

به نام خدا

اسماعیل زارع

Fstab چیست ؟

شما برای اینکه بتوانید در سیستم عامل لینوکس از یک فایل سیستم یا پارتیشن استفاده کنید بایستی آن را در سیستم mount کنید ، یکی از فایل هایی که در خصوص mount کردن بسیار به شما و سیستم عامل کمک می کند فایلی به نام fstab است

Fstab مخفف File System Table محل برای اطلاعات ثابت سیستم فایل است. نگاشت های پارتیشن ها به دایرکتوری های مورد نظر به همراه گزینه های آنها در این فای ذخیره می شود. برای راه اندازی لینوکس حداقل سه دایرکتوری root یا ریشه، دایرکتوری boot و حافظه swap لازم است. در این فایل پس از راه اندازی سیستم حداقل دایرکتوری هایی که هستند این سه دایرکتوری گفته شده به همراه یک دایرکتوری منطقی به نام proc وجود دارند.

دستورهایی که با این فایل کار می کنند، مانند دستور mount است که برای اتصال یک پارتیشن یا دایرکتوری به دایرکتوری دیگر بکار می رود. فایل fstab در دایرکتوری /etc قرار دارد و برای مشاهده آن همه می توانند از دستور cat استفاده کنند ولی فقط کاربر root حق ویرایش آنرا دارد و می تواند از ویرایشگر vi استفاده کند. جای دیگری که از این فایل استفاده می شود در راه اندازی NFS یا Network File System است. در هر خط این فایل، یک سیستم فایل(پارتیشن) بصورت مجزا نوشته می شوند. فیلدها با چندین Space و یا Tab از هم جدا می شوند. اطلاعات درون این فایل ثابت یا Static است و هنگام بوت شدن سیستم این فایل خوانده می شود و پارتیشن های نوشته شده در این فایل، لود خواهند شد. پس یکی از فایل هایی که هنگام بوت شدن توسط سیستم خوانده می شود همین فایل است. هنگامی که سیستم را با دستور shutdown راه اندازی مجدد می کنید و طوری آنرا تنظیم کردید که پس از بوت شدن مجدد برنامه fsck سیستم فایل را بررسی کند و این دستور این فایل را می خواند.

```
/etc/fstab
```

# <file system>	<dir>	<type>	<options>	<dump>	<pass>
/dev/sda1	/	ext4	defaults,noatime	0	1
/dev/sda2	none	swap	defaults	0	0
/dev/sda3	/home	ext4	defaults,noatime	0	2

در بالا نمونه ای از فایل fstab را مشاهده می کنید اما این فایل همیشه دارای فیلدهای زیر است به فیلدهای زیر دقت کنید چون قرار است آنها را به ترتیب به شما معرفی کنیم .

فرمت کلی هر خط از این فایل بصورت ۶ فیلد زیر است:

<file system> <dir> <type> <options> <dump> <pass>

خوب ما به ترتیب در خصوص هر کدام از این فیلدها بصورت مجزا توضیح می دهیم:

۱. فیلد **file__system** این فیلد به سیستم فایل سیستمی یا پارتیشنی که قرار است mount بشود را معرفی می کند برای مثال `dev//sdb1//` یکی از این مثال ها است ، شما در این قسمت می توانید از UUID فایل سیستم و یا LABEL فایل سیستم مورد نظر نیز استفاده کنید ، در حالت عادی ما از قالب آدرس دهی دایرکتوری استفاده می کنیم به شکلی که مشاهده کردید.
۲. **dir** : هر فایل سیستم بایستی به شکل یک دایرکتوری به سیستم معرفی شود ، در این فیلد ما به سیستم عامل می گوییم که می خواهیم فایل سیستم ما با چه آدرسی در دسترس باشد ، در مثالهای بالا ما فایل سیستم هایی را در دایرکتوری ریشه و در دایرکتوری home متصل یا mount کرده ایم.
۳. **type** : همانطور که از نام این قسمت هم مشخص است این فیلد مشخص کننده نوع فایل سیستمی است که قرار است بر روی سیستم mount شود ، شما باید فایل سیستمی که پارتیشن شما با آن format شده است را در این قسمت وارد کنید ، قبلا در خصوص انواع فایل سیستم در لینوکس صحبت کرده ایم و فقط باید بدانید که در این قسمت انواع و اقسام فایل سیستم های لینوکس را می توانید وارد کنید ، اگر نمی دانید یا مطمئن نیستید که فایل سیستم شما از چه نوع است می توانید از کلمه auto استفاده کنید ، کلمه auto باعث می شود که خود سیستم عامل تلاش کند که نوع فایل سیستم مورد نظر را حدس بزند.
۴. **options** : این فیلد روشهای مختلفی که فایل سیستم می تواند بر روی سیستم قرار بگیرد و mount بشود را مشخص می کند ، دقت کنید که این [option](#) ها بصورت کلی بر اساس نوع فایل سیستم ها می تواند متغیر باشد ، برخی از مهمترین option هایی که در این قسمت می توان تعریف کرد حرف r یا rw می باشد که مشخص کننده فقط خواندنی بودن یا خواندنی و نوشتنی بودن فایل سیستم mount شده است ، شما با استفاده از کاراکتر ، می توانید چندین option بصورت همزمان اعمال کنید ، یکی از پرکاربرد ترین option ها در اینجا auto است که باعث می شود مقداری که شما در این قسمت وارد کرده اید بصورت خودکار بعد از restart شدن سیستم عامل لینوکس بر روی سیستم mount شود و نیازی به mount دستی مجدد نباشد ، اگر دقت کرده باشید شما می توانید این option ها را هم بصورت دلخواه بر عهده سیستم عامل بگذارید ، با استفاده از قرار دادن گزینه defaults ما به سیستم عامل می گوییم که از option های پیشفرض استفاده کنید که خود سیستم عامل در نظر می گیرد.
۵. **dump** : اما این فیلد حرف و حدیثهای فراوانی دارد و امروزه تقریبا منسوخ شده است ، dump یک مکانیزم از رده خارج شده برای backup گیری از فایل سیستم است. در لینوکس ابزاری به نام dump وجود دارد که برای backup گیری استفاده می شود و این فیلد مربوط به این ابزار است ، ابزار dump این فیلد را بررسی می کند و با استفاده از اعدادی که در این فیلد قرار گرفته اند تصمیم می گیرد که آیا این فایل سیستم باید Backup گرفته شود یا خیر ؟ مقادیری که در این فیلد می تواند قرار بگیرد ۰ و ۱ هستند ، در صورتیکه مقدار dump ما عدد یک باشد dump از فایل سیستم backup می گیرد و در

صورتیکه مقدار ۰ وجود داشته باشد از فایل سیستم backup گرفته نمی شود. امروزه در بسیاری از توزیع های لینوکس ابزار dump نصب نشده است و به همین دلیل اکثر کاربران این مقدار را ۰ قرار می دهند.

۶. **pass** : ابزار fsck برای بررسی سلامت و اسکن کردن فایل سیستم استفاده می شود ، فایل سیستم ها دچار مشکل می شوند و سیستم بصورت خودکار بعد از مدتی آنها را اسکن می کند ، فیلد pass مشخص کننده اولویت یا order یا priority برای اسکن کردن فایل سیستم ها توسط ابزار fsck است ، مقادیری که در این قسمت می تواند قرار بگیرد ۰ و ۱ و ۲ هستند ، طبیعتا فایل سیستم ریشه یا root بایستی دارای بالاترین اولویت اسکن شدن باشد که در حالت عادی عدد ۱ بالاترین اولویت اسکن است البته اگر فایل سیستم شما btfs نباشد ، در اینصورت عدد ۰ بالاترین اولویت در نظر گرفته می شود. سایر فایل سیستم هایی که قرار است بررسی شوند عدد ۲ را خواهند داشت ، اگر فایل سیستم شما عدد ۰ را در این قسمت داشته باشد توسط fsck اسکن نخواهد شد. از طرفی اگر دو فایل سیستم عدد pass یکسانی داشته باشند یعنی بصورت همزمان و با اولویت یکسان اسکن خواهند شد

مهمترین گزینه های بخش option در fstab

auto و **noauto** : گزینه پیش فرض auto است و به این معنی است که سیستم فایل یا پارتیشن باید هنگام بوت شدن بصورت خودکار mount شوند در غیر اینصورت شما باید اینکار را با کاربر ریشه و بعد وارد شدن به سیستم و بصورت دستی با دستور mount انجام دهید. همانطور که گفته شد این حالت پیش فرض است و اگر می خواهید پارتیشن بصورت خودکار mount نشود باید از گزینه noauto استفاده کنید.

user و **nouser** : گزینه ای کاربردی و مهم user. اجازه می دهد که کاربران معمولی هم بتوانند mount کنند. ولی اگر می خواهید که کاربران معمولی نتوانند این کار را انجام دهند از nouser استفاده کنید nouser. بصورت پیش فرض است. در خط زیر گزینه پیش فرض nouser است ولی اگر می خواهید کاربران معمولی هم پارتیشن را به یک دایرکتوری دیگر mount کنند باید user را بعد default اضافه کنید. برای جلوگیری از صدمه زدن به سیستم توسط کاربران معمولی و تازه کار گزینه nouser صورت پیش فرض استفاده شده است.

exec و **noexec** : گزینه exec باعث می شود تا بتوان فایل های باینری را اجرا کرد noexec. برعکس exec است و از اجرای فایل های باینری جلوگیری می کند. این گزینه زمانی بکار می آید که نمی خواهید فایل های باینری که در پارتیشن دارید در نقطه اتصال اجرا شوند. پیش فرض exec است. فرض کنید یک دایرکتوری را از یک سیستم عاملی مانند FreeBSD در شبکه با NFS اشتراکی می کنید و این دایرکتوری شامل فایل های معمولی و باینری است و شما نمی خواهید که فایل های باینری اجرا شوند پس از گزینه noexec استفاده می کنید.

ro/rw : گزینه ro یعنی Read-Only و فقط کاربران حق دیدن فایل ها و محتوا را دارند ولی نمی تواند دایرکتوری یا فایل جدید ایجاد یا در یک دایرکتوری NFS چیزی کپی کنند. اما rw یعنی Read-Write که هم اجازه خواندن و هم اجازه نوشتن را می دهد که به این معنی است می توانید فایل و دایرکتوری هم ایجاد کنید. حالت پیش فرض rw است.

sync و **async** : این دو گزینه به ورودی و خروجی اشاره می کنند. (Input/Output) گزینه sync مخفف synchronously است و اشاره به همگام سازی I/O دارد. بطور مثال با فلاپی ها بکار می رود و مفهوم آن این سات که وقتی شما یک فایل را با

دستور `cp` روی فلاپی کپی می کنید، تغییرات بصورت همزمان با اجرای دستور روی فلاپی بصورت فیزیکی نوشته می شوند. در نقطه مقابل آن گزینه `async` که مخفف `asynchronously` است. وقتی `async` را استفاده می کنید تغییرات مدتی پس از اجرای دستور `cp` روی فلاپی بصورت فیزیکی نوشته می شوند. البته این حالت بد نیست و برخی مواقع مفید هم هست ولی شاید باعث بروز مشکلاتی هم بشود. اگر شما بدون `unmounting` کردن فلاپی آنرا بیرون بکشید چون کپی ها مدتی پس از اجرا شدن دستور روی فلاپی نوشته می شوند ممکن است هنوز کپی نشده باشند. پس بطور خلاصه برای استفاده از `async` می توان این طور گفت که پس از اجرا شدن دستور ابتدا آنرا `unmount` کنید `async`. بصورت پیش فرض است.

در گزینه های بالا گفتیم برخی ها بصورت پیش فرض هستند گزینه دیگری به نام `default` وجود دارد که پیش فرض هایی مانند `async, exec, rw, auto` را در بر می گیرد.