAT and ExFAT
- ∅ Snapshot of Boot Sector of ExFAT
- ∅ Directory Entries of FAT
- ∅ Directory Entries of ExFAT
- ∅ Clusters Heap
- ∅ Up-Case Table
- ∅ Snapshot of Root directory
- ∅ Performance Benchmarking in Linux
- ∅ Performance Benchmarking in Windows
- ∅ References

FAT File system



BPB - BIOS Parameter Block

- BIOS : Basic Input-Output System
- Also Called as “Boot Sector” or “Volume Boot Record”
- Specifies
 - Number of sectors in the storage partition/disk/device
 - Number of FATs (File Allocation Table)
 - Sectors per cluster

FAT1 – File Allocation Table

- Linear linking (chain) of data clusters of the file/directory

FAT2 – Backup of FAT1

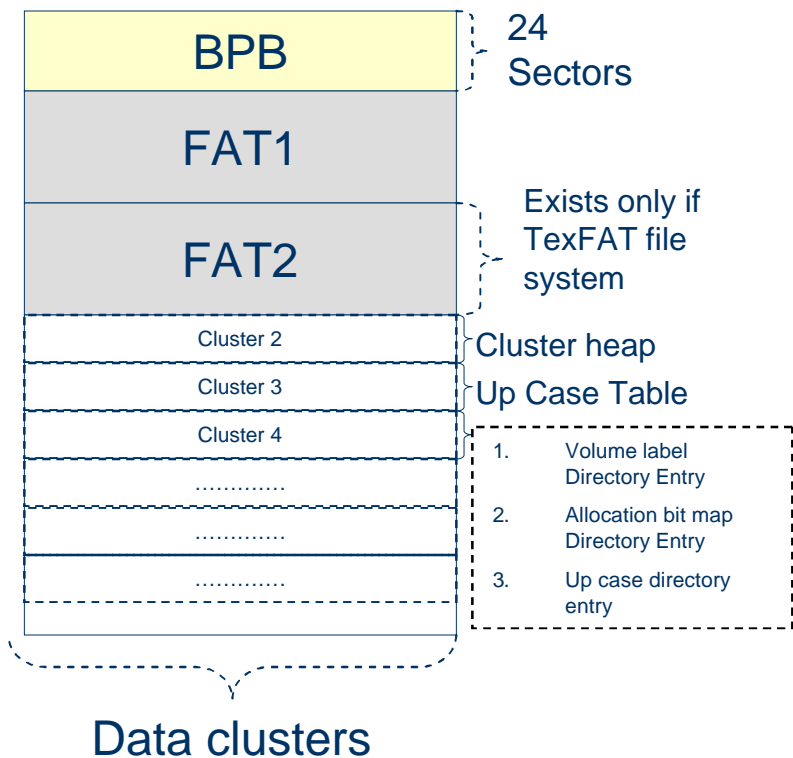
Data clusters

- Group of physical/logical sectors/blocks
- Contains directories or Files data

Need For ExFAT file system

- | FAT File system limited to support only 32GB.
 - FAT does not support Higher size SDXC cards.
- | NTFS
 - Security features
 - Optional for Removable storage devices
 - Meta data overhead for file/directory
 - Write caching mechanism for performance optimizations causes the data corruption in when removable storage device is unplugged.

ExFAT file system Organization



| 12 Sectors for Boot Area
| Another 12 sectors for backup of Boot Area

| FAT2 exists if the TexFAT (Transactional FAT) is used)

| First few Data clusters starting from cluster 2 contains

- Cluster allocation bit map table
- Up case table

Next Cluster Contains

- Cluster allocation Bit map Directory Entry
- Up-case Table Directory Entry
- Volume label Directory Entry

Key Elements of Boot Sector of ExFAT and FAT32

Field name	ExFAT Boot Sector			FAT32 Boot Sector		
	Offset (Byte)	Size (Byte)	Description	Offset (Byte)	Size (Byte)	Description
File system name	3	8	"ExFAT "	"MSWIN4.1" or "MSDOS5.0"		
				82	8	"FAT32 "
Volume length	72	8	Total number of Sectors	32	4	Total number of Sectors
FAT offset	80	4	Sector address of 1 st FAT			
FAT length	84	4	Size of FAT in sectors	36	4	Size of 1 st FAT in sectors
Cluster offset	88	4	Starting sector of cluster heap			
Cluster count	92	4	Number of clusters			
Root directory	96	4	First cluster of root directory	44	4	First cluster of root directory
Volume flags	106	2	Bit 0 – Active FAT 0 – 1 st , 1 – 2 nd Bit 1 – Volume Dirty 0 – Clean, 1- dirty Bits 2 & 3 – Media failure 0 – No failures, 1 – failures reported			
Bytes per Sector	108	1	This is power of 2; Minimal value is 9; 2 ⁹ =512 Bytes and maximum 2 ¹² =4096 Bytes	11	2	Count of bytes per sector; This can have following values 512,1024,2048 or 4096.
Sector per cluster	109	1	This is power of 2; Minimal value is 1; 2 ⁰ =1 sector (512 Bytes) and maximum 32 MB cluster size in bytes	13	1	Number of sectors per cluster; allowed values are 1,2,4,8,32,64 and 128. Maximum of 128 sectors per cluster
Number of FATs	110	1	Either 1 or 2; if TexFAT is supported then it will be 2	16	1	Usually 2

Boot Sector of ExFAT

Boot sector snap shot of ExFAT file system in 4GB USB stick

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	EB	76	90	45	58	46	41	54	20	20	20	00	00	00	00	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	3F	00	00	00	00	00	00	00	A9	9F	77	00	00	00	00	00
00000050	80	00	00	00	00	00	00	00	80	04	00	00	6C	DE	01	00
00000060	04	00	00	00	FF	02	56	00	00	01	00	00	09	06	01	80
00000070	00	00	00	00	00	00	00	00	33	C9	8E	D1	BC	F0	7B	8E
00000080	D9	A0	EB	7D	B4	7D	8B	F0	AC	98	40	74	0C	48	74	0E
00000090	00	00	00	00	00	00	00	00	EF	A0	FD	7D	EB	E6	8D	16
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	0D	0A	52	85	6D	6F	76	65	20	04	00	00	00	00	00	00
00000110	72	28	6F	74	68	65	72	20	6D	65	64	69	61	2E	FF	0D
00000120	0A	44	69	73	6B	20	65	72	72	6F	72	FF	0D	0A	50	72
00000130	65	73	73	20	00	00	00	00	00	00	00	00	00	00	00	00
00000140	65	73	74	61	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	FF
000001C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000001D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000001E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000001F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	1F	2C	55	AA
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

