

# فایل سیستم چیست؟

کاری از محمد علی ناظری

**File System** در یک سیستم عامل به ساختار کلی نامگذاری، ذخیره سازی و سازماندهی فایل ها گفته می شود. سیستم فایل ها، داده ها را به واحدهایی به نام **Cluster** تقسیم بندی می کنند. هر کلاستر می تواند فقط بخشی از یک فایل را نگه دارد. یک فایل ممکن است چندین کلاستر را به خود اختصاص دهد، و مقداری فضای خالی در آخرین کلاستر بلا استفاده باقی بماند. کلاسترهای کوچک می توانند داده های بیشتری در یک دیسک جای بدهند، (چون فضای بلا استفاده کمتری به وجود می آورند) اما کلاسترهای بزرگ، کارایی را بالا می برند.

(در کلاسترهای بزرگتر، داده های بیشتری می تواند در هر کلاستر نگهداری شود. بدین معنی که برای بازیابی آن، سیستم کار کمتری انجام می دهد و در نتیجه به طور کارآمدتر و سریعتر آن را بازیابی می کند) اندازه کلاستر در اندازه دیسک سختی که کامپیوتر می تواند پشتیبانی کند نیز نقش دارد. اما جالب است بدانیم که تا کنون بیش از ۴۰ نوع سیستم فایل طراحی و پیاده سازی شده است که هر کدام مخصوص یک یا چند سیستم عامل هستند.

برخی از سیستم عامل ها برای افزایش قابلیت های خود از سیستم فایل های دیگر سیستم عامل ها نیز پشتیبانی می کنند. به عنوان نمونه سیستم فایل های **ex2** و **ex3** مخصوص سیستم عامل های لینوکس است و **Fat** مخصوص ویندوز. اما به نسخه های جدید لینوکس قابلیت پشتیبانی از سیستم فایل **Fat** نیز افزوده شده است که در صورتی که یک رایانه از هر دو سیستم عامل استفاده می کند بتواند فایل های موجود به صورت مشترک استفاده کنند.

همانطور که می دانید هارد دیسک هنگامی که در کارخانه ساخته می شود کاملاً خام است. پس از طی شدن فرآیند تولید عمل فرمت سطح پایین یا همان **low level format** روی آن انجام می شود. با این عمل هارد به صورت شیار شیار و در حالتی که هر شیار از چندین سکتور تشکیل شده قالب بندی و وضعیت بدسکتورها مشخص می شود. مابین شیارها شکافی برنامه ریزی شده وجود دارد که به آن **Gap** یا گوه گفته می شود. وظیفه این گوه ها این است که آدرس

سکتورهای فیزیکی دو طرف خود را نگهداری کرده و برای اطلاع کنترل گر هارد دیسک علامت لازم را فراهم کنند. معمولاً هاردهای فعلی در کارخانه فرمت سطح پایین می‌شوند و نیازی به این کار توسط کاربر نیست. بعد از اینکه هارد فرمت سطح پایین شد لازم است پارتیشن بندی شود. پارتیشن بندی شامل: ایجاد جدول پارتیشن ( که وظیفه نگهداری آدرس پارتیشنها ) یا همان **Master Boot Record ( MBR )** را دارد و خود قطعاتی که بعداً به صورت درایوهای هارد دیسک شناخته می‌شود و بوت رکورد پارتیشن‌ها است.

پس از پایان پارتیشن بندی هارد، فرمت سطح بالا (یا همان فرمت معمولی) لازم است. در این نوع فرمت هر پارتیشن به صورت مجزا از دیگران قابل بندی شده و علاوه بر بوت رکورد و سکتورهای منطقی ( که با سکتورهای مرحله فرمت سطح پایین متفاوت است ) دو سکتور نیز جهت جدول اسکان فایل یا همان **FAT** که مبحث اصلی ما است، تشکیل می‌شود. قابل ذکر است که دو نسخه تکراری از یک جدول در هر هارد قرار می‌گیرد به جهت افزایش اطمینان از اینکه حذف یا خراب نمی‌شوند. این جدول مهمترین جدول برای اطلاعات موجود در هارد است.

هنگامی که کاربر در سطح سیستم عامل با کامپیوتر کار می‌کند، در واقع با داده‌هایی سر و کار دارد که به صورت منطقی در فایل‌های مختلف سازماندهی شده‌اند. یعنی سکتورهای منفرد به صورت مستقیم مورد استفاده قرار نمی‌گیرند. اما سیستم عامل با سکتورها سروکار داشته و به علت بعضی محدودیتها در هنگام طراحی، هر بار می‌تواند با مجموعه‌ای از سکتورهای متوالی که به آنها کلاستر یا خوشه گفته می‌شود می‌گویند. این کلاسترها با توجه به اینکه به هم پیوسته هستند واحدهای تخصیص یافته یا **Allocation unit** نیز نامیده می‌شوند. برای اینکه سیستم عامل بتواند عمل درست خود را انجام دهد تعداد سکتورها در هر کلاستر باید توانی از ۲ باشد ( برای مثال ۴، ۸، ۱۶، ۳۲ و ... ). در حال استفاده از کامپیوتر دایما فایل‌ها ایجاد یا حذف شده و یا تغییر داده می‌شوند. به همین دلیل به سختی می‌توان کلاسترهای مربوط به یک فایل را در کنار هم نگه داشت. مثلاً فرض کنید که دیسکی حاوی ۳ فایل باشد. اولین فایل خوشه ۳ خوشه، دومی ۲ خوشه و فایل سوم ۳ خوشه اشغال کرده باشند. حالا فرض کنید فایل دوم از روی دیسک پاک شود بنابراین به اندازه ۲ خوشه مابین فایل‌های اول و سوم ( که الان دومی شده است ) فاصله می‌افتد. حالا اگر فایلی که ۴ خوشه دارد به

دیسک اضافه شود، دو امکان پیش می آید: اول اینکه فایل جدید بعد از فایل دوم قرار گیرد و دوم اینکه قسمتی از فایل در دو خوشه آزاد قبلی نوشته شود و بقیه آن در ادامه فایل دوم ذخیره شود. به دلیل کاهش شدید سرعت دسترسی به فایل استفاده از روش اول مقرون به صرفه نیست. بنا بر این فایل سوم فعلی ما در دو خوشه مجزا (دو کلاستر غیر متوالی) ذخیره می شود به این حالت پراکندگی فایل گفته می شود.

نکته قابل ذکر این است که برنامه هایی مانند Defrag جهت رفع این حالت به کار گرفته می شود. در صورتی که فایل پراکنده نمی شد، مشخص کردن خوشه تمام فایل ها به سادگی صورت می گرفت زیرا کافی بود آدرس خوشه ابتدایی و تعداد خوشه های استفاده شده نگهداری شود ولی به دلیل پراکندگی این امکان وجود ندارد و به همین علت جدول اسکان فایل یا همون FAT خودمون ایجاد شده است.

در حال حاضر ۳ نوع FAT وجود دارد ۱۲ و ۱۶ و ۳۲ بیتی. FAT دوازده بیتی مخصوص فلاپی دیسک ها، FAT شانزده بیتی مخصوص استاندارد قدیمی هاردها و FAT سی و دو بیتی مخصوص سیستم عاملهایی مانند ویندوز ۹۸ به بالا!

FAT زنجیره خوشه ها را مشخص می کند به ازای هر شماره خوشه یک داده ثبت شده در Fat وجود دارد. همچنین هر یک از داده های ثبت شده شماره خوشه بعدی را در زنجیره خوشه ها مشخص می کند. اگر خوشه ای در این زنجیره قرار نداشته باشد یعنی آزاد بوده و می توان از آن برای ذخیره استفاده کرد.

**Fat 16**: این سیستم فایل، یک سیستم فایل ۱۶ بیتی است، زیرا کلاسترهایی را تشخیص می دهد که ۱۶ بیت طول (یا رقم) دارند. اندازه کلاسترهای Fat بسته به مقدار فضا درایوی است و از آنجا که Fat16 نمی تواند به اندازه

Fat32 کلاستر داشته باشد، برای پوشش دادن به مقدار یکسانی از فضا به کلاسترهای بزرگتری نیاز دارد.

Windows XP کلاسترهای ۶۴ کیلو بایتی را برای Fat16 پشتیبانی می کند. چون Fat16 فقط ۶۵،۵۳۶ کلاستر را پشتیبانی می کند، با این سیستم فایل، حداکثر یک پارتیشن ۴ گیگابایتی را می توان تقسیم بندی کرد.

**Fat32**: یک سیستم فایل ۳۲ بیتی است ، می تواند کلاسترهای بیشتری را نسبت به **Fat16** آدرس دهی کند . در نتیجه

**Fat32** می تواند از کلاسترهای کوچکتری برای کارآمدتر کردن سیستم ذخیره سازی ، بهره بگیرد . همچنین بدین

مفهوم است که **Fat32** می تواند پارتنشهای بزرگتری را پشتیبانی کند . **Fat32** با کلاسترهای ۳۲ بیتی می تواند

پارتنشهایی تا ۸ ترا بایت را پشتیبانی کند .

### افزایش قابلیت ها، کاهش محدودیتها

**NTFS** بر گرفته از حروف اول عبارت **New Technology File System** است که توسط شرکت

مایکروسافت در سال ۱۹۹۳ طراحی شد. این سیستم فایل استاندارد برای ویندوز **NT** و نسخه های بعد از آن یعنی

**۲۰۰۰** ، **Server 2003** ، **Xp** و **Vista** طراحی شد. نسخه های قدیمی تر ویندوز از قبیل **۳،۱** و **۹۵** و **۹۸** و **Me** از

این سیستم فایل پشتیبانی نمی کنند.

**NTFS** پیشرفت ها و قابلیت های زیادی نسبت به **FAT** دارد که از این قابلیت می توان به مواردی اشاره نمود. یک

قابلیت **NTFS** پشتیبانی از فوق داده (**Meta Data**) و استفاده از ساختمان داده های پیشرفته برای بهبود کارایی ،

قابلیت اطمینان ؛ استفاده بهینه از فضای دیسک را دارد. لازم به توضیح است که به داده هایی که داده های دیگر را

توصیف می کنند فوق داده یا **Meta Data** می گویند.

در سیستم فایل **FAT** ، جدول تخصیص فضا (**File Allocation Table**) داشتیم و در اینجا جدول فایل اصلی

**MFT (Master File Table)** داریم . البته **MFT** بسیار پیچیده تر است . **MFT** صفات فایل را برای هر فایل

ذخیره شده در پارتنش **NTFS** ذخیره می کند . ( صفات فایل هر چیزی را که لازم است درباره یک فایل دانست ،

توصیف می کند . در **NTFS** حتی داده های موجود در یک فایل نیز صفت فایل محسوب می شود . اسم ، مکان ، و

اطلاعات امنیتی نیز ، از دیگر صفات فایل محسوب می شود )

**NTFS** در صورت امکان همه صفات فایل را ، شامل داده های فایل ، در **MFT** ذخیره می کند . ( هر چند در اکثر موارد ، جای کافی برای ذخیره همه صفات در **MFT** وجود ندارد . در این زمان صفت داده ای به خارج **MFT** انتقال می یابد . )

**MFT** بعنوان یک فایل **Metadata** مشهور است . ( **Metadata** اساسا داده هایی درباره خود **Data** است . بعنوان مثال عنوان ، موضوع ، مولف و اندازه یک فایل ، **Metadata** آن فایل را تشکیل می دهند ) **NTFS** از فایل های **Metadata** برای مدیریت داده های روی پارتیشن بهره می گیرد . بعضی از مهمترین فایل های **Metadata** بشرح زیر است :

**MFT Mirror** : یک نسخه پشتیبان از **MFT** است . **MFT** در ابتدای سطح دیسک ذخیره می شود ، در حالیکه **MFT Mirror** در وسط دیسک قرار می گیرد . و در صورت خراب شدن **MFT** اصلی می تواند مساله را حل کند . و اطلاعات لازم را برای بازیابی داده ها روی سیستم فراهم کند .

**Log File** : که **Log Journal** نیز نامیده می شود ، تغییرات روی سیستم فایل را ثبت می کند . **Log File** فقط عملیاتی را که انجام می گیرد ثبت می کند ، نه داده هایی که اصلاح می شوند .

**Cluster Allocation Bitmap** : نقشه ای از پارتیشن را در خود دارد . سیستم می تواند این اطلاعات را برای یافتن کلاسترهای موجود به کار گیرد .

**Bad Cluster File** : بخشهایی از دیسک را که معیوب شده اند ، علامت گذاری می کند . بدین ترتیب **NTFS** از کلاسترهای معیوب که در این فایل علامت گذاری شده اند برای ذخیره داده های جدید استفاده نخواهد کرد .

**Quota Table** : جدول سهمیه ها ، یک خصوصیت جدید در **windows 2000/xp** با استفاده از این خصوصیت می توانید بعنوان مدیر سیستم ، دایرکتوری ها و زیر دایرکتوری ها را در دیسک سخت کنترل کنید . و میزان مصرف فضای دیسک سخت کاربران مختلف یک کامپیوتر را کنترل کنید .

از قابلیت های ویژه ی **NTFS** می توان به لیست های کنترل دسترسی و ثبت وقایع سیستم فایل اشاره نمود. با اینکه سیستم فایل های قدیمی تر و همچنین ویندوز ملنیوم (**Me**) و **۹۸** از **NTFS** پشتیبانی نمی کنند ، اما نرم افزار های خاصی وجود دارند که این قابلیت استفاده از **NTFS** را نه تنها در این ویندوزها ، حتی در محیط **Dos** فراهمی می آورند که برخی از آنها به رایگان در اینترنت وجود دارند.

**NTFS** پنج نگارش دارد.

نگارهای **۱/۰** و **۱/۱** و **۱/۲** در ویندوز **NT** ، نگارش **۳/۰** در ویندوز **۲۰۰۰** و نگارش **۳/۱** در ویندوز **XP**

**Server2003** به کار گرفته شده است.

با افزایش شماره نگارش ، برخی قابلیت ها و امکانات اضافه شدند، به عنوان نمونه از نگارش **۳/۰** به بعد مفهومی به نام سهمیه بندی (**quotas**) به وجود آمد.

پیاده سازی **NTFS** پیچیده و مشکل است و جزئیات کار آن به عنوان اطلاعات محرمانه تنها نزد شرکت مایکروسافت ، طراح این سیستم فایل است. البته مهندسان و طراحان با تجربه از طریق مهندسی معکوس توانسته اند بسیاری از ساختارهای آنرا پیاد سازی کنند، اما پیاده سازی تمامی خصوصیات آن تا کنون پنهان مانده است. مهندسی معکوس روشی است که از مطالعه و تجزیه یک محصول، پی به اجزای ساخت و روش های پیاده سازی می برند، بر خلافی مهندسی که از طریق مطالعه روی اجزا و روش ها به محصول دسترسی می یابد.

خصوصیات و محدودیت ها!

**NTFS** از ساختار دودویی (**B+tree**) برای شاخص بندی اطلاعات فایل ها استفاده می کند. اگرچه پیاده سازی این روش پیچیده است، در برخی موارد سرعت دستیابی سریعتر را ممکن می سازد. هم چنین از نظر امنیت و قابلیت اطمینان نیز نسبت به **Fat** برتری دارد.

**NTFS** می تواند تعداد کلاسترهای بیشتری را نسبت به **Fat32** پشتیبانی کند . در نتیجه ، کلاسترهای **NTFS** عموماً

کوچک و برای حفظ فضای دیسک سخت ، کارآمد هستند . یکی دیگر از مزایای **NTFS** پشتیبانی آن برای

مجوزهای فایل و دایرکتوری است. این خصوصیت امنیتی، اطمینان می دهد که فقط کاربران خاصی به فایلها و دایرکتوری های ویژه دسترسی دارند.

در میان آنچه ذکر شد، سیستم فایل کارآمد **NTFS** است. چون می تواند برای پارتیشنهای ۲ گیگابایت و بیشتر، کلاسترهای ۴ کیلو بایتی ایجاد کند، که بهترین تعادل بین سرعت عمل دیسک سخت و صرفه جویی در فضای دیسک را فراهم می سازد. و در واقع بهترین کارآیی را به سیستم می دهد.

در سیستم فایل **NTFS** با پیاده سازی کنونی یک فایل حداکثر می تواند ۱۶ ترابایت باشد که می توان گفت فایلی با این حجم تاکنون پای به عرصه هستی نگذاشته!

البته در تئوری **NTFS** اندازه یک فایل می تواند حتی یک میلیون بار بزرگتر از این مقدار باشد. حد اکثر تعداد فایل ها در یک پارتیشن **NTFS** نیز حدود ۴ میلیارد و ۳۰۰ میلیون عدد می تواند باشد. حد اکثر اندازه یک پارتیشن

**NTFS** هم با پیاده سازی کنونی ۲۵۶ ترابایت و در تئوری ۸ هزار بار بیشتر از این مقدار می تواند باشد. نکته جالب توجه این است که حجم بزرگترین هارد دیسک های کنونی به یک ترابایت هم نمی رسد!! نام فایل ها در **NTFS** حد اکثر ۲۵۶ کاراکتر به جز کاراکتر / (Slash) می تواند باشد و هیچ گونه محدودیت زبانی ندارد. یعنی هر کاراکتر یونیکد (UTF-16) می تواند باشد.

حال کمی دقیقتر به محدودیت های **NTFS** می پردازیم:

اگرچه سیستم فایل حدود ۳۲ هزار کاراکتر یونیکد را پشتیبانی می کند، اما نام های مشخصی نمی توانند به عنوان نام فایل یا فولدر انتخاب شوند که از آن دسته میتوان به **\$Mft , \$Boot , \$Badclust**، ... اشاره نمود و یا نقطه (.) به تنهایی نمی تواند نام فایل انتخاب شود.

در تئوری، حداکثر اندازه پارتیشن **NTFS**، دو به دو به توان ۴۶ منهای یک کلاستر می تواند باشد. با این وجود حد اکثر اندازه پیاده سازی شده جهت پشتیبانی از پارتیشن **NTFS** در ویندوز اکس پی حرفه ای ۲ به توان ۳۲ منهای یک

کلاستر است. برای نمونه اگر از کلاسترهای ۶۴ کیلوبایتی استفاده کنیم حداکثر اندازه یک پارتیشن ۲۵۶ ترابایت (البته به اندازه ۶۴ کیلوبایت کمتر) و اگر از کلاسترهای معمول ۴ کیلوبایتی استفاده کنیم حداکثر اندازه آن ۱۶ ترابایت

(منهای ۴ کیلوبایت) خواهد شد.

البته به دلیل اینکه جدول پارتیشن ها در **MBR** تنها پارتیشن های حداکثر ۲ ترابایتی را پشتیبانی می کنند برای داشتن

پارتیشن های بیشتر از ۲ ترابایت در سیستم فایل **NTFS** میبایست از پارتیشن های پوبا استفاده نمود.