آموزش فارسی لینوکس : قسمت ششم

جمعه, ۷ آذر ۱۳۹۳، ۰۷:۳۰ ب.ظ

سر فصل های قسمت ششم :

```
102.3 Manage shared libraries
102.1 Design hard disk layout
104.1 Create partitions and file systems
```

102.3 Manage shared libraries

Description : Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary .

Key Knowledge Areas :

Identify shared libraries.

Identify the typical locations of system libraries.

Load shared libraries.

The following is a partial list of the used files, terms and utilities :

ldd

ldconfig

/etc/ld.so.conf

LD_LIBRARY_PATH

مقدمه :

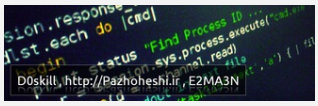
زمانی که برنامه را اجرا می کنید مثلا برنامه ی ls , این برنامه برای اجرا به یک سری کتابخانه نیازمند است و باید این کتابخانه ها در داخل ram قرار بگیرد . ابزاری این کار را برای برنامه های مختلف انجام می دهد به عنوان مثال در هنگام اجرای دستور ls ابتدا چک می کند که این دستور به چه کتابخانه هایی احتیاج دارد و از بین این کتابخانه ها کدام در داخل ram قرار ندارد , سپس آن ها را به داخل ram می برد تا در نتیجه دستور ls کار کند . این ابزار linker نام دارد .

نکته ای که باید به آن توجه کرد این است که در یک سیستم عامل تعداد زیادی کتابخانه وجود دارد و linker نمی تواند هر بار برای اجرای هر برنامه , تمامی مسیر های کتابخانه ها را به صورت آنلاین بگردد لذا از دیتابیسی استفاده می کند که در آن , آدرس دقیق هر کتابخانه نوشته شده است . پس زمانی که linker به کتابخانه ای احتیاج داشته باشد ابتدا در دیتابیس خود به دنبال نام آن گشته سپس مسیر آن را از داخل db می خواند و در آخر آن را در داخل ram قرار می دهد .

این دیتابیس را می توان در دو فایل زیر دید که اولی یک فایل متنی و دومی یک فایل باینری است :

```
/etc/ld.so.conf
/dtc/ld.so.cache
```

طبیعتا به صورت پیش فرض در داخل این دیتابیس مسیر و کتابخانه هایی مشخص در نظر گرفته شده است اما ممکن است به مرور زمان احتیاج باشد که تعدادی کتابخانه به این دیتابیس اضافه کنیم . برای این منظور ابتدا



طبقه بندی موضوعی

- مقاله آموزشی (۱۱)
- فیلم آموزشی (۹)
- برنامه اختصاصی (۵)
- ارسال بلاگ (۲)
- آموزش فارسی لینوکس (۶)
- آموزش فارسی سیسکو (۲)

خلاصه آمار

مجموع نمایش‌ها	۲۴۲۸۴
مجموع بازدیدکننده‌ها	۸۸۷۸
بازدیدکننده‌های امروز	۴۸
نمایش‌های دیروز	۱۴۷
مجموع مطالب	۳۰
مجموع نظرات	۶۱
عمر سایت	۲۳۴ روز
حاضرین در سایت	۲۴

کلمات کلیدی

آموزش فارسی لینوکس | متااسپلویت | آموزش فارسی lpic | آموزش فارسی lpic 1 | کتاب آموزش سیسکو | exploits | مدارک معتبر سیسکو | مدرک ccna | آموزش فارسی سیسکو | آموزش فارسی network+ | آموزش سیسکو | آموزش فارسی ccna | RDP | Port Forwarding | debian | کرک WPA | کرک هندشیک | آموزش lpic | pyrit | پسورد لیست | mini wireless ids | دستورات لینوکس | جعل dns | اسنیف VPN | جلوگیری از هک | کاربرد SGID | Heartbleed | امنیت | باگ | پرمیشن SGID