File system name "ExFAT"

Total number of sectors

Number of clusters

Number of FATs

Number of sectors per cluster
 $2^6 = 64$ sectors

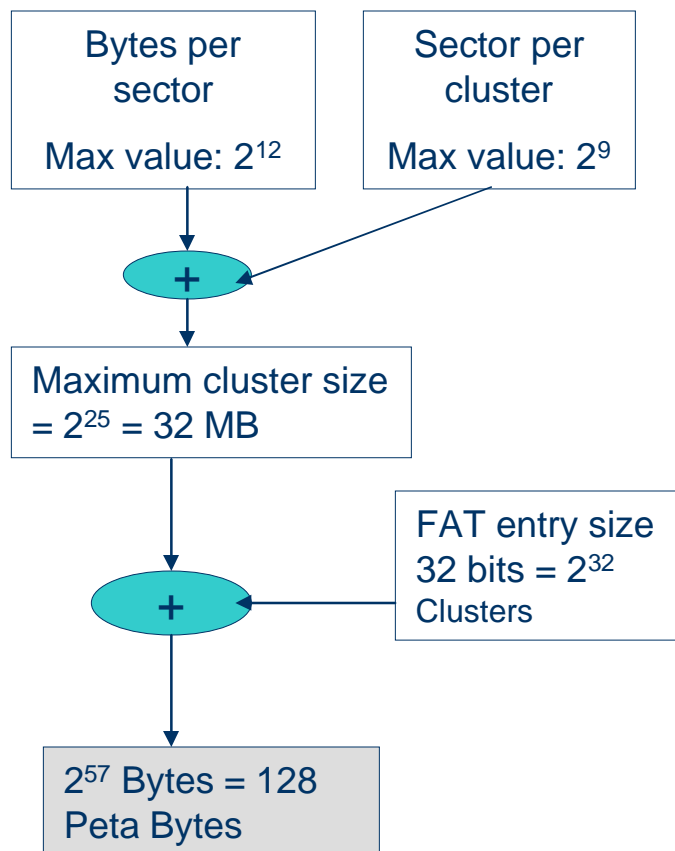
Number of Bytes per sector
 $2^9 = 512$ bytes

Cluster size = $2^6 * 2^9 = 2^{15} = 32KB$

Default Cluster size of ExFAT, FAT and NTFS

Disk size	FAT16	FAT32	NTFS	ExFAT	Note: *	
4 MB – 16MB	1KB	512 Bytes	4KB	4KB	Note: * KB – Kilo Bytes MB – Mega Bytes GB - Giga Bytes TB – Tera Bytes	
16 MB–64MB	2KB					
64 MB–128 MB						
128 MB–256 MB	4KB					
256MB- 512MB	8KB	4KB		32KB		
512MB- 1GB	16KB					
1GB- 2GB	32KB					
2GB – 8GB	Not supported					
8GB–16GB						8KB
16GB-32GB						16KB
32GB- 127GB						32KB
127GB-2TB		Not supported; *Assuming Sector size is 512 bytes, then beyond 127GB is not supported by FAT32, as per FAT spec, the FAT entry size is 32 bits but only 28 bits are used for cluster entry.		128KB		
2TB-16TB						
16TB – 32TB						8KB
32TB-64TB						16KB
64TB – 128TB						32KB
128TB – 256TB					64KB	
>256TB					Custom specific	Custom specific

ExFAT Maximum Volume Support



- | The “Volume length” field of Boot sector is 8 bytes; so the maximum value is 2^{64} Sectors
- | The Maximum value of “Bytes per sector” field of boot sector is $2^{12} = 4096$ bytes
- | So, the Maximum theoretical value support is $2^{(64+12)} = 2^{76}$ which is 64 Zetta Bytes (ZB)
- | The FAT entry size is 32, Maximum cluster number is 2^{32}
- | So, To support 64 ZB , it requires at least $2^{(32+12)} = 16$ Terra Bytes (TB) cluster size.
- | But, Microsoft has set the limitation that cluster size should be of maximum of 32 MB
- | Hence “Bytes per sector” + “Sectors per cluster” of boot sector can not exceed 25. hence the maximum value will be 2^{25} and the maximum cluster are 2^{32} , so it yields maximum value of $2^{(32+25)} = 2^{57}$ Bytes = 128 Peta Bytes (PB)
- | Practically, Maximum Supported Size is 128 PB

Directory Entries of FAT

32 Bytes Directory Entry of FAT32			
Name	Offset (byte)	Size (bytes)	Description
DIR_Name	0	11	Short name.
DIR_Attr	11	1	File attributes
DIR_NTRes	12	1	Reserved
DIR_CrtTimeTenth	13	1	Millisecond stamp at file creation time
DIR_CrtTime	14	2	Time file was created.
DIR_CrtDate	16	2	Date file was created.
DIR_LstAccDate	18	2	Last access date.
DIR_FstClusHI	20	2	High word of this entry's first cluster number
DIR_WrtTime	22	2	Time of last write.
DIR_WrtDate	24	2	Date of last write.
DIR_FstClusLO	26	2	Low word of this entry's first cluster number.
DIR_FileSize	28	4	file size in bytes.

FAT Long Directory Entry Structure			
Name	Offset (byte)	Size (bytes)	Description
LDIR_Ord	0	1	Masked with 0x40 (LAST_LONG_ENTRY), this indicates the entry is the last long dir entry in a set of long dir entries. All valid sets of long dir entries must begin with an entry having this mask.
LDIR_Name1	1	10	Characters 1-5 of the long-name sub-component in this dir entry.
LDIR_Attr	11	1	Attributes - must be ATTR_LONG_NAME
LDIR_Type	12	1	Zero
LDIR_Chksum	13	1	Checksum of name in the short dir entry at the end of the long dir set.
LDIR_Name2	14	12	Characters 6-11 of the long-name sub-component in this dir entry.
LDIR_FstClusLO	26	2	Must be ZERO.
LDIR_Name3	28	4	Characters 12-13 of the long-name sub-component in this dir entry.

Directory Entries of ExFAT

- | Following are the Directory Entry Types defined
 - Critical Primary
 - | Volume Label Directory Entry
 - | Allocation Bitmap Directory Entry
 - | Up-case Table Directory Entry
 - | File/Folder directory Entry
 - Benign primary
 - | Volume GUID
 - | TexFAT Padding
 - | Windows CE Access Control Table
 - Critical Secondary
 - | Stream Extension Directory Entry
 - | File name Extension Directory Entry
- | All Directory Entries are of 32 Bytes

Volume Label Directory Entry

Volume Label Directory Entry of ExFAT

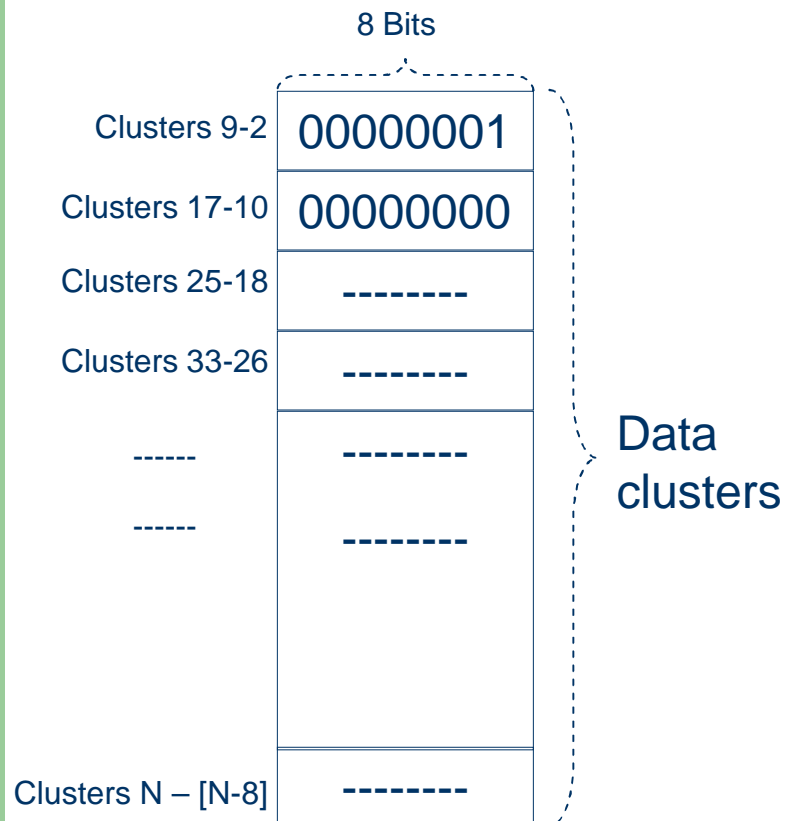
Name	Offset (byte)	Size (bytes)	Description
Entry Type	0	1	0x83 – Indicates that the Volume label exists 0x03- Indicates that the volume label is deleted
Character count	1	1	Number of characters in label
Volume label	2	22	Volume label in Unicode; Maximum of 11 characters of 16 bit Unicode string
Reserved	24	8	

Allocation Bit map (Cluster Heap) Directory Entry

Cluster Heap Directory Entry of ExFAT

Name	Offset (byte)	Size (bytes)	Description
Entry Type	0	1	0x81
Bit Map Flags	1	1	Bit 0 0 – 1 st cluster Heap 1 – 2 nd cluster Heap
Reserved	2	18	
First cluster	20	4	First data cluster number; Usually it will be 2
Data length	24	8	Size of cluster heap in bytes It is calculated as: Ceil (Cluster count / 8)

Cluster Heap



- | Generally the cluster 2 contains the cluster heap
- | if cluster heap size is more than cluster size; then the index 2 of File Allocation Table (FAT) contains the next cluster number which contains the cluster heap.
- | There can exist multiple clusters (cluster chain in FAT) as cluster heap
- | Every Bit in the Cluster heap contains the status of the data cluster; 0 – Free; 1- Allocated
- | Byte 0 contains allocation status of cluster 2 to cluster 9
- | Byte 1 contains the allocation status of cluster 10 to cluster 17
- | Byte 2 contains the allocation status of cluster 18 to cluster 25 and so on

Up-Case Table Directory Entry

Up Case Table Directory Entry of ExFAT

Name	Offset (byte)	Size (bytes)	Description
Entry Type	0	1	0x82
Reserved	1	3	
Table checksum	4	4	Check sum of the Up-case table
Reserved	8	12	
First cluster	20	4	First data cluster
Data length	24	8	Up case table size in bytes

Up-Case Table

The screenshot shows the WinHex application window for Drive F: with the Up-Case Table displayed. The table consists of two columns: hexadecimal offsets and their corresponding ASCII characters. The hexadecimal column starts at 00098000 and ends at 00098270. The ASCII column shows characters from 0 to 255, including spaces, punctuation, and control characters. The table is used for converting lowercase letters to uppercase in the file system.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00098000	00	00	01	00	02	00	03	00	04	00	05	00	06	00	07	00
00098010	08	00	09	00	0A	00	0B	00	0C	00	0D	00	0E	00	0F	00
00098020	10	00	11	00	12	00	13	00	14	00	15	00	16	00	17	00
00098030	18	00	19	00	1A	00	1B	00	1C	00	1D	00	1E	00	1F	00
00098040	20	00	21	00	22	00	23	00	24	00	25	00	26	00	27	00
00098050	28	00	29	00	2A	00	2B	00	2C	00	2D	00	2E	00	2F	00
00098060	30	00	31	00	32	00	33	00	34	00	35	00	36	00	37	00
00098070	38	00	39	00	3A	00	3B	00	3C	00	3D	00	3E	00	3F	00
00098080	40	00	41	00	42	00	43	00	44	00	45	00	46	00	47	00
00098090	48	00	49	00	4A	00	4B	00	4C	00	4D	00	4E	00	4F	00
000980A0	50	00	51	00	52	00	53	00	54	00	55	00	56	00	57	00
000980B0	58	00	59	00	5A	00	5B	00	5C	00	5D	00	5E	00	5F	00
000980C0	60	00	41	00	42	00	43	00	44	00	45	00	46	00	47	00
000980D0	48	00	49	00	4A	00	4B	00	4C	00	4D	00	4E	00	4F	00
000980E0	50	00	51	00	52	00	53	00	54	00	55	00	56	00	57	00
000980F0	58	00	59	00	5A	00	7B	00	7C	00	7D	00	7E	00	7F	00
00098100	80	00	81	00	82	00	83	00	84	00	85	00	86	00	87	00
00098110	88	00	89	00	8A	00	8B	00	8C	00	8D	00	8E	00	8F	00
00098120	90	00	91	00	92	00	93	00	94	00	95	00	96	00	97	00
00098130	98	00	99	00	9A	00	9B	00	9C	00	9D	00	9E	00	9F	00
00098140	A0	00	A1	00	A2	00	A3	00	A4	00	A5	00	A6	00	A7	00
00098150	A8	00	A9	00	AA	00	AB	00	AC	00	AD	00	AE	00	AF	00
00098160	B0	00	B1	00	B2	00	B3	00	B4	00	B5	00	B6	00	B7	00
00098170	B8	00	B9	00	BA	00	BB	00	BC	00	BD	00	BE	00	BF	00
00098180	C0	00	C1	00	C2	00	C3	00	C4	00	C5	00	C6	00	C7	00
00098190	C8	00	C9	00	CA	00	CB	00	CC	00	CD	00	CE	00	CF	00
000981A0	D0	00	D1	00	D2	00	D3	00	D4	00	D5	00	D6	00	D7	00
000981B0	D8	00	D9	00	DA	00	DB	00	DC	00	DD	00	DE	00	DF	00
000981C0	C0	00	C1	00	C2	00	C3	00	C4	00	C5	00	C6	00	C7	00
000981D0	C8	00	C9	00	CA	00	CB	00	CC	00	CD	00	CE	00	CF	00
000981E0	D0	00	D1	00	D2	00	D3	00	D4	00	D5	00	D6	00	D7	00
000981F0	D8	00	D9	00	DA	00	DB	00	DC	00	DD	00	DE	00	DF	00
00098200	00	01	00	01	02	01	02	01	04	01	04	01	06	01	06	01
00098210	08	01	08	01	0A	01	0A	01	0C	01	0C	01	0E	01	0E	01
00098220	10	01	10	01	12	01	12	01	14	01	14	01	16	01	16	01
00098230	18	01	18	01	1A	01	1A	01	1C	01	1C	01	1E	01	1E	01
00098240	20	01	20	01	22	01	22	01	24	01	24	01	26	01	26	01
00098250	28	01	28	01	2A	01	2A	01	2C	01	2C	01	2E	01	2E	01
00098260	30	01	31	01	32	01	32	01	34	01	34	01	36	01	36	01
00098270	38	01	39	01	39	01	3B	01	3B	01	3D	01	3D	01	3F	01

- | The Up case Table is series of Unicode character mappings.
- | One of the uses of the Up case table is “lower case” to “Upper case” conversion when file/directory searching is performed.
- | Up case table is always Read-only.
- | If cluster heap is only one cluster and it is cluster 2; then the Up case table starts from cluster 3.
- | By default, Up-case table starts immediately after the cluster heap.
- | If Up case table size is more than cluster size; File Allocation Table (FAT) contains the cluster chain specifying the data clusters in which content of Up case table is available.
- | Example:
 - WinHex tool Snap shot of first few bytes of Up-case Table
 - Lower case letter “a (0x:61)” converted to Upper case “A (0x41)” by having value 0x0041 at the 16 bit index 0x61.

File Directory Entry

File/Folder Directory Entry of ExFAT

Name	Offset (byte)	Size (bytes)	Description
Entry Type	0	1	0x85
Secondary count	1	1	Number of Secondary directory entries ; Minimum value is 2 and maximum value is 18. These entries will be "Stream Extension directory entry" and "File name extension directory entries"; In general, A file/directory contains 1 File directory Entry, 1 stream extension directory and maximum of 17 file name extension directory entries.
Set checksum	2	2	Checksum is calculated on File directory entry (this entry) + All secondary entries
File Attributes	4	2	Attributes of the file/directory
Reserved	6	2	
Creation time	8	4	Created Date and Time in DOS time stamp format
Last Modified	12	4	Last modified Date and Time in DOS time stamp format
Last Accessed	16	4	Last accessed Date and Time in DOS time stamp format
Create 10ms Increment	20	1	10ms increments between 0 – 199
Last modified 10ms Increment	21	1	10ms increments between 0 – 199
Last accessed 10ms Increment	22	1	10ms increments between 0 – 199
Reserved	23	9	

File Attributes Field structure

File Attributes field of File Directory Entry of ExFAT			
Name	Offset (byte)	Size (bytes)	Description
Read only	0	1	0 – Write permission 1 – Read only
Hidden	1	1	0 – Visible 1 – Hidden
System	2	1	1 – system
Reserved	3	1	In FAT32, this bit is used to indicate that the file/directory entry is volume label
Directory	4	1	0 – File 1 – Directory/Folder
Archive	5	1	1- Archive
Reserved	6	10	

- | The Read-only, Hidden, System, Directory and Archive bit fields are file attribute are same as FAT32 File attributes structure.
- | Volume Id is reserved in file attribute of ExFAT, because there is exist a dedicated directory entry to represent the volume name of the disk/partition

File/Directory Timestamp Structure

Timestamp field of File Directory Entry of ExFAT

Name	Offset (byte)	Size (bytes)	Description
Double Seconds	0	5	Represents the seconds in two-second multiples. Valid range of values is: 0 – which represents 0 seconds 29 – represents 58 seconds
Minute	5	6	Minutes; Valid range 0 -59
Hour	11	5	Hours; Valid range 0 -23
Day	16	5	Day of month; valid range 1-31
Month	21	4	Month of year; 1= January , valid range 1-12
Year	25	7	Count of years from 1980; Valid range is: 0 – year 1980 127 – year 2107

The time stamp format used in the ExFAT is same as FAT

Stream Extension Directory Entry

Stream Extension Directory Entry of ExFAT			
Name	Offset (byte)	Size (bytes)	Description
Entry Type	0	1	0xC0
General Secondary Flags	1	1	Bit 0 : Allocation possible 0 – No cluster allocated; 1 – cluster allocation is possible Bit 1 : No FAT chain 0 – Yes ; The clusters of this file/directory are NOT contiguous 1 – No; The Contiguous Cluster are allocated to this file/directory; This improves the File read performance Bits 2 – 7 : Reserved
Reserved	2	1	
Name length	3	1	Length of file name in bytes (Maximum of 255 Unicode characters)
Name hash	4	2	Hash of the file name; Used while searching for fie/directory name
Reserved	6	2	
Valid data length	8	8	The size of the file of directory in bytes
Reserved	16	4	
First cluster	20	4	First data cluster of the file/directory
Data length	24	8	The size of the file of directory in bytes; In case of directory maximum size is 256MB

File Name Extension Directory Entry

File Name Extension Directory Entry of ExFAT			
Name	Offset (byte)	Size (bytes)	Description
Entry Type	0	1	0xC1
General Secondary Flags	1	1	Bit 0 : Allocation possible 0 – No cluster allocated; 1 – cluster allocation is possible Bit 1 : No FAT chain 0 – Yes ; The clusters of this file/directory are NOT contiguous 1 – No; The Contiguous Cluster are allocated to this file/directory; This improves the File read performance Bits 2 – 7 : Reserved
File name	2	30	15 Unicode character of the part of the file name

- | Single File name Extension directory entry can contain 15 Unicode characters.
- | Maximum File name size is 255 Unicode characters
- | Hence , for a single long file/directory name maximum of 17 file name extension directory entries are possible.

File/Directory and Volume label Deletion process

- | The Most Significant Bit (MSB) of Entry Type field of the directory entry will be changed from 1 to 0 to indicate the directory entry is deleted.
- | For example, A file contains the following directory entries
 - File Directory Entry , Entry Type is $0x85 = 0x10000101$ in binary format
 - Stream Extension Directory Entry, Entry type is $0xC0 = 0x11000000$ in binary format
 - File name extension Directory Entry, Entry type is $0xC1 = 0x11000001$ in Binary format
- | The MSB (7th) Bit indicates “In Use” field; it means 1 – In use, 0 – deleted.
- | This MSB bit is changed from 1 to 0 to indicate the deletion process;
 - File Directory Entry , Entry Type is $0x85$ is changed to $0x05 = 0x00000101$ in binary format
 - Stream Extension Directory Entry, Entry type is $0xC0$ is changed to $0x40 = 0x01000000$ in binary format
 - File name extension Directory Entry, Entry type is $0xC1$ is changed to $0x41 = 0x01000001$ in binary format
- | The Entry type of the volume label is $0x83 = 0x10000011$ in binary format.
- | This Entry type value $0x83$ will be changed to $0x03 = 0x00000011$ to indicate that there is no volume label.

Snapshot of the Root Directory

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
000A0000	83	0A	45	00	78	00	46	00	41	00	54	00	44	00	52	00	E x F A T D R
000A0001	49	00	56	00	45	00	00	00	00	00	00	00	00	00	00	00	I V E
000A0002	81	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I ;
000A0003	00	00	00	00	02	00	00	00	CE	3B	00	00	00	00	00	00	I
000A0004	82	00	00	00	0D	D3	19	E6	00	00	00	00	00	00	00	00	I 0 *
000A0005	00	00	00	00	03	00	00	00	CC	16	00	00	00	00	00	00	I
000A0006	8E	02	1A	05	20	00	00	00	D2	9A	53	3F	26	8E	53	3F	I S ? S ? S ?
000A0007	D2	9A	53	3F	95	00	96	96	96	00	00	00	00	00	00	00	I S ? S ? S ?
000A0008	C0	03	00	0B	44	A2	00	00	00	02	00	00	00	00	00	00	A De
000A0009	00	00	00	00	05	00	00	00	00	02	00	00	00	00	00	00	I
000A000A	C1	00	4B	00	45	00	53	00	48	00	41	00	56	00	41	00	A K E S H A V A
000A000B	2E	00	54	00	58	00	54	00	00	00	00	00	00	00	00	00	. T X T
000A000C	85	02	4A	A8	10	00	00	00	E4	9A	53	3F	E4	9A	53	3F	I J ? a I S ? a I S ?
000A000D	E4	9A	53	3F	A0	A0	96	96	96	00	00	00	00	00	00	00	a I S ? S ? S ?
000A000E	C0	03	00	02	2F	00	00	00	00	80	00	00	00	00	00	00	A /
000A000F	00	00	00	00	06	00	00	00	00	80	00	00	00	00	00	00	I
000A0010	C1	00	54	00	49	00	00	00	00	00	00	00	00	00	00	00	A T I
000A0011	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A0012	05	02	38	81	20	00	00	00	C8	AA	53	3F	26	8E	53	3F	I 8 I E a S ? S ? S ?
000A0013	D2	AA	53	3F	9C	00	96	96	96	00	00	00	00	00	00	00	O a S ? S ? S ?
000A0014	40	03	00	0B	0C	2B	00	00	00	02	00	00	00	00	00	00	@ +
000A0015	00	00	00	00	08	00	00	00	00	02	00	00	00	00	00	00	I
000A0016	41	00	46	00	49	00	46	00	45	00	2E	00	54	00	58	00	A F I L E . T X
000A0017	54	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	T
000A0018	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A0019	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A001A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A001B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A001C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A001D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A001E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A001F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A0200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A0210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A0220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I
000A0230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	I

Root Directory ; The starting cluster 4 ; The WinHex tool is showing contents of cluster 4

Volume label directory entry, named "ExFATdrive"

Cluster heap directory entry; with starting data cluster 2

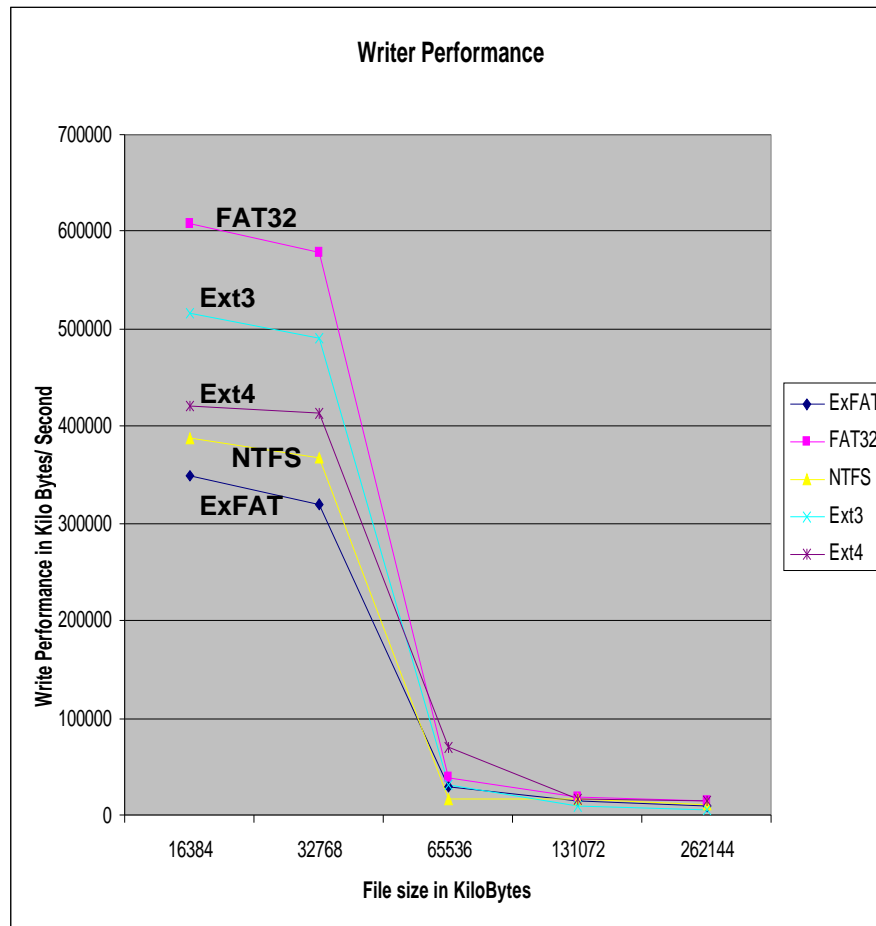
Up case directory entry; with starting data cluster 3

File created with
- file /directory entry
- Stream extension entry , starting data cluster 5.
- File name extension entry with the name "KESHAVA.TXT"

Directory created with
- file /directory entry
- Stream extension entry , starting data cluster 5.
- File name extension entry with the name "TI"

File named "FILE.TXT" is DELETED

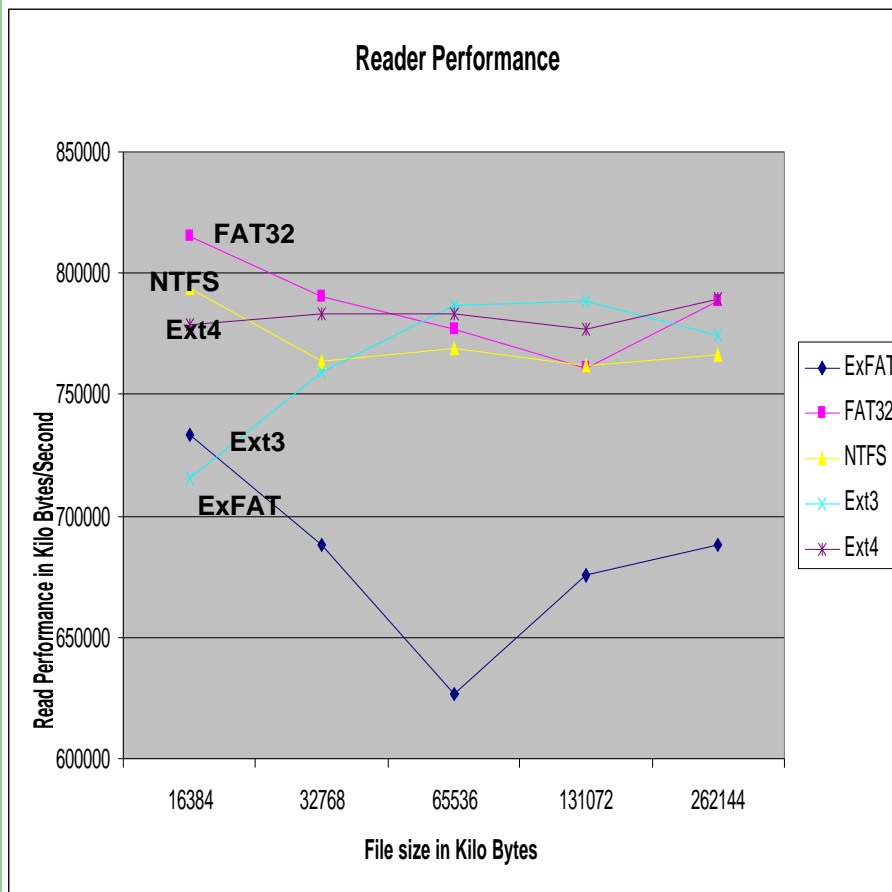
File systems Write performance numbers in Linux



Test Setup	
Computer	Dell Computer, Optiplex GX620 Duel core 3.4Ghz with 2GB RAM
Operating system	Ubuntu 9.04 with Linux kernel version 2.6.28
Storage device	Transcend 4GB Thumb drive
Performance tool	lozone version 3.3

- | The file size range 16MB to 256MB
- | The buffer size : 4MB to write
- | For small files higher performance range is 300MB/S (Mega bytes per second) to 700MB/S.
- | For larger files, the performance range is 1MB/S to 12MB/S.
- | The ExFAT is implemented as FUSE(File system in User space) module.
- | FAT32 has seen good and in performance for both smaller and larger files.

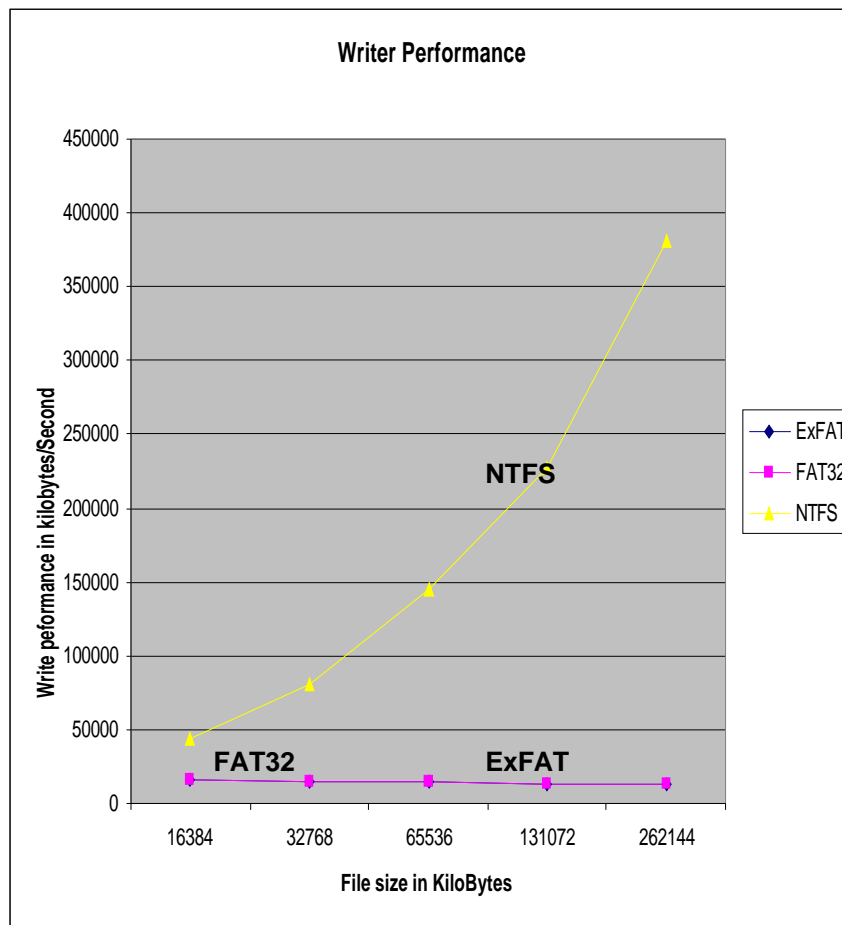
File systems Read performance numbers in Linux



Test Setup	
Computer	Dell Computer, Optiplex GX620 Duel core 3.4Ghz with 2GB RAM
Operating system	Ubuntu 9.04 with Linux kernel version 2.6.28
Storage device	Transcend 4GB Thumb drive
Performance tool	lozone version 3.3

- | The file size range 16MB to 256MB
- | The buffer size : 4MB to Read
- | For small files higher performance range is 700MB/S (Mega bytes per second) to 850MB/S.
- | For larger files, the performance range is 600MB/S to 800MB/S.
- | Compared to write, the read performance degradation is less.
- | FAT32 and Ext4 has seen good and consistent in performance for both smaller and larger files.

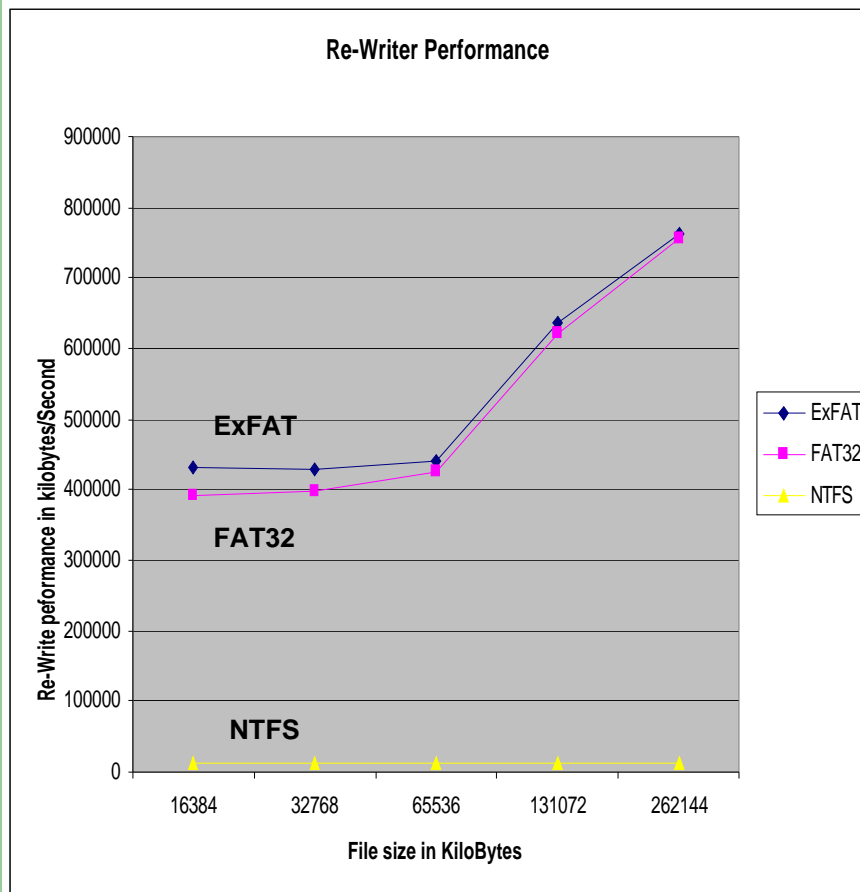
File systems Write performance numbers in WindowsXP



Test Setup	
Computer	Dell Lap-top, Latitude D630 Intel Core 2Duo CPU 2.4 GHz, &1.17Ghz
Operating system	Windows XP Profession version 2002, Service pack3
Storage device	Transcend 4GB Thumb drive
Performance tool	lozone version 3.3

- | The file size range 16MB to 256MB
- | The buffer size : 4MB to Write
- | NTFS is has highest performance range 50MB/S to 400MB/S
- | Both ExFAT and FAT32 has lower performance less than 50 MB/s
- | The NTFS performance is higher because of "Write caching" mechanism
- | In Window XP, user has to set the "Write caching mechanism by enabling the performance optimization policy in the Hardware tab of the drive.

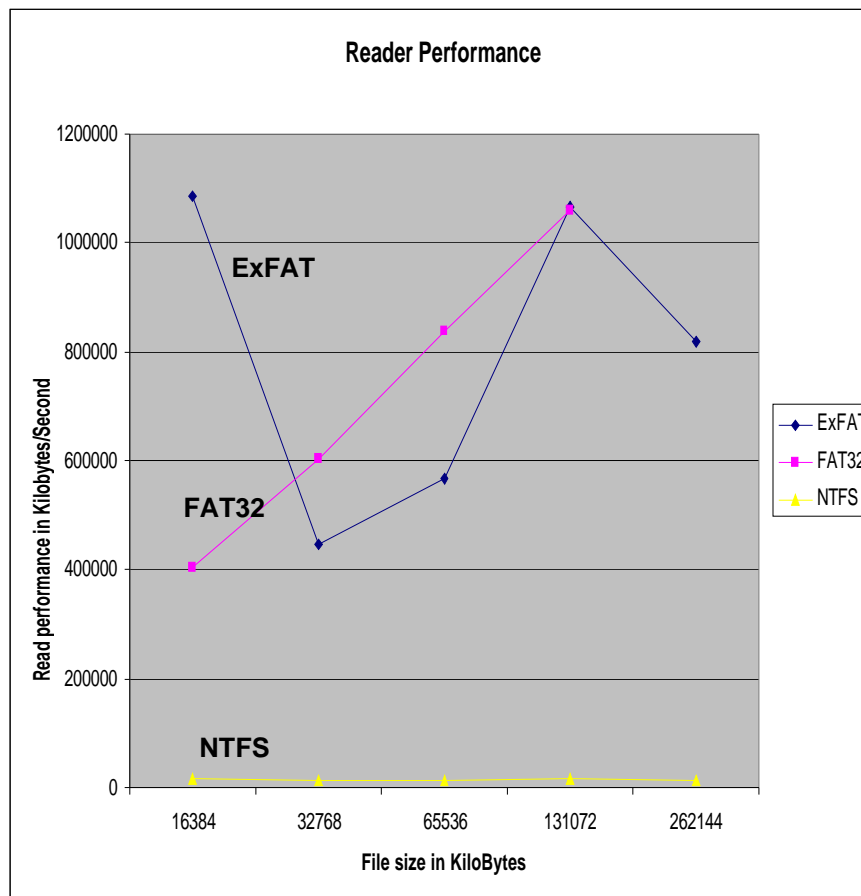
File systems Re-Write performance numbers in WindowsXP



Test Setup	
Computer	Dell Lap-top, Latitude D630 Intel Core 2Duo CPU 2.4 GHz, &1.17Ghz
Operating system	Windows XP Profession version 2002, Service pack3
Storage device	Transcend 4GB Thumb drive
Performance tool	lozone version 3.3

- | Re-Write means write the file that already exists.
- | The file size range 16MB to 256MB
- | The buffer size : 4MB to Write
- | NTFS is has lowest performance below 100 MB/S
- | Both ExFAT and FAT32 has higher performance range 400MB/s to 800MB/s
- | ExFAT has slightly higher performance than FAT32.

File systems Read performance numbers in WindowsXP



Test Setup	
Computer	Dell Lap-top, Latitude D630 Intel Core 2Duo CPU 2.4 GHz, &1.17Ghz
Operating system	Windows XP Profession version 2002, Service pack3
Storage device	Transcend 4GB Thumb drive
Performance tool	lozone version 3.3

- | The file size range 16MB to 256MB
- | The buffer size : 4MB to Read
- | NTFS is has lowest performance below 100 MB/S
- | Both ExFAT and FAT32 has higher performance range 400MB/s to 2GB(Giga Bytes)/S

References

- | US patent Application: 12/389396, “QUICK FILENAME LOOKUP USING NAME”, Ravishankar V.Puipeddi, Vishal V.Ghotge, Ravinder S.Thind, Microsoft Corporation.
- | US patent Application: 11/752872, “EXTENDING CLUSTER ALLOCATIONS IN EXTENSIBLE FILE SYSTEM”, Ravishankar V.Puipeddi, Vishal V.Ghotge, Sarosh C.Havewala, Ravinder S.Thind, Mark J Zbikowski, Microsoft Corporation.
- | US patent Application: 11/229485, “EXTENSIBLE FILE SYSTEM”, Ravishankar V.Puipeddi, Vishal V.Ghotge, Sarosh C.Havewala, Ravinder S.Thind, Mark J Zbikowski, Microsoft Corporation.
- | “Reverse Engineering the Microsoft Extended FAT file system(ExFAT)”, Robert Shullich, SANS Institute InfoSec Reading Room
- | “Extended FAT file system”, Jeff Hamm, Paradigm Solutions, Techno Forensics Conference, October 2009.
- | www.ntfs.com
- | Free Fuse based ExFAT implementation for Linux : <http://code.google.com/p/exfat/> , Developer : Andrew Nayenko
- | lozone,” File systems performance benchmarking tool”: <http://www.iozone.org/>
- | FAT Specification:
<http://msdn.microsoft.com/en-us/windows/hardware/gg463080>

Questions

Queries and Feedback

- keshava_mgowda@ti.com
- keshava.gowda@gmail.com