

Inode چیست ؟

در لینوکس تمام فایلها دارای بخش خاصی به نام inode هستند. خب در همین ابتدا مثالی برای بیان کردن مفهوم inode میزنیم. تمام ایرانی‌ها دارای شناسنامه یا کارت ملی هستند، که اطلاعاتی مانند نام و نام خانودگی، تاریخ تولد، مشخصات والدین و ... در آن نگهداری میشود. این کار شناسایی یک شخص را آسان میسازد. Inode را هم میتوانید یک شناسنامه برای یک فایل در نظر بگیرید. در inode یک فایل اطلاعاتی مانند تاریخ ایجاد، تغییر و دسترسی یک فایل، مالک فایل، گروه مالک، مجوز فایل و ... ذخیره میشود. همانطور که هر شناسنامه یک شماره دارد که با آن شناخته میشود، هر inode هم یک شماره منحصر به فرد دارد که برای اشاره به آن inode مورد استفاده قرار میگیرد. به این شماره منحصر به فرد inode number میگویند و میتوانید برای دسترسی به یک inode از آن استفاده کنید. حالا که با مفهوم کلی inode آشنا شدیم، اجازه دهید ببینیم که چه چیزهایی در یک inode نگهداری میشود:

size: اندازه‌ای که فایل بر روی دیسک اشغال میکند.

Device ID: مشخص کننده دستگاهی که فایل بر روی آن نگهداری میشود.

owner: هر فایل یک مالک دارد که میتواند کنترل کامل بر روی یک فایل داشته باشد.

group: گروهی که فایل به آن تعلق دارد.

permission: برای جلوگیری از دسترسی افراد غیر مجاز به فایل، برای آن مجوز تعریف میکنند.

flags: پرچم، علامتی است که بر روی فایل زده می‌شود. کاربر یا سیستم میتواند flag هایی را بر روی یک فایل جهت محافظت از آن اعمال کند (مثلاً محدود کردن استفاده و تغییر فایل)

access time: مشخص کننده آخرین زمان دسترسی به یک فایل است. منظور از دسترسی این است که مثلاً شما بر روی فایل کلیک کرده و محتویات آن را بخوانید. یا یک اسکرپت را از طریق خط فرمان اجرا کنید.

modification time: مشخص کننده آخرین زمان تغییر محتویات یک فایل است. منظور از تغییر کردن این است که شما یک فایل متن را باز کنید و محتویات آن را ویرایش کنید.

Change time: نشان دهنده آخرین زمانی است که اطلاعات inode یک فایل تغییر کند. مثلاً وقتی ما مجوز یک فایل را دستکاری کنیم، این فیلد بروز میشود. این فیلد را با modification time اشتباه نگیرید. اگر محتویات فایل را ویرایش کنیم، modification time بروز میشود. ولی اگر inode یک فایل را ویرایش کنیم، change time بروز میشود.

inode pointer: یک اشاره‌گر به بلاکهای از دیسک که فایل به صورت فیزیکی در آنجا نگهداری میشود.

Link: مشخص کننده تعداد لینکهای سخت به فایل است.

دقت کنید که 'نام فایل' در inode آن نگهداری نمیشود. (در مورد آن توضیح خواهیم داد)

وقتی که فایلی را درون یک دایرکتوری ایجاد میکنیم، یک نام به همراه یک شماره inode به آن فایل اختصاص می‌یابد. ممکن است فرض کنید که دایرکتوری که فایل را در آن ایجاد کردید، دربرگیرنده تمام فایل و اطلاعات مربوط به آن است (یعنی یک دایرکتوری را یک شئی مانند یک پارتیشن فرض کنید) ولی باید بگوییم که اینطور نیست. دایرکتوری‌ها هم یک نوع فایل هستند و بیشتر به فایل‌های متنی ساده شبیه هستند. اگر میتوانستیم یک دایرکتوری را با یک ویرایشگر متن باز کنیم، اطلاعاتی شبیه این میدیدیم:

shadow	۱۴۵۷۲۱۵
hosts	۵۸۷۱۲۵۶
modules	۲۱۴۵۸۹۶
fstab	۲۵۴۱۲۱۴

یعنی اینکه دایرکتوری‌ها یا بهتر بگوییم 'فهرست‌ها' فایل‌های متنی ساده‌ای هستند که از دو ستون تشکیل شده‌اند. در یک ستون شماره inode یک فایل و در ستون دیگر نامی که میتوان با آن به inode number دسترسی داشت (این نام برای دسترسی راحت‌تر به inode number است و همان نام فایل است) در مثال بالا inode ای با شماره ۲۵۴۱۲۱۴ داریم که نام passwd را برای دسترسی راحت‌تر به آن اختصاص داده‌ایم. میتوانیم خودمان یک سطر به فایل اضافه کنیم مثل زیر:

fstab	۲۵۴۱۲۱۴
mnttab	۲۵۴۱۲۱۴

در مثال بالا ستون inode number در هر دو سطر یکسان است. یعنی به یک inode number دو نام اختصاص داده‌ایم که هر دو به یک فایل اشاره میکنند. به این نامها hard link یا لینک سخت میگویند. یعنی لینک سخت چیزی نیست جز یک ورودی با inode number تکراری به یک فهرست. وقتی یک کاربر سعی میکند به فایلی به نام fstab دسترسی پیدا کند، سیستم عامل ابتدا fstab را به شماره inode تبدیل میکند و بعد از طریق آن شماره، اطلاعات را برای کاربر ارسال میکند. کار تبدیل یک اسم به شماره inode از طریق جدول inode یا inode table انجام میشود. این جدول در هنگام ایجاد سیستم فایل بر روی پارتیشن ساخته میشود. اگر یکی از این سطرها را پاک کنیم، سطر دیگر هنوز باقی است، پس فایل پاک نمیشود. ولی اگر هر دو سطر را پاک کنیم فایل پاک خواهد شد. دقت کنید یک لینکهای سخت را نمیتوان برای دایرکتوری‌ها ایجاد کرد و همچنین یک لینک سخت باید به یک i-number موجود در همان پارتیشن اشاره کند. چرا؟

Inode ها چه زمانی ایجاد میشوند؟

تا اینجا فهمیدیم یک inode یک ساختمان داده است که اطلاعاتی را در مورد یک فایل نگهداری میکند. حالا بیایید ببینیم که این inode ها چه زمانی ساخته می‌شوند. وقتی که ما دیسک سخت را پارتیشن بندی میکنیم، باید بر روی پارتیشن‌های ایجاد شده یک 'سیستم فایل' درست کنیم. سیستم فایل نحوه ذخیره داده‌ها بر روی یک پارتیشن را مشخص میکنند. سیستم فایل‌های مختلفی در لینوکس وجود دارد مثل ext3, ext4, btrfs, reiserfs, xfs, jfs و غیره. در هنگام ایجاد سیستم فایل بر روی پارتیشن، در صدی از حجم آن پارتیشن به جدولی اختصاص داده می‌شود که به آن جدول inode table میگویند. تمام inode ها در همین هنگام ساخته شده و در جدول قرار میگیرند. این جدول را میتوانید چیزی مانند یک دایرکتوری تصور کنید ولی با مقیاس بسیار بزرگ‌تر که تمام inode های موجود در آن پارتیشن را نگه داری میکند. دقت کنید که این جدول تعداد خاصی inode را در خود نگه داری میکند. یعنی تعداد inode ها در یک پارتیشن محدودیت دارد و نمیتوانیم هر چند تا inode که خواستیم داشته باشیم. خوب وقتی ما یک پارتیشن را تازه ساخته ایم و در آن هیچ فایلی کپی نکرده ایم، پس الان در داخل inode ها چه اطلاعاتی است؟ باید بگوییم که inode ها خالی هستند و وقتی شما یک فایل ایجاد کنید یکی از این inode ها پر میشود.

از مطالب بالا میتوان به یک نتیجه رسید: یک پارتیشن به دو حالت پر میشود:

۱- وقتی که به اندازه حجم آن پارتیشن، در آن اطلاعات ریخته شود. مثلاً در یک پارتیشن ۱۰ گیگابایتی، ۱۰ گیگ اطلاعات بریزیم.

۲- وقتی که جدول inode تمام شود و تمام inode ها استفاده شوند.

فرض کنید یک پارتیشن ۱۰۰ گیگابایتی دارید و هزاران فایل کوچک در آن کپی کرده اید. طوری که

جدول inode پارتیشن تمام شده است. ولی کل فایلها مثلاً ۴۰ گیگ حجم دارند. با اینکه هنوز ۶۰ گیگ فضای خالی وجود دارد، ولی عملاً غیر قابل استفاده است. چون دیگر inode ای نیست که به یک فایل جدید اختصاص دهیم!

برای روشن تر شدن مطلب خروجی دستور df -i را مشاهده کنید:

```
df -i $
Filesystem          Inodes    IUsed   IFree IUse% Mounted on
/ dev/sda1          1449984  268783 1181201 19% /
tmpfs               128088     5 128083    1% /lib/init/rw
udev               126750     712 126038    1% /dev
tmpfs               128088     1 128087    1% /dev/shm
dev/sda6           13688832 269500 13419332    2% /home/
```

همچنین دستور stat هم برای دیدن اطلاعات موجود در یک inode استفاده میشود.

```
stat /etc/passwd $
'\File: `/etc/passwd
Size: 1648      Blocks: 8      IO Block: 4096   regular file
Device: 801h/2049d    Inode: 567477    Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 0/ root)   Gid: ( 0/ root)
Access: 2012-02-02 23:13:16.000000000 +0330
Modify: 2012-01-18 01:55:59.000000000 +0330
Change: 2012-01-18 01:55:59.000000000 +0330
```