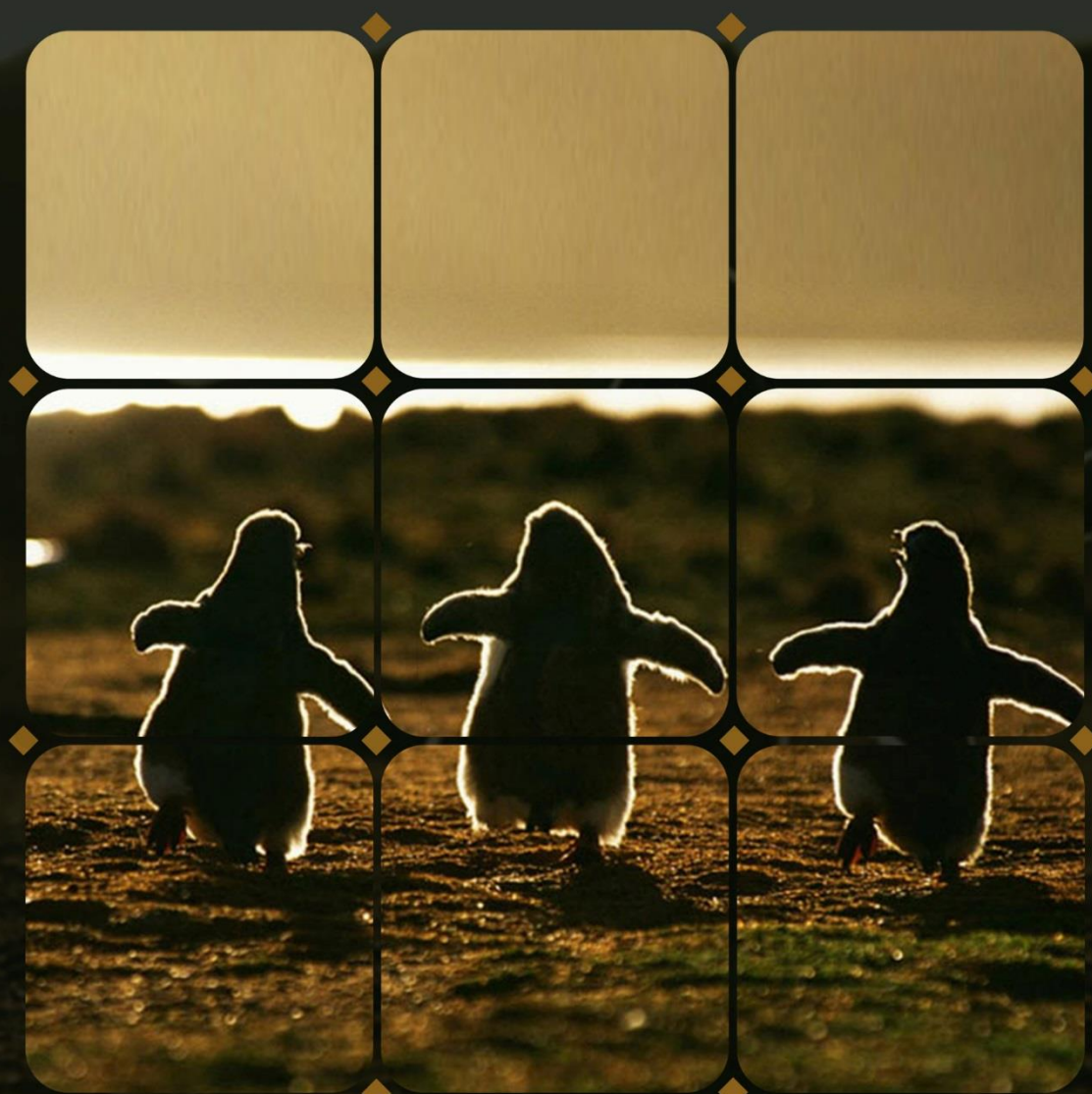


system file in linux

ساختار سیستم فایل در لینوکس



برگرفته از کتاب لینوکس برای همه
نویسنده: حسین سیلانی

ساختار فایل در لینوکس

نکته های آموزشی سیستم عامل لینوکس

برگرفته از کتاب

لینوکس برای همه از همین نویسنده

ویرایش اول ۱۳۹۷

نشر آزاد

نشر آزاد

نام کتاب : معرفی ساختار فایل در لینوکس

نویسنده : حسین سیلانی

ویراستار و طراح جلد: ا. قاسمی نژاد

ناشر : نشر آزاد

نوبت چاپ : اول

تاریخ نشر: ۱۳۹۷

شمارگان : ۱۰۰۰۰ نسخه

قیمت : ۸۰۰۰ تومان

شابک : ۹۶۵-۷۵۶-۵۶۸۷-۶۸-۳



• درباره نویسنده:

حسین سیلانی، فارغ التحصیل مقطع کارشناسی ارشد رشته امنیت اطلاعات با تجربه بیش از ۱۰ سال سابقه تدریس در زمینه فناوری اطلاعات و نرم افزارهای کامپیوتری، دارای مدرک مربیگری از سازمان آموزش فنی و حرفه ای کشور می باشد. از دیگر فعالیت های وی می توان به طراحی و توسعه ابزارها، محیط های متن باز سیستم عامل لینوکس و همچنین تولید محتوا در زمینه های مربوطه از جمله: نویسندگی، طراحی فلش کارت های لینوکسی، تهیه فیلم های آموزشی و... اشاره کرد.

پاییز ۹۷

فایل و فهرست‌ها

نمایش ساختار فایل و فهرست در لینوکس

در لینوکس به نحوه نمایش ساختار فایل‌ها و فهرست‌ها سیستم فایل می‌گویند که به معنی استاندارد سلسله مراتبی سیستم فایل می‌باشد. این استاندارد برای این ارائه شده تا تعیین کند که ساختار فایل و فهرست‌ها در لینوکس به چه صورت سازماندهی شود.

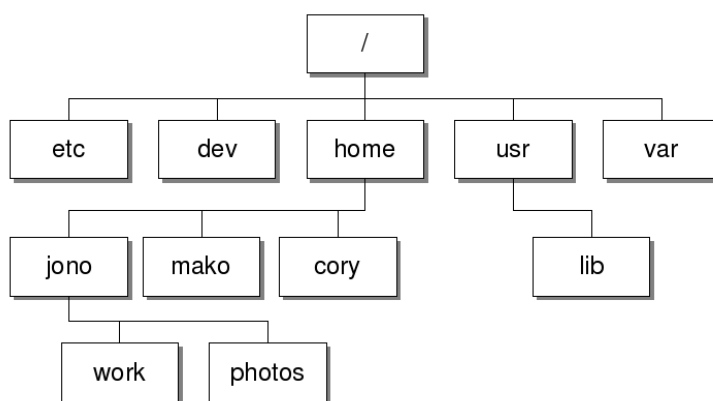
کاربردهای fsh

نرم افزارها و کاربران را قادر می‌سازد که در مسیر از پیش تعیین شده فایل‌ها و فهرست‌های خود را نصب کنند. تعیین اصول راهنما برای هر ناحیه از سیستم فایل تعیین حداکثر اندازه و تعداد فایل‌ها و فهرست‌های مورد نیاز سیستم

🐱 منظور از اشتراک پذیر این است که همه فایل‌ها در روی یک هاست آقراری گیرد. منظور از غیر اشتراک پذیر این است که تنها بر روی همان هاست قابل دسترسی است.

🐱 منظور از استاتیک بودن فایل‌ها این است که تمامی فایل‌ها بدون اجازه و دسترسی مدیر سیستم قابل ویرایش نیستند و منظور از فایل‌های متغیر فایل‌هایی هستند کاربر معمولی قابلیت تغییر و دستکاری کردن آنها را دارد.

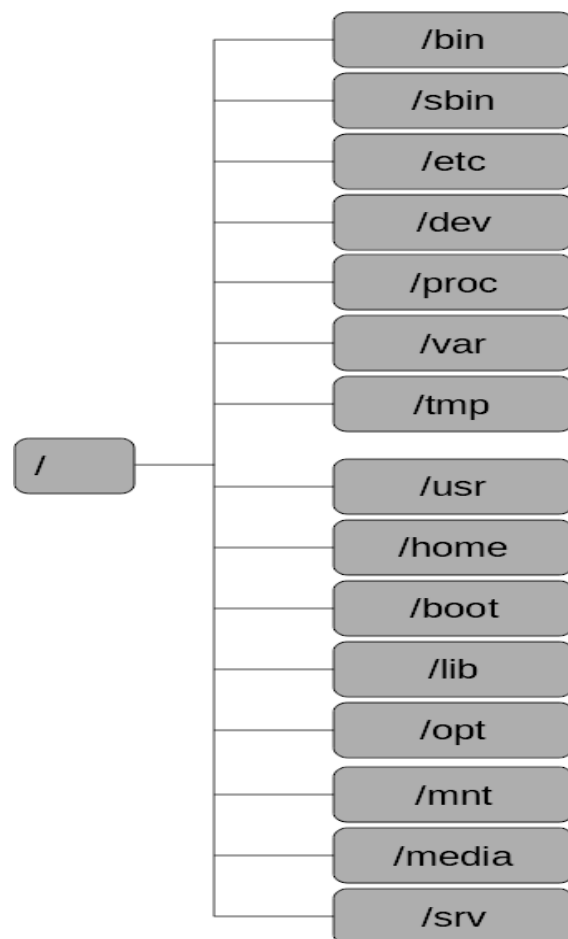
در لینوکس فهرست ریشه به چند بخش تقسیم می‌شود که هر بخش بر روی دیسک یا پارتیشن مربوط به خود قرار می‌گیرد. سه فهرست بزرگ و مهم در لینوکس `/home` , `/var` , `/usr` می‌باشد که این فهرست‌ها دارای زیرمجموعه‌های بی شماری می‌باشند.



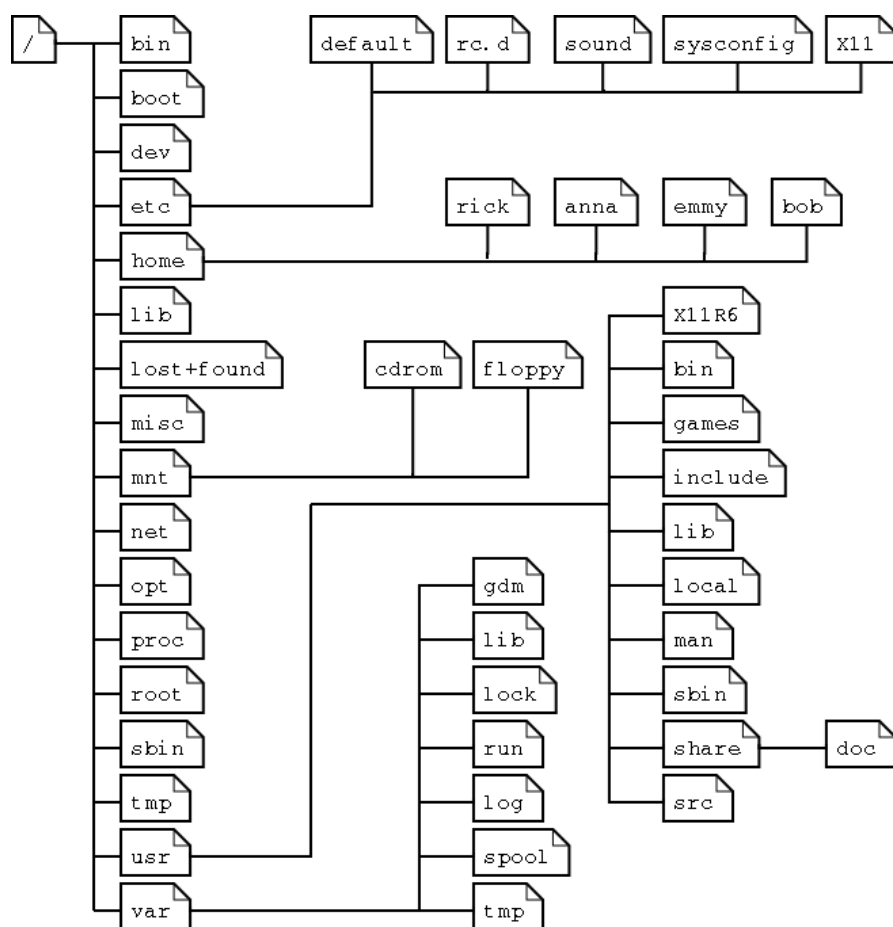
ساختار فهرست ریشه

¹File Systems Hierarchy Standard (Fhs)

²Host



ساختار فهرست ریشه



ساختار فهرست ریشه

ساختار فهرست در لینوکس

در سیستم عامل لینوکس مفهوم ساختار درختی¹ بیان می شود. زیرا لینوکس برپایه یونیکس بوده و مفهوم درخت از آنجا گرفته شده است و دیگر خبری از درایونیست. ساختار فایل به صورت درختی وارونه است که ریشه این درخت به نام root است. این ریشه در بالاترین بخش درخت قرار دارد و بقیه بخش ها زیر مجموعه آن می باشد.

هرچیزی که فایل نباشد یک پردازش به شمار می آید. توجه داشته باشید که کرنل لینوکس در تمامی پارتیشن ها توزیع شده و در یک بخش وجود ندارد.

فهرست ریشه

فهرست ریشه دارای فهرست های زیرین زیادی می باشد که به طور کلی آنها را معرفی می کنیم :

/bin : user binaries

در این فهرست بیشتر دستورهای مدیریتی کاربر سیستم که قابل دسترسی برای همه کاربران و از جمله فایل های باینری قابل اجرا هستند، قرار می گیرد.

/sbin: system binaries

این فهرست همانند فهرست قبلی شامل دستورهای مدیریتی کاربر سیستم و هم چنین دستورهای مدیریتی مدیر سیستم را شامل می شود.

/etc: configuration files

این فهرست مثل سیستم عصبی بدن عمل می کند. کلیه تنظیمات سیستم که در قالب فایل وجود دارد در این فهرست نگهداری و ذخیره می شوند. فهرست های زیادی به همراه فایل های تنظیمات در این مکان وجود دارد که به طور چکیده به معرفی آنها می پردازیم:

فهرست های موجود در /etc

/etc/x11/ : تمامی تنظیمات مربوط به پنجره موسوم به x window که تنظیمات رابط گرافیکی کاربران است در این فهرست ذخیره می شود.

/etc/apt/sources.list

محل ذخیره لیست منابع مربوط به سرورهای بروزرسانی سیستم عامل و مخازن می باشد.

/dev : device files

فهرستی برای نگهداری فایل های سخت افزاری و وسیله های متصل به سیستم است.

/proc: process information

این فهرست محل نگهداری اطلاعات مربوط به پردازش های در حال اجرا و مخفی سیستم است. هم چنین اطلاعات منابع سیستم نیز در این فهرست موجود است.

¹ Trees (Data Structures)

² process

/var :variable files

این فهرست معمولاً جداگانه ایجاد می شود و محل نگهداری اطلاعاتی هست که مرتباً در حال تغییر و بروزرسانی هستند و به عبارتی متغیر هستند.
فهرست های موجود در این مسیر:

/var/backups
/var/cache
/var/crash
/var/games
/var/log
/var/mail
/var/spool

/tmp :temporary files

این فهرست برای نگهداری فایل های موقت¹ سیستم و کاربران است و معمولاً فایل های موجود در آن، پس از راه اندازی مجدد² سیستم پاک می شوند.

/usr: user programs

در این فهرست اطلاعاتی مربوط به کتابخانه ها، سندها، راهنماها، سورس کدها، برنامه های نصب شده و غیره می باشند.

/home : home directories

این فهرست محل نگهداری تمامی اطلاعات شخصی³ کاربرانی است که به سیستم عامل login کرده اند. مثلاً
/home/ali

/boot: boot loader files

محل نگهداری فایل های بوت سیستم است. فایل هایی که سیستم نیازمند آنها برای راه اندازی سیستم است و همچنین فایل هایی که در زمان بوت به آنها نیاز دارد

/lib: system libraries

محل نگهداری فایل های کتابخانه برنامه ها است. این فایل ها مثل ماژول های کرنل، کتابخانه های اشتراکی (مثل کتابخانه های زبان C) که نیاز است به همراه سیستم بارگذاری شوند در این محل قرار می گیرند.

/opt: optional add-on applications

محل نگهداری برنامه های add-on که برنامه های کمکی برای دیگر نرم افزارها هستند و همیشه به صورت پیش فرض نصب نیستند. مثل staroffice یا kget.

¹Temporary

² Reboot

³Privacy

/mnt: mount directory

محل نگهداری نقطه‌های متصل شده^۱ به طور موقت به سیستم مثل فلش دیسک یا یک هارد خارجی^۲ است.

/media : removable media devices

این فهرست برای محل نگهداری وسایل جانبی که قابلیت ذخیره سازی^۳ را دارند، مثل فلش دیسک ها، دیسک گردان ها و غیره می باشد.

/srv: service data

محل نگهداری سرویس‌های که مربوط به سرورها می باشد و معمولاً در پس زمینه^۴ در حال اجرا هستند.

دیگر سیستم فایل ها:

/lost+found

برای هرپارتیشن در سیستم یک فهرست برای آن در فهرست lost+found ساخته می شود. در صورتی که اطلاعات اصلی در این مسیر موجود باشد عمل بازیابی صورت گرفته و سیستم اجرا می شود و در غیر این صورت کاربر باید برنامه و یا سیستم خود را دوباره نصب و تنظیم نماید.

/sys

در توزیع‌های جدید لینوکس از این فهرست برای نگهداری سیستم فایل مجازی استفاده می شود.

/boot/vmlinuz

محل برای نگهداری فایل‌های کرنل لینوکس

/etc/fstab

محل نگهداری هارد دیسک‌هایی که به سیستم متصل شده اند، قرار دارد.

/etc/default/grub.conf

محل نگداری فایل تنظیمات راه انداز بوت سیستم grub:

/etc/lilo.conf

محل نگداری فایل تنظیمات راه انداز بوت سیستم lilo:

/etc/init.d

محل نگداری اسکریپت‌های start/stop سرویس ها

/etc/hosts

محل نگهداری آدرس‌های ipهاست‌های مجازی و دیگر آدرس‌های ip

/etc/passwd

محل نگهداری پسوردهای کاربران سیستم

/etc/printcap

محل ذخیره اطلاعات مربوط به پرینترهای موجود بر روی سیستم عامل

/etc/x11

محل ذخیره تنظیمات پنجره‌های X. تنظیمات رابط‌های گرافیکی سیستم

/etc/resolv.conf

^۱Mount

^۲External

^۳ Removable Devices

^۴ Background

محل ذخیره تنظیمات شبکه مربوط به دامین ها و سرور. (Domain Name Servers (DNS).

/usr/share

محل ذخیره فهرست‌های اشتراک گذاشته شده در میان سیستم ها

/proc/cpuinfo

محل ذخیره اطلاعات مربوط به CPU

/proc/filesystems

محل ذخیره اطلاعات مربوط به سیستم فایل که در حال حاضر استفاده می شود.

/proc/interrupts

محل ذخیره اطلاعات درباره رخ دادهای سیستم

/proc/ioprocs

محل ذخیره تمام اطلاعات آدرس پورت‌های ورودی و خروجی که توسط سیستم استفاده می شوند.

/proc/meminfo

محلی برای ذخیره میزان حافظه استفاده شده توسط سیستم

/proc/modules

محل ذخیره ماژول‌های کرنل

/proc/mount

محلی برای ذخیره اطلاعات مربوط به وسایل و رسانه‌های متصل شده به سیستم

/proc/stat

محل ذخیره چکیده ای از وضعیت سیستم در قالب آمار

/proc/swaps

محل ذخیره اطلاعات مربوط به حافظه مجازی سیستم

/version

محلی برای نگهداری اطلاعات مربوط به نسخه لینوکس

/var/log

محل ذخیره رخ داده‌ها و وقایع سیستم

مسیرها

در لینوکس به طور کلی دو نوع مسیر داریم. منظور ما از مسیر، مسیر فایل تا فهرست ریشه بوده یا مسیر از فهرست ریشه آغاز شده و تا نام فایل مورد نظر پیش می رود. به عنوان مثال :

/home/ali/bin

(۱) مسیر مطلق

مسیری است که با نماد اسلش^۱ شروع می شود و ادامه دارد.

/usr/local/bin

(۲) مسیر نسبی

مسیری است که با نمادپیش از نماد slash با یک نقطه . یا دو نقطه .. آغاز می شود و ادامه دارد.

^۱ Slash

./configure

از مسیرهای نسبی برای کامپایل کردن و یا اجرا کردن سورس کدهای برنامه استفاده می کنیم. که برای مثال دستور ./configure به فهرست جاری اشاره می کند و برای تنظیم سورس کد به کار گرفته می شود.

فایل ها

همان گونه که گفته شد در لینوکس هر چیزی فایل محسوب می شود حتی فهرست ها نیز فایل می باشند. بسیاری از فایل ها در حالت عادی به نام Regular Files نامیده می شوند در جدول زیر نماد فایل ها را مشاهده می نمایید که با نمایش هرحرف در کنار نام فایل در پنجره ترمینال نوع فایل مشخص می شود.

```
$ ls -l
```

خروجی

```
-drw-rw-r-- 1 ali ali 41472 17:56 linux.doc
```

Symbol	Meaning
-	Regular File
d	Directory
l	Link
c	Special File
s	Socket
p	Named Pipe
b	Block Device

سیستم نامگذاری فایل ها در لینوکس

اندازه فایل ها

هر فایل در ساختار کامپیوتر دارای اندازه ای می باشد که با یک عدد و در قالب بایت نشان داده شده و تعیین کننده اندازه آن فایل و میزان استفاده از فضای رسانه ذخیره سازی(مثل: هارדיسک، فلش مموری و غیره) را نشان می دهد

پسوند فایل ها

پسوند فایل نشانگر نوع فایل و اینکه در چه قالبی ارائه شده است را نشان می دهد. این پسوند پس از نام فایل پس از نماد نقطه^۱ می آید. بسیاری از سیستم فایل ها دارای محدودیت هایی از لحاظ تعداد کارکترها در پسوند فایل ها هستند. پسوندهای فایل معمولاً ۳ یا ۴ کاراکتر می باشند مثل : txt-jpeg.mp4

نام فایل ها در لینوکس

در لینوکس نام فایل ها پس از نماد(/) می آید و به فهرست ریشه بر می گردند. نام فایل ها به صورت حروف الفبایی و به عنوان رشته بیان می شود که اکثراً به صورت حروف کوچک نوشته شده است. در ابتدا در

¹ Dot

سیستم‌های یونیکس اندازه نام فایل‌ها محدود به ۱۴ بایت بودند یعنی ۱۴ کاراکتر. اما امروز تا ۲۵۵ بایت (کاراکتر) دارای اندازه هستند.

قواعد نامگذاری فایل‌ها در لینوکس

- (۱) برای نامگذاری باید از حروف‌های الفبا، اعداد ۰ تا ۹، نماد نقطه و خط تیره و خط زیر استفاده نمایید.
- (۲) نام فایل‌ها با نماد _ شروع نمی‌شود. معمولاً در سیستم عامل‌ها نام فایل‌ها با حروف بزرگ و کوچک متفاوت و گاهی حساس^۳ می‌باشند. مثل test.txt و Test.txt

کاراکترهای رزرو شده برای نام گذاری

کاراکتر	نام	محل استفاده
/	slash	برای نامگذاری فهرست ریشه و جدا کننده مسیر و نام فایل‌های می باشد.
\	backslash	جدا کننده مسیر ها در سیستم عامل‌های ویندوز می باشد.
?	question mark	تنها مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس هستید.
%	percent	در زبان‌های برنامه نویسی استفاده می شود و مجاز به استفاده در نامگذاری نیستید.
*	star	تنها مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس هستید.
:	colon	تنها مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس هستید.
	pipe	تنها مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس هستید.
"	quote	برای تعیین اول و پایان رشته ها می باشد و در هیچ سیستمی قابل استفاده نیست.
<	less than	تنها مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس هستید.
>	greater than	تنها مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس هستید.
.	dot	مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس و ویندوزی هستید.
	space	مجاز در نامگذاری در سیستم عامل‌های بر پایه یونیکس و ویندوزی هستید.

جدول کاراکترهای غیرمجاز در نام گذاری فایل‌ها

رنگ‌های فایل در لینوکس

برای هر فایل یک رنگ در نظر گرفته شده است. در جدول زیر نام رنگ و موارد کاربرد آن بیان شده است:

سبز: رنگ فایل‌های اجرایی تحت لینوکس

آبی: رنگ فهرست‌ها

مژنتا^۴(ارغوانی): رنگ مربوط به انواع فرمت‌های تصویری (jpg, gif, bmp, png, tif) و socket

فیروزه‌ای: رنگ لینک‌های نمادین

زرد: برای Pipeها

رنگ سفیدچشمک زدنبا پس زمینه قرمزپررنگ: لینک‌های نمادین بی خانواده^۵

رنگ متن زردپررنگ، با پس زمینه سیاه: مربوط به Block Device Driver

^۱Hyphen

^۲Underline

^۳Sensitive

^۴Magenta

^۵Orphaned

رنگ سفیدچشمک زدنها پس زمینه قرمزپررنگ: لینک‌های گم شده با فایل‌های اشاره شده
قرمز: رنگ مربوط به فایل‌های آرشیو (tar,gz,zip,rpm)

کار با فایل ها و فهرست‌ها

دستور ls

دستور ls مخفف دستور list که یکی از پرکاربردترین دستورهای لینوکس است

مشاهده لیست فایل ها و فهرست ها به ترتیب حروف الفبا^۱ ولی به صورت صعودی^۲

```
$ ls
```

مشاهده فایل ها و فهرست ها به ترتیب حروف الفبا ولی به صورت نزولی^۳

```
$ ls -r
```

مشاهده لیست فایل ها و فهرست ها به همراه اندازه و نوع دسترسی هاشان

```
$ ls -l
```

مشاهده لیست فایل‌های یک فهرست یا مسیر خاص

مشاهده لیست فایل‌ها و فهرست‌ها یا یک مسیر خاص با تمامی جزئیات (حرف l کوچک)

```
$ stat directory
```

مثال:

```
stat /var/log/
```

```
root@hossein-pc: ~
root@hossein-pc:~# stat /var/log
  File: '/var/log'
  Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: 801h/2049d Inode: 172036      Links: 12
Access: (0775/drwxrwxr-x)  Uid: (  0/   root)   Gid: ( 108/  syslog)
Access: 2016-04-21 02:45:52.000000000 +0430
Modify: 2017-11-29 04:48:50.744000000 +0330
Change: 2017-11-29 04:48:50.744000000 +0330
 Birth: -
root@hossein-pc:~#
```

مشاهده اندازه فایل ها

برای مشاهده اندازه فایل و فهرست ها برحسب کیلوبایت^۴ از سوئیچ^۵ lh استفاده می شود. که سوئیچ h برای نمایش بهتر و خروجی خواناتر برای کاربر می باشد.

```
$ ls -lh
```

^۱ Alphabet

^۲ Ascendant

^۳ Descent

^۴ Kb

^۵ List Human View

تعیین نوع خروجی بر حسب نوع مقادیر دلخواه

```
$ ls -l --block-size=k  
$ ls -l --block-size=m  
$ ls -l --block-size=g
```

نمایش فایل‌های مخفی

برای دیدن فایل‌های مخفی در مسیر جاری باید از سوئیچ `a` استفاده نمایید.

```
$ ls -a
```

`.bashrc documents .gconfd install.log .nautilus .pulse-cookie`
لیست خروجی که در کنار آنها نماد نقطه¹ دیده می‌شود به این معنی است که فایل یا فهرست مخفی هستند.
مشخصه فایل‌ها و فهرست مخفی در لینوکس نماد نقطه در کنار نام آنها می‌باشد.

مدیریت فایل و فهرست‌ها

یکی از این دستورها، دستور `cat` می‌باشد. `cat` مخفف `Concatenate` یعنی الحاق و بهم پیوستن است. برای مشاهده محتوای فایل ابتدا دستور `cat` و سپس نام فایل خود را به همراه مسیر آن در برابر آن بنویسید. افزون بر یک فایل می‌توان محتوای چند فایل را با هم مشاهده کرد.

```
$ sudo cat <options><file>  
$cat نام فایل  
$cat filename.txt
```

مثال :

```
$ sudo cat /etc/ntp.conf
```

خروجی:

```
permit time synchronization our time resource but do not  
permit the source to query or modify the service on this system  
restrict default kod nomodify notrap nopeer noquery  
restrict -6 default kod nomodify notrap nopeer noquery
```

مشاهده محتوای چند فایل

برای مشاهده محتوای چند فایل کافی است نام چند فایل را با فاصله پشت سر هم تایپ کرده تا محتوای آنها نمایش داده شود.

```
$ cat filename1 filename2  
$ sudo cat test; cat test1; cat test2
```

```
this is test file  
this is test1 file.  
this is test2 file.
```

¹Dot

²Path

ساخت یک فایل جدید

یکی دیگر از کاربردهای cat علاوه بر مشاهده محتوای محتوای یک فایل، ایجاد فایل است. به این صورت که در برابر دستور cat نام فایل جدید را به همراه نماد هدایت گر > وارد نمایید. سپس فایل را ایجاد و منتظر وارد کردن مقادیر خود می باشد. برای بستن و ذخیره آن کلید ترکیبی ctrl + c را وارد نمایید.

```
cat > a.txt
```



a.txt

```
root@hossein-pc: /home/hossein/Desktop
root@hossein-pc:/home/hossein/Desktop# cat > a.txt
asdasd
asdasd
^C
root@hossein-pc:/home/hossein/Desktop#
```

مشاهده محتوای فایل ها با نماد کنترل صفحات

بسیاری از دستورها در لینوکس دارای خروجی زیادی هستند که برای مدیریت صفحات در خروجی از دو آرگومان ورودی less و more استفاده می کنیم. پیش از استفاده این آرگومان باید از نماد | که pop more می باشد، استفاده کرد.

```
$cat /etc/passwd | more
```

```
$cat /etc/passwd | less
```

مشاهده محتوای فایل با نمایش شماره خطوط

برای قرار گرفتن یک شماره برای هر خط از سوئیچ n استفاده می نمایم.

```
cat -n filename
```

```
$ sudo cat -n /etc/ntp.conf
```

خروجی:

```
1 # permit time synchronization our time resource but do not
2 # permit the source to query or modify the service on this system
3 restrict default kod nomodify notrap nopeer noquery
4 restrict -6 default kod nomodify notrap nopeer noquery
```

مشاهده محتوای فایل با نمایش تعداد خطوط خالی

به منظور نمایش خطوط خالی یک فایل متنی از سوئیچ b استفاده می نمایم.

```
$ cat -b نام فایل
```

```
$ sudo cat -b /etc/ntp.conf
```

خروجی:

```
1 # permit time synchronization our time resource but do not
2 # permit the source to query or modify the service on this system
```

```
3 restrict default kod nomodify notrap nopeer noquery
4 restrict -6 default kod nomodify notrap nopeer noquery
```

ارسال محتوای یک فایل به فایل دیگر

در صورتی که از یک هدایت گر < استفاده شود، محتوای جدید جایگزین محتوی قبلی خواهد شد.

```
$ cat filename1>filename2
```

ارسال محتوای یک فایل به فایل دیگر

در صورتی که از دو هدایت گر << استفاده شود، محتوای جدید به آخر محتوی قبلی اضافه خواهد شد.

```
$ cat filename1>>filename2
```

ارسال محتوای دو فایل به درون یک فایل

```
$ cat filename1filename2>> filename3
```

```
$ sudo cat f1
welcome1
```

```
$ sudo cat f2
welcome2
```

```
$ sudo cat f3
welcome3
```

```
$ sudo cat f1 f2 f3 > f4
```

```
$ sudo cat f4
welcome1
welcome2
welcome3
$ sudo
```

به منظور اضافه کردن نماد \$ به پایان هر خط در یک فایل متنی باید از سویچ e استفاده کرد.

```
$ cat -e filename1
```

```
$ sudo cat -e /etc/hosts
```

```
root@hossein-pc: ~
root@hossein-pc:~# cat -e /etc/hosts
127.0.0.1      localhost$
127.0.1.1      hossein-pc$
$
# The following lines are desirable for IPv6 capable hosts$
::1          ip6-localhost ip6-loopback$
fe00::0 ip6-localnet$
ff00::0 ip6-mcastprefix$
ff02::1 ip6-allnodes$
ff02::2 ip6-allrouters$
root@hossein-pc:~#
```


حذف یک فهرست

برای حذف یک فهرست از طریق ترمینال از دستور `rmdir` که مخفف `remove directoy` است استفاده می کنیم. برای این منظور در برابر دستور مسیر و نام فهرست را وارد نمایید.

```
$rmdir نام فهرست
$rmdir /dev/abc/edf
```

در صورتی که فهرست خالی نباشد با خطای زیر رو برو خواهید شد:

```
root@hossein-pc: ~
root@hossein-pc:~# ls
ali a.txt
root@hossein-pc:~# rmdir ali
rmdir: failed to remove 'ali': Directory not empty
root@hossein-pc:~#
```

برای حذف فهرست های پر از سوئیچ `--ignore-fail-on-non-empty` استفاده نمایید.

```
root@hossein-pc: ~
root@hossein-pc:~# rmdir ali --ignore-fail-on-non-empty
root@hossein-pc:~#
```

حذف فایل

برای حذف یک فایل موردنظر کافی است نام و مسیر آن را وارد نمایید.

```
$rm نام فایل
$ rm computer.log
```

حذف چندین فایل با پسوند مشخص

```
$ rm *.log
```

حذف یک فهرست خالی با سوئیچ `d`

```
$ rm -d documents
```

حذف یک فهرست دارای اطلاعات (خالی نمی باشد) با سوئیچ `r`

```
$ rm -r documents
```

پرسیدن و تاییدیه^۱ حذف در زمان حذف فهرست با سوئیچ `i`

```
$ rm -ri documents
```

ساخت فایل

برای ایجاد فایل^۲ از طریق محیط ترمینال از دستور `touch` استفاده می کنیم. پس از دستور نام فایل دلخواه را وارد نمایید. افزون بر یک فایل، قادر خواهید بود چندین فایل را باهم ایجاد نمایید.

^۱Prompt

^۲ Create

```
$ touch filename
```

ساخت چندین فایل خالی و هم زمان

```
$ touch filename1 filename2 filename3 filename4
```

تغییر آخرین زمان دسترسی و تغییر فایل با سوئیچ **-a**

```
$ touch -a test
```

تغییر زمان تغییر کردن فایل¹ با سوئیچ **m**

```
$ sudo touch -m ali
```

ساخت فهرست در مسیر دلخواه

برای ایجاد کردن یک فهرست از طریق ترمینال از دستور **mkdir** که مخفف Make Directory می باشد استفاده می شود. کافی است پس از دستور نام فهرست دلخواه خود را وارد نمایید.

```
$ mkdir نام_فهرست
```

تغییر فهرست در مسیر مشخص

از دو دستور زیر برای ورود به یک فهرست خاص استفاده نمایید.

```
$ cd /home/
```

```
$ cd /home/
```

تعیین دسترسی فایل و فهرست ها

در لینوکس بسیاری از فایل ها و فهرست ها دارای دسترسی خاص و پیش فرضی هستند و در برخی مواقع کاربر برای دسترسی و یا اجرای آنها باید مجوز لازم را داشته باشد. در لینوکس مجوزهای تعریف شده خاصی برای فایل یا فهرست ها وجود دارد که بسیاری از آن ها محدود و بسیاری دیگر قابلیت دسترسی بیشتری به کاربر می دهند. برای تغییر این نوع دسترسی ها در لینوکس از برنامه **chmod** (یا همان Change Mode) از طریق ترمینال استفاده می شود. قبل استفاده از دستور باید از نوع و مقدار دسترسی های فایل و فهرست ها آشنا شوید.

¹ Modification

² Grant

روشی برای ذخیره و سازمان‌دهی پرونده‌های رایانه‌ای و داده‌هایشان است تا یافتن و دسترسی به آن‌ها را آسان کند .

در رایانش، یک سیستم فایل برای کنترل نحوه ذخیره و بازیابی اطلاعات استفاده می‌شود. بدون یک سیستم فایل اطلاعات قرار گرفته در فضای ذخیره‌سازی، یک بدنه کلی و بزرگ از اطلاعات خواهد بود که نمی‌توان تشخیص داد یک تکه از اطلاعات در کجا ختم می‌شود و تکه بعدی از کجا شروع می‌شود .

با جداسازی اطلاعات به قطعه‌های منحصر به فرد و نام‌گذاری هر قطعه، اطلاعات به راحتی از هم جدا و تشخیص داده می‌شوند. هر قطعه از اطلاعات فایل نامیده می‌شود .

ساختار و قوانین منطقی که برای مدیریت دسته‌ای از اطلاعات و نام آن‌ها استفاده می‌شود «سیستم فایل» نامیده می‌شود . انواع مختلفی از سیستم فایل وجود دارد که هر یک دارای ساختار و منطق متفاوتی می‌باشند. هر یک دارای خواص مختلفی از سرعت، انعطاف‌پذیری، امنیت، اندازه و غیره هستند. بعضی از سیستم فایل‌ها برای کاربردهای خاصی طراحی شده‌اند. برای مثال سیستم فایل ایزو ۹۶۶۰ مخصوص دیسک‌های نوری طراحی شده است .

یک سیستم پرونده‌ای همه فراداده‌های تخصیص داده شده به پرونده را ذخیره می‌کند - شامل نام پرونده، طول محتوایات یک پرونده و مکان پرونده در سلسله مراتب پوشه - که از محتوایات پرونده متمایز می‌سازد . بسیاری از سیستم‌های پرونده‌ای نام‌های همه پرونده‌ها را در فهرست راهنما در یک مکان ذخیره می‌کند - جدول فهرست راهنما برای آن فهرست راهنما - که غالباً مانند بقیه پرونده‌ها ذخیره می‌شود. بسیاری از سیستم‌های پرونده‌ای فقط تعدادی از فراداده‌ها را برای یک پرونده در جدول فهرست راهنما قرار می‌دهند و باقی ماندهٔ فراداده‌ها برای آن پرونده در یک ساختار کاملاً مجزا قرار دارند، مانند گروه اطلاعاتی .

صفات اضافی می‌توانند به سیستم‌های پرونده‌ای تخصیص داده شوند، مانند ex3, NTFS, XFS, ex2, بعضی از نگارش‌های UFS، و HFS+ ، که از صفات پرونده‌های قابل تعمیم استفاده می‌کنند. بعضی از سیستم‌های پرونده‌ای برای صفات تعرف شده توسط کاربر تهیه شده‌اند، مانند نویسندگان، کدبندی نویسه‌های یک سند یا اندازه یک تصویر .

بعضی سیستم‌های پرونده‌ای اجازه می‌دهند که مجموعه‌های داده‌ای مختلف به یک نام پرونده تخصیص داده شوند. این مجموعه‌های جدا ممکن است به عنوان جریان‌هایی یا محل‌های انشعاب، ارجاع داده شوند. شرکت اپل مدت طولانی از سیستم انشعاب در مک‌ینتاش استفاده می‌کرد، و همچنین شرکت مایکروسافت نیز در NTFS جریان‌ها را پشتیبانی می‌کند. بعضی از سیستم‌های پرونده‌ای چندین تجدید نظر از یک پرونده را در یک نام پرونده حفظ می‌کنند. خود نام پرونده بیشتر نسخه اخیر را بازیابی می‌کند، این در حالی است که نسخه‌های قدیمی تر تنها با یک نام‌گذاری خاص قابل دسترسی هستند، همانند "filename;4" یا "filename(-4)" که برای دسترسی به چهار نسخه قبل نگه داشته شد .

سیستم‌های پرونده‌ای از نوع دیسک

یک سیستم پرونده‌ای از نوع دیسک از توانایی‌های رسانه‌های (media) ذخیره‌گاه دیسک (disk storage) استفاده می‌کند تا در مدت کوتاهی، به صورت تصادفی داده‌ها را آدرس‌دهی کند. از دیگر ملاحظات که انجام می‌شوند، می‌توان به سرعت دسترسی به داده‌ها اشاره کرد، که ابتدا باید درخواست شود و پیش‌بینی شود که داده مورد نظر نیز ممکن است درخواست شده باشد. این موضوع به چندین کاربر یا پردازنده اجازه می‌دهد، تا به

داده‌های متفاوتی روی دیسک، بدون توجه به توالی مکان قرارگیری داده، دسترسی داشته باشند. به عنوان مثال: جدول تخصیص پرونده (FAT) فت ۱۲ (FAT12)، فت ۱۶ (FAT16)، فت ۳۲ (FAT32)، اکس فت (exFAT)، NTFS، HFS، HFS+، HPFS، APFS، UFS، ext2، ext3، ext4، btrfs، XFS، ایزو ۹۶۰ (ISO 9960)، Files-11، سیستم پرونده‌ای وریتاس (Veritas File System)، VMFS، UDF. و Zfs، ReiserFS بعضی از سیستم‌های پرونده‌ای از نوع دیسک عبارتند از سیستم پرونده‌ای ژورنالینگ (journaling file system) با سیستم پرونده‌ای ورژنینگ (versioning file system).

سیستم پرونده‌ای از نوع دستگاهی

سیستم پرونده‌ای از نوع دستگاه، دستگاه‌های ورودی/خروجی (I/O devices) و دستگاه‌های سودو (pseudo-devices) را به عنوان پرونده‌هایی به نام پرونده‌های دستگاه، نشان می‌دهد. مثال‌های این سیستم پرونده‌ای در سیستم عامل‌های شبه یونیکس عبارت اند از devfs: و لینوکس ۲.۶، udev. و در سیستم عامل‌های دیگر عبارت اند از TOPS-10: و دیگر سیستم عامل‌های مشابه این، که نام کامل پرونده یا نام مسیر آن در پیشوند دستگاه می‌آید. دیگر دستگاه‌های شامل سیستم پرونده‌ای به کمک پیشوند یک دستگاه، که نمایانگر آن دستگاه است، بدون اینکه چیزی در انتهایش بیاید، ارجاع داده می‌شوند.

سیستم فایل	بیشترین طول اسم پرونده	کاراکتر مجاز در ورودی‌های فهرست	حداکثر طول مسیر	حداکثر حجم فایل	حداکثر اندازه ظرفیت
exFAT	۲۲۶ کارکتر	Any یونی‌کد except NUL	محدودیت تعریف نشده	۱۲۷PB (۱۲۷ × ۱۰۲۴ ^۵ بایت)	۶۴ (۶۴ZB × ۵۱۲) ترابایت recommended
TexFAT	۲۴۷ کارکتر	Any یونی‌کد except NUL	محدودیت تعریف نشده	۲ گیگابایت	۵۰۰ گیگابایت Tested
۱۲FAT	UTF-16 ۲۵۵ (۸۰۳ code units with LFN)	Any یونی‌کد except NUL (with LFN)	محدودیت تعریف نشده	۳۲ مگابایت	۱ مگابایت to 32 مگابایت
۱۶FAT	UTF-16 ۲۵۵ (۸۰۳ code units with LFN)	Any یونی‌کد except NUL (with LFN)	محدودیت تعریف نشده	۲ گیگابایت	۱۶ مگابایت to 2 گیگابایت
۳۲FAT	UTF-16 ۲۵۵ (۸۰۳ code units with LFN)	Any یونی‌کد except NUL (with LFN)	محدودیت تعریف نشده	۴ گیگابایت	۵۱۲ مگابایت to 8 ترابایت
Fossil	ناشناخته	ناشناخته	ناشناخته	ناشناخته	ناشناخته
MFS	۲۵۵ بایت	Any byte except :	No path (flat filesystem)	۲۲۶ مگابایت	۲۲۶ مگابایت
HFS	۳۱ بایت	Any byte except	نامحدود	۲ گیگابایت	۲ ترابایت
HPFS	۲۵۵ بایت	هر بایتی به جز NUL	محدودیت تعریف نشده	۲ گیگابایت	۲ ترابایت
NTFS	۲۵۵ کارکتر	Any یونی‌کد except NUL and \ / : * ? " < >	Unicode ۳۲۰۷۶۷ کارکتر with each path component (directory or filename) commonly up to 255 کارکتر	۱۶ (۱۶EB × ۱۰۲۴ ^۶ بایت)	۱۶EB
HFS Plus	UTF-16 code ۲۵۵ units	Any valid یونی‌کد	نامحدود	slightly less than 8 EB	slightly less than 8 EB
FFS	۲۵۵ بایت	هر بایتی به جز NUL	محدودیت تعریف نشده	ZB۸	ZB۸
۱UFS	۲۵۵ بایت	هر بایتی به جز NUL	محدودیت تعریف نشده	۴ گیگابایت to 226 ترابایت	۲۲۶ ترابایت
۲UFS	۲۵۵ بایت	هر بایتی به جز NUL	محدودیت تعریف نشده	۵۱۲ گیگابایت to 32 PB	۱ (۱YB × ۱۰۲۴ ^۸ بایت)
۲ext	۲۵۵ بایت	هر بایتی به جز NUL and /	محدودیت تعریف نشده	۱۶ گیگابایت to 2 ترابایت	۲ ترابایت to 32 ترابایت
۳ext	۲۵۵ بایت	هر بایتی به جز NUL and /	محدودیت تعریف نشده	۱۶ گیگابایت to 2 ترابایت	۲ ترابایت to 32 ترابایت
۴ext	۲۵۶ بایت	هر بایتی به جز NUL and /	محدودیت تعریف نشده	۱۶ گیگابایت to 16 ترابایت	۱EB
ReiserFS	۴۰۳۲ بایت/ ۲۲۶ کارکتر	هر بایتی به جز NUL	محدودیت تعریف نشده	۸ ترابایت (۷, ۳۰۶, ۲)	۱۶ ترابایت

	گیگابایت ۷) ۳۰۵				
EB۸	EB۸	محدودیت تعریف نشده	هر بایتی به جز NUL	۲۵۵ بایت	NILFS
ناشناخته	۸ ترابایت on ۸۶X	محدودیت تعریف نشده	Any byte except / and NUL	۳۰۹۷۶ بایت	۴Reiser
۵۱۲ ترابایت to 4 PB	EB۸	محدودیت تعریف نشده	هر بایتی به جز NUL	۲۵۵ بایت	۱JFS
PB۳۲	PB۴	محدودیت تعریف نشده	Any یونی کد except NUL	۲۵۵ بایت	JFS
۶۴ مگابایت	۶۴ مگابایت	محدودیت تعریف نشده	هر بایتی به جز NUL	۱۴ or 30 بایت، set at filesystem creation time	Minix V1 FS
۱ گیگابایت then 2 ترابایت	۴ گیگابایت	محدودیت تعریف نشده	هر بایتی به جز NUL	۱۴ or 30 بایت، set at filesystem creation time	Minix V2 FS
۱۶ ترابایت	۴ گیگابایت	محدودیت تعریف نشده	هر بایتی به جز NUL	۶۰ بایت	Minix V3 FS
۶۴ ترابایت	۴ ترابایت	۲۰۰۴۸	هر بایتی به جز NUL and /	۱۲۸	۲VMFS
۶۴ ترابایت	۲ ترابایت	۲۰۰۴۸	هر بایتی به جز NUL and /	۱۲۸	۳VMFS
۸ ترابایت	۴ گیگابایت & ۱ (Level ۸) to ۲ (Level 3)	۱۸۰~ بایت؟	Depends on Level	Level 1: 8.3, : ۳ & ۲Level ۱۸۰~	ISO ۹۶۶۰:۱۹۸۸
same as ISO ۹۶۶۰:۱۹۸۸	same as ISO ۹۶۶۰:۱۹۸۸	ناشناخته	All UCS-2 code except *, /, \, ., :, l۴۲ and ?l	۶۴ یونی کد کارکتر	Joliet ("CDFS")
EB۱۶	EB۱۶	ناشناخته	هر بایتی به جز NUL	۲۵۵ بایت	Btrfs

File system	لینوکس
QFS	via client software
NWFS	via ncpfs client software
Reiser۴	with a kernel patch
DECtape	with AncientFS
ZFS	with FUSE
LFS	with logfs and others
NSS	with Novell OES2
exFAT	with third party driver
FAT۱۲	آری
FAT۱۶	آری
FAT۳۲	آری
Apple HFS	آری
HPFS	آری
FFS	آری
ext۲	آری
ext۳	آری
Lustre	آری
GFS	آری
ReiserFS	آری
OCFS	آری
OCFS۲	آری
Reliance Nitro	آری
XFS	آری
JFS	آری
UDF	آری
VxFS	آری
IBM GPFS	آری
Btrfs	آری
LTFS	آری

NTFS	Kernel 2.2 or newer, or ان تی اف اس - with جی ^۳ or ntfsprogs
ext ^۴	آری since kernel ۲,۶,۲۸
NILFS	آری since kernel ۲,۶,۳۰
UFS ^۱	بخشی - read only
UFS ^۲	بخشی - read only
BFS	بخشی - read- only
Apple HFS Plus	بخشی - write support occurs if journal is empty, but requires a force mount.
ODS-۵	read-only بخشی with kernel module ^[۱۵۲]
ODS-۲	read-only بخشی with tool or kernel module
VMFS ^۳	read-only بخشی with vmfs
VMFS ^۲	ناشناخته
Level-D	ناشناخته
RT-۱۱	ناشناخته
Reliance	نه
Fossil	نه
IBM HFS	نه
IBM zFS	نه

ext	Rémy Card	1992	Linux
Xiafs	Q. Frank Xia	1993	Linux
ext3	Stephen Tweedie	1999	Linux
GFS	Sistina (Red Hat)	2000	Linux
ReiserFS	Namesys	2001	Linux
OCFS	Oracle Corporation	2002	Linux
SquashFS	Phillip Lougher, Robert Lougher	2002	Linux
Lustre	Cluster File Systems[5]	2002	Linux
Google File System	Google	2003	Linux
Reiser4	Namesys	2004	Linux
GlusterFS	Gluster Inc.	2005	Linux
OCFS2	Oracle Corporation	2005	Linux
NILFS	NTT	2005	Linux
GFS2	Red Hat	2006	Linux
ext4	various	2006	Linux
Btrfs	Oracle Corporation	2007	Linux
CASL	Nimble Storage	2010	Linux
OrangeFS	Omnibond and others	2011	Linux
F2FS	Samsung Electronics	2012	Linux
bcacheFS	Kent Overstreet	2015	Linux
NOVA	UC, San Diego	2017	Linux
LSFS	StarWind Software	2009	Linux, FreeBSD, Windows
ext2	Rémy Card	1993	Linux, Hurd
BeeGFS	Fraunhofer/ ThinkParQ	2005	Linux, Windows via Samba

نام سیستم فایل	دستور ترمینالی ساخت
Btrfs	mkfs.btrfs(8)
VFAT	mkfs.fat(8)
exFAT	mkexfatfs(8)
F2FS	mkfs.f2fs(8)
ext3	mke2fs(8)
ext4	mke2fs(8)
HFS	mkfs.hfsplus(8)
JFS	mkfs.jfs(8)
NILFS2	mkfs.nilfs2(8)
NTFS	mkfs.ntfs(8)
Reiser4	mkfs.reiser4(8)
ReiserFS	mkfs.reiserfs(8)
UDF	mkfs.udf(8)
XFS	mkfs.xfs(8)
ZFS	

دستور df

```
$ df -Th
```

OR

```
$ df -Th | grep "^/dev"
```

```
tecmin@TecMint ~ $ df -Th
Filesystem      Type      Size  Used Avail Use% Mounted on
udev            devtmpfs  3.9G   0    3.9G   0% /dev
tmpfs           tmpfs     788M  9.6M  779M   2% /run
/dev/sda10      ext4      324G  202G  106G  66% /
tmpfs           tmpfs     3.9G  118M  3.8G   3% /dev/shm
tmpfs           tmpfs     5.0M   4.0K  5.0M   1% /run/lock
tmpfs           tmpfs     3.9G   0    3.9G   0% /sys/fs/cgroup
cgmanagerfs    tmpfs     100K   0    100K   0% /run/cgmanager/fs
tmpfs           tmpfs     788M   36K  788M   1% /run/user/1000
tecmin@TecMint ~ $
tecmin@TecMint ~ $ df -Th | grep "^/dev"
/dev/sda10      ext4      324G  202G  106G  66% /
tecmin@TecMint ~ $
```

```
$ fsck -N /dev/sda3  
$ fsck -N /dev/sdb1
```

```
tecmint@TecMint ~ $ fsck -N /dev/sda3  
fsck from util-linux 2.27.1  
[/sbin/fsck.ext2 (1) -- /dev/sda3] fsck.ext2 /dev/sda3  
tecmint@TecMint ~ $  
tecmint@TecMint ~ $ fsck -N /dev/sdb1  
fsck from util-linux 2.27.1  
[/sbin/fsck.ext2 (1) -- /dev/sdb1] fsck.ext2 /dev/sdb1  
tecmint@TecMint ~ $
```

```
$ lsblk -f
```

```
tecmint@TecMint ~ $ lsblk -f
NAME        FSTYPE LABEL        UUID                               MOUNTPOINT
sda
├─sda1      ntfs      WINRE_DRV    D4A45AAAA45A8EBC
├─sda2      vfat      SYSTEM_DRV   185C-DA5B
├─sda3      vfat      LRS_ESP      0E60-2E0E
├─sda4
├─sda5      ntfs      Windows8_OS  18D0632AD0630CF6
├─sda6      ntfs      LENOVO       9286FFD986FFBC33
├─sda7      ntfs      PBR_DRV      ECD06683D066543C
├─sda8
├─sda9      swap
├─sda10     ext4      e040de62-c837-453e-88ee-bd9000387083 [SWAP]
sr0          bb29dda3-bdaa-4b39-86cf-4a6dc9634a1b /
```

```
tecmint@TecMint ~ $
```

```
$ mount | grep "^/dev"
```

```
tecmin@TecMint ~ $  
tecmin@TecMint ~ $ mount | grep "^/dev"  
/dev/sda10 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)  
tecmin@TecMint ~ $  
tecmin@TecMint ~ $
```

```
$ blkid /dev/sda3
```

```
tecmint@TecMint ~ $ blkid /dev/sda3
/dev/sda3: LABEL="LRS_ESP" UUID="0E60-2E0E" TYPE="vfat" PARTLABEL="Basic data partition" PARTUUID="d464feab-0791-4866-a36b-90dbe6d6a437"
tecmint@TecMint ~ $ blkid /dev/sda10
/dev/sda10: UUID="bb29dda3-bdaa-4b39-86cf-4a6dc9634a1b" TYPE="ext4" PARTUUID="26b60905-1c39-4fd4-bdce-95c517c781fa"
tecmint@TecMint ~ $
tecmint@TecMint ~ $
```

```
$ sudo file -sL /dev/sda3
```

```
tecmin@TecMint ~ $ sudo file -sL /dev/sda3
/dev/sda3: DOS/MBR boot sector, code offset 0x58+2, OEM-ID "MSDOS5.0", sectors/cluster 8,
reserved sectors 4206, Media descriptor 0xf8, sectors/track 63, heads 255, hidden sectors
2582528, sectors 2048000 (volumes > 32 MB) , FAT (32 bit), sectors/FAT 1993, serial number
0xe602e0e, unlabeled
tecmin@TecMint ~ $
tecmin@TecMint ~ $ sudo file -sL /dev/sda10
/dev/sda10: Linux rev 1.0 ext4 filesystem data, UUID=bb29dda3-bdaa-4b39-86cf-4a6dc9634a1b
(needs journal recovery) (extents) (large files) (huge files)
tecmin@TecMint ~ $
tecmin@TecMint ~ $
tecmin@TecMint ~ $
```



```
$ cat /etc/fstab
```

```
tecmin@TecMint ~ $ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda10 during installation
UUID=bb29dda3-bdaa-4b39-86cf-4a6dc9634a1b / ext4 errors=remount-ro 0
1
# swap was on /dev/sda9 during installation
UUID=e040de62-c837-453e-88ee-bd9000387083 none swap sw 0
0
```

تماس با ما

telegram: @linux
web site: www.linuxtnt.ir

لینوکس را فارسی یاد بگیرید
Linux Tips And Tricks
www.linuxtnt.ir

نشر آزاد