بایگانی

- آذر ۱۳۹۳ (۱)
- آبان ۱۳۹۳ (۴)
- مهر ۱۳۹۳ (۵)
- شهریور ۱۳۹۳ (۱)
- تیر ۱۳۹۳ (۷)
- خرداد ۱۳۹۳ (۱)
- اردیبهشت ۱۳۹۳ (۳)
- فروردین ۱۳۹۳ (۸)

مسیر کتابخانه های خود را در فایل متنی ld.so.conf اضافه می کنیم سپس با استفاده از دستوری دیگر به نام ldconfig , ایندکس کتابخانه های جدید (موجود در مسیری که اضافه کردیم) را به دیتابیس اضافه می کنیم .

```
/etc/ld.so.cache
# ldconfig
```

توجه داشته باشید که برای اضافه کردن کتابخانه به فایل باینری ls.so.cache نیاز به یوزر root است .

دستور ldd :

با استفاده از این دستور می توان یافت که فایل اجرایی یک برنامه (exeute) برای اجرا به چه کتابخانه (ها) ای وابسطه است . به عنوان مثال برای دیدن وابستگی برنامه ی ls به چه کتابخانه ای می توان از این دستور استفاده کرد .

توجه کنید که باید مسیر فایل اجرایی را به این دستور بدهیم . مثال :

```
$ which ls
/bin/ls
$ ldd /bin/ls
```

برای راحتی می توان به صورت زیر عمل کرد :

```
$ ldd `which ls`
```

در مثال بالا ابتدا دستور داخل `` اجرا شده سپس خروجی آن دقیقا در جلوی دستور ldd قرار می گیرد . دقیقا همین کار را برنامه ی xargs انجام می دهد با این تفاوت که xargs خروجی دستور قبل را وارد ورودی دستور بعد می کند اما این روش دقیقا خروجی را در جلوی دستور ldd قرار می دهد .

در قسمت های قبل ورودی پذیر نبودن دستور rm را با هم بررسی کردیم و دیدیم که با استفاده از xargs این مشکل چگونه حل میشد , حال با استفاده از همین روش می توان تمامی مسیر های ذخیره شده یک فایل متنی را پاک کرد . مثال :

```
$ rm `cat badfile`
```

اگر از چند برنامه ldd بگیرید خواهید دید که یک خروجی در همه ی آن ها یکی است . یعنی :

```
/lib64/ld-linux-x86-64.so.2
```

در واقع این همان آدرس linker می باشد !

102.1 Design hard disk layout

Description : Candidates should be able to design a disk partitioning scheme for a Linux system .

Key Knowledge Areas :

- Allocate filesystems and swap space to separate partitions or disks.
- Tailor the design to the intended use of the system.
- Ensure the /boot partition conforms to the hardware architecture requirements for booting.
- Knowledge of basic features of LVM.

The following is a partial list of the used files, terms and utilities :

- / (root) filesystem.
- /var filesystem.
- /home filesystem.
- swap space.
- mount points.
- partitions.

مقدمه :

پیوندها

- پلیس فضای تولید و تبادل اطلاعات نیروی انتظامی جمهوری اسلامی ایران
- وبلاگ تخصصی تیم امنیتی سرزمین امن
- آریا دیتا سنتر
- تیم امنیتی هکران کلاه سیاه ایران - آموزش امنیت و راه های مقابله با هک
- TBH | Forums | Turk Black Hat
- وب سایت شخصی مسلم حقیقیان
- اشتراک هک و امنیت

در این قسمت در مورد نحوه ی درست پارتیشن بندی صحبت خواهیم کرد و همانند قسمت های قبل از دستوری خاص استفاده نمی کنیم . سوالی که در ابتدا مطرح می شود این است که چرا باید پارتیشن بندی کنیم ؟ این سوال شاید کمی بی منطق باشد چرا که معمولا هر کسی هارد دیسک های خود را بسته به نیاز خود پارتیشن بندی می کند . به عنوان مثال یکی از دلایل پارتیشن بندی این است که نمی خواهیم تمامی فضای هارد را به درایو C اختصاص دهیم چرا که با از بین رفتن ویندوز , دیگر اطلاعات نیز آسیب می بینند و یا این که بسته به کارمون نیاز به فایل سیستم های مختلف داریم برای مثال در یک هارد می خواهیم هم از فایل سیستم NTFS و هم از فایل سیستم Fat و هم از فایل سیستم ext2 استفاده کنیم و ...

پس می توان نتیجه گرفت که پارتیشن بندی یک هارد لازم است اما حالت های زیادی برای پارتیشن بندی یک هارد خام (نو) وجود دارد که هر کسی متناسب با کار و کیس خود آن را پارتیشن بندی می کند برای مثال فرض کنید که یک لپ تاپ نو بدون سیستم عامل خریدید و می خواهید که یکی از توزیع های گنو لینوکسی را بر روی آن نصب کنید . برای مثال اوبونتو . برای این منظور می توان 10 الی 20 گیگ به فایل سیستم اصلی و ما بقی را به دایرکتوری home اختصاص داد .

یا مثلا تازه وارد هستید و دوست دارید در کنار ویندوز اوبونتو نیز داشته باشید . برای این منظور 30 الی 40 گیگ مخصوص ویندوز و 10 الی 20 گیگ مخصوص اوبونتو و ما بقی را یک پارتیشن مجزا با فایل سیستم NTFS درست می کنید تا هم در لینوکس و هم در ویندوز بتوانید به آن دسترسی داشته باشید .

اما زمانی را فرض کنید که از محیط گرافیکی و دایرکتوری home خبری نیست مثلا یک سرور . در این صورت ما بقی هارد را به دایرکتوری var اختصاص می دهیم چرا که در قسمت های قبل دیدیم که این دایرکتوری حجمی متغییر دارد یعنی به مرور زمان حجم آن افزایش پیدا می کند .

پس پارتیشن بندی در لینوکس سلیقه ای است و هر کسی متناسب با کار و کیس خود باید پارتیشن بندی کند .

نکته : در مثال های بالا فقط دو دایرکتوری home و var برای اختصاص فضا نام برده شد در صورتی که می توان برای هر دایرکتوری پارتیشنی جدا در نظر گرفت اما معمولا این دو دایرکتوری بیش تر استفاده می شوند .

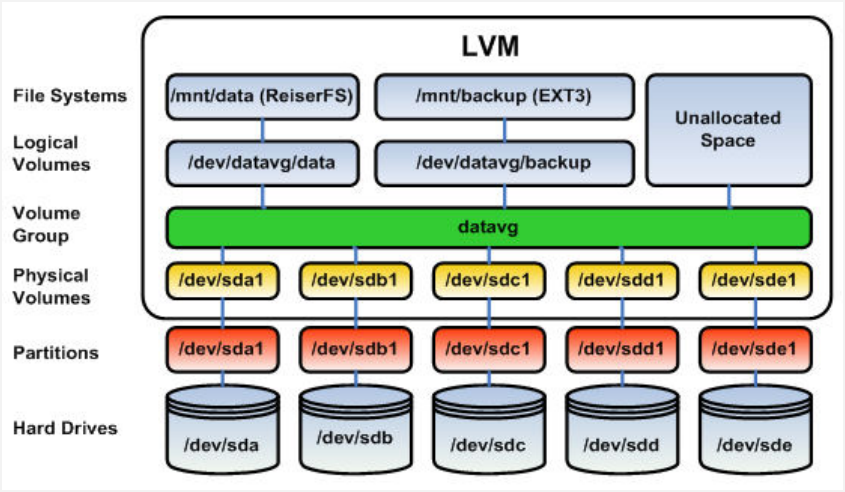
نکته : در مواقعی که ram دیگر جوابگوی حجم اطلاعاتی نیست , سیستم عامل از یک فضای مشخص از هارد دیسک به عنوان ram استفاده می کند (virtual memory) . این فضا swap نام دارد و معمولا مقدار آن را 2 برابر مقدار ram در نظر می گیرند . دوباره همان متفاوت بودن کار ها و کیس ها در این جا نیز خود را نشان می دهد برای مثال یک سرور ممکن است از تمام ظرفیت ram و swap خود استفاده کند اما ممکن است یک لپ تاپ هیچ فقط از تمامی ظرفیت ram خود استفاده نکند یعنی احتیاجی به استفاده از swap نباشد .

یاد آوری : به متصل کردن یک پارتیشن به یک دایرکتوری (برای استفاده از اطلاعات درون پارتیشن) Mount کردن و به آن دایرکتوری mount point می گویند .

LVM یا Logical Volume Manager :

زمانی رو فرض کنید که چندین هارد دیسک داریم برای مثال 6 تا هارد 1 TB . در حالت عادی می توان هر یک از این هارد ها (max) به 4 پارتیشن primary و یا 3 پارتیشن primary و یک پارتیشن extended تقسیم کرد . اما فرض کنید که تعداد پارتیشن ها برای ما مهم نیست اما حجم پارتیشن ها برای ما بسیار مهم هست برای مثال می خواهیم 3 پارتیشن با حجم های 3 و 2 و 1 TB داشته باشیم .

در این صورت از LVM استفاده می کنیم به این صورت که LVM تمامی این 6 هارد را یک هارد می بیند سپس ما این 6 هارد را با حجم های دلخواه پارتیشن بندی می کنیم . مثلا به 3 پارتیشن 3 و 2 و 1 TB تقسیم می کنیم .



104.1 Create partitions and files ystems

Description : Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This includes the handling of swap partitions .

Key Knowledge Areas :

Use various mkfs commands to set up partitions and create various filesystems such as:

ext2/ext3/ext4

xfs

reiserfs v3

vfat

The following is a partial list of the used files, terms and utilities :

fdisk

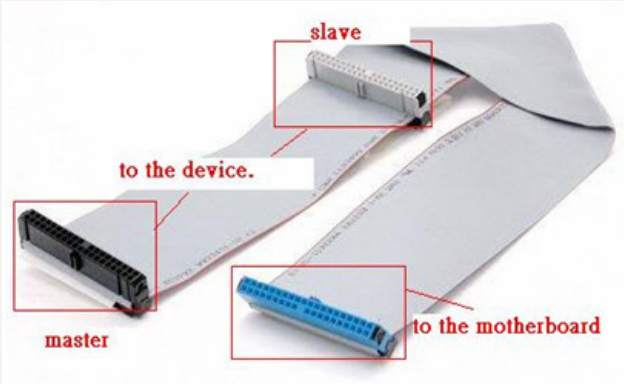
mkfs

mkswap

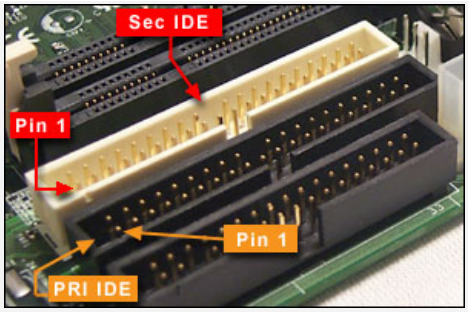
مقدمه :

همان طور که در بالا گفته شد یک هارد دیسک را می توان به 4 پارتیشن Primary و یا 3 پارتیشن Primary و یک پارتیشن Extended تقسیم کرد . هم چنین یک پارتیشن Extended را می توان به 15 پارتیشن دیگر تقسیم کرد .

هارد دیسک های IDE که بیش تر در سیستم ها و هارد دیسک های قدیمی به چشم می خورد از یک کابل ریبونی برای ارتباط با یکدیکر استفاده می کنند . این کابل 3 کانکتور دارد . کانکتور اول به MB متصل شده و کانکتور دوم و سوم Master و Slave نام دارد که کانکتور Master به درایو اولیه و کانکتور Slave به درایو ثانویه متصل می شود .



اکثر MB ها دارای دو Inter Face اولیه و ثانیه هستند یعنی در مجموع 4 دیوایس را می توان از طریق IDE به MB متصل کرد . به اینترفیس اصلی و یا اولیه primary و به اینترفیس ثانویه secondary می گویند .



پس به صورت خلاصه می توان گفت :

1. Primary Master
2. Primary Slave
3. Secondary Master
4. Secondary Salve

اما نسل جدید هارد دیسک ها و MB ها از اتصال Sata پشتیبانی می کنند . بر خلاف کابل های IDE دیگر از master و slave خبری نیست و از هر کابل برای اتصال یک دیوایس به MB استفاده می شود . در واقع ظرفیت پشتیبانی MB است که تعداد دیوایس هایی که می توانند از طریق Sata متصل شوند , تعیین می کند . برای مثال یک MB ممکن است 6 پورت sata داشته باشد .

نوع دیگری از اتصال scsi نام دارد و زمانی استفاده می شود که (برای مثال) جایی خالی برای اتصال از طریق IDE نداشته باشیم . پس یکی از دیوایس هایی که می توان از طریق scsi به MB متصل کرد هارد دیسک است .

اما در لینوکس هر یک از این نوع اتصال ها علایم مشخصی دارند . ابتدا اجازه بدید هارد هایی که از طریق IDE متصل شده اند را بررسی کنیم .

```
1. /dev/hda
2. /dev/hdb
3. /dev/hdc
4. /dev/hdd
```

ابتدا یاد آوردی می کنم که مسیر تمامی دیوایس ها در آدرس dev/ قرار دارد . همان طور که در بالا مشاهده می کنید عبارت hd در هر 4 مورد تکراری و مشخصه ی اتصال از طریق IDE است . پس :

1. اتصال از طریق IDE و Primary Master
2. اتصال از طریق IDE و Primary Slave
3. اتصال از طریق IDE و Secondary Master
4. اتصال از طریق IDE و Secondary Slave

پس دیدیم که hd مشخصه ی اتصال از طریق IDE است . هم چنین sata و scsi با مشخه ی sd معیین می شوند .

به عنوان مثال :

```
1. /dev/sda
2. /dev/sdb
3. /dev/sdc
```

که sda و sdb و sdc یک دیوایس می باشند .
نکته : فلش های مموری نیز به این صورت مشخص می شوند .

پس انواع اتصال و مشخصه ی هر کدام در لینوکس را بررسی کردیم . اما برای مشخص کردن پارتیشن های یک هارد به چه صورت باید عمل کنیم ؟

در جواب این سوال باید گفت که پارتیشن با اعداد از هم جدا و مشخص می شوند . به صورت زیر :

1. پارتیشن های Primary و Extended یک عدد از بین 1 تا 4 دریافت می کنند .
2. پارتیشن های Logical که همان پارتیشن های زیر مجموعه ی Extended هستند . از عدد 5 شروع می شوند .

نکته : ممکن است یک هارد , یک پارتیشن Primary و 2 پارتیشن Logical داشته باشد . در آن صورت (به عنوان مثال) ترتیب نام گذاری این پارتیشن ها به این صورت می شود :

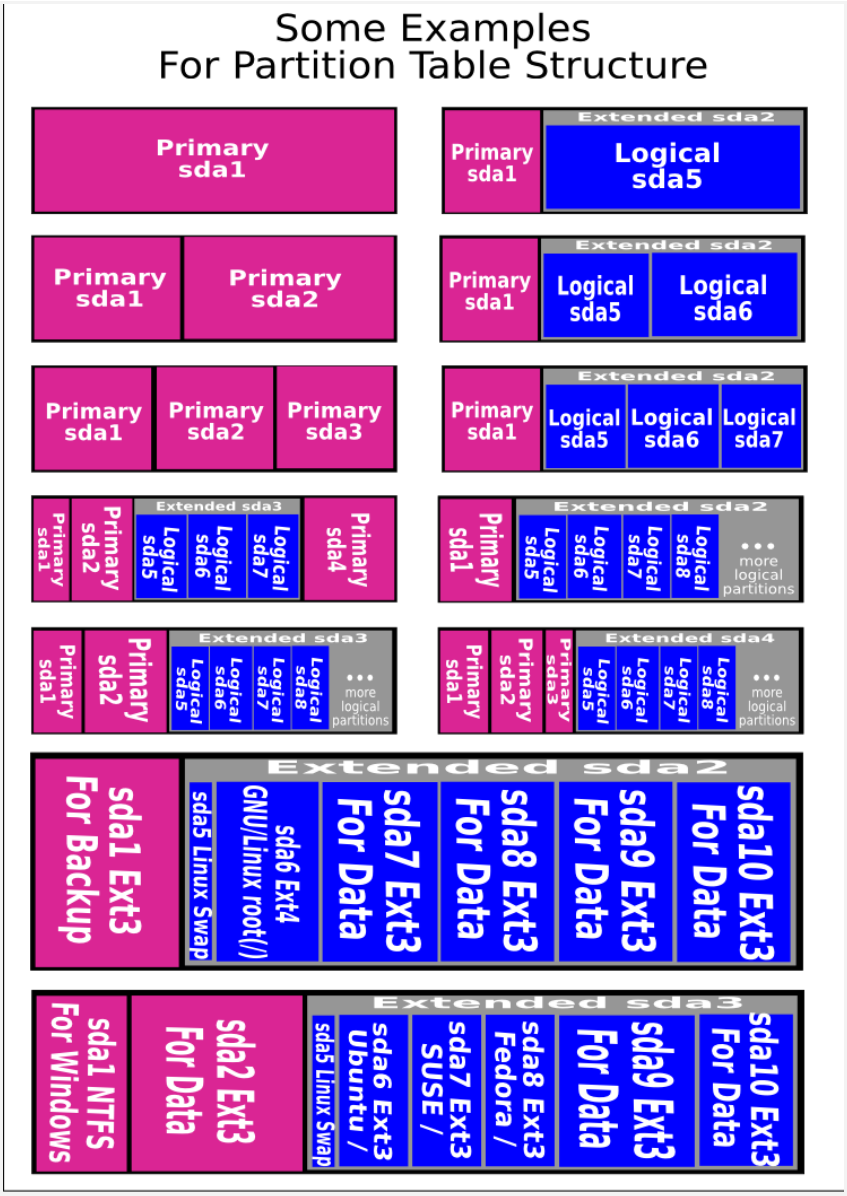
```
/dev/sda1
/dev/sda2
/dev/sda5
/dev/sda6
```

با توجه به مطالب گفته شده در بالا , sda1 و sda2 پارتیشن ها ی Primary و Extended و sda5 و sda6 پارتیشن های Logical می باشند .

نکته : دلیل خالی بودن sda3 و sda4 این است که در این هارد فقط دو پارتیشن Primary و Extended داریم .
نکته : اگر به غیر از فرض های بالا دو پارتیشن Primary دیگر نیز داشتیم , می توانستیم sda3 و sda4 را مشاهده کنیم .

برای مشاهده ی حالات احتمالی در پارتیشن بندی عکس زیر را دریافت کنید :

```
http://bayanbox.ir/id/5736200999538201163?view
```



پس می توان گفت تا به این جا یک درک نسبی ای از پارتیشن ها و هارد دیسک ها بدست آوردیم . حال به سراغ برنامه ها و دستورات لینوکسی برای پارتیشن بندی می رویم و آن ها را در قالب یک مثال در ماشین مجازی بررسی می کنیم .

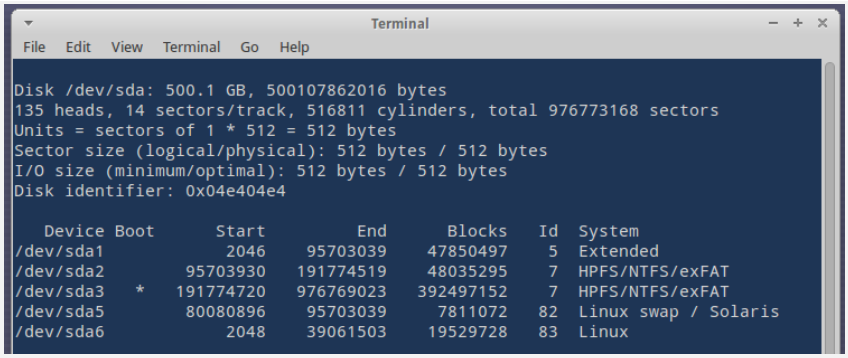
دستور fdisk :

اگر از این برنامه which بگیرید خواهید دید که فایل اجرایی این برنامه در مسیر sbin/ قرار دارد و اگر قسمت های قبل را مطالعه کرده باشید , می دونید که برنامه ی هایی که در این مسیر قرار دارند برای گروه مدیران سیستم می باشد . لذا برای اجرا این برنامه داشتن پرمیشن الزامی است .

همان طور که از اسم این برنامه پیداست برای پارتیشن بندی هارد دیسک از این ابزار استفاده می کنیم .

مثال : (از سویچ l- برای مشاهده ی پارتیشن ها همراه با جزییات استفاده می کنیم)

```
# fdisk -l /dev/sda
```



تجربه نشون داده که خیلی از تازه واردان با این مبحث مشکل دارند طوری که حتی اطلاعات خودشون رو هم فرمت می کنند . بنابر این یک فیلم 15 دقیقه ای آماده کردم تا به طور مفصل این قسمت رو با هم بررسی کنیم . داخل فیلم از ماشین مجازی استفاده شده است به این صورت که یک هارد دیسک sata را به صورت مجازی به اوبونتو خودمون اضافه می کنیم , سپس به پارتیشن بندی آن می پردازیم .

سازنده : E2MA3N
منبع : پژوهشی دات آی آر
حجم : 5 MB
مدت : 15 دقیقه

لینک دانلود

```
http://bayanbox.ir/info/3940369783529870700/Create-Partitions-and-Filesystem
```

دستور mkfs :

بعد از آن که پارتیشن بندی را انجام دادیم نوبت به مشخص کردن فایل سیستم آن پارتیشن است . همان طور که در ویندوز ابتدا یک پارتیشن را با حجمی دلخواه ایجاد می کنیم سپس آن را به NTFS و یا ... فرمت می کنیم . در این جا نیز همین روند را دنبال می کنیم یعنی در این مرحله باید فایل سیستم پارتیشن خودمون رو مشخص کنیم . برای این منظور می توانیم از دستور و برنامه ی mkfs استفاده کنیم .

به عنوان مثال یک پارتیشن ایجاد کرده ایم و می خواهیم فایل سیستم ext3 را برای آن در نظر بگیریم :

```
mkfs -t ext3 /dev/sdb1
```

برای مشاهده ی تغییرات به این صورت عمل کنید :

```
file -Ls /dev/sdb1
```

```
root@ubuntu:/home/elab# file -Ls /dev/sdb1
/dev/sdb1: Linux rev 1.0 ext3 filesystem data, UUID=7e6a3ba0-4f84-479e-921a-c60951d97320 (large files)
root@ubuntu:/home/elab#
```

نکته : با استفاده از سوییچ t- نوع فایل سیستم خودمون رو انتخاب کردیم که در مثال بالا ext3 را در نظر گرفتیم . توجه داشته باشید که می توانستیم از ext2 و یا ext4 و یا حتی NTFS و ... نیز استفاده کنیم .

دستور mkswap :

از این دستور برای مشخص کردن فایل سیستم swap استفاده خواهیم کرد . کار با این دستور آسان است اما تبدیل و اضافه کردن یک پارتیشن به عنوان swap به سیستم دارای نکاتی است که در جلوتر با هم بررسیش خواهیم کرد .

ابتدا فرض می کنیم که یک پارتیشن با حجم 512 MB جهت swap آماده کرده اید . به عنوان مثال من پارتیشن sdb1 که حجمی برابر با 512 MB دارد را برای این منظور آماده کرده ام .

```
root@ubuntu:/home/elab# fdisk /dev/sdb -l

Disk /dev/sdb: 2147 MB, 2147483648 bytes
16 heads, 16 sectors/track, 16384 cylinders, total 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x929c0d6c

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     1050623      524288   82  Linux swap / Solaris
root@ubuntu:/home/elab#
```

توجه داشته باشید که اگر با استفاده از fdisk پارتیشن خودتون رو آماده می کنید , بعد از ساخت پارتیشن با استفاده از سویچ t نوع فایل سیستم را از لینوکس (که به صورت دیفالت است) به swap تغییر دهید . برای تغییر همان طور که گفته شد از سویچ t استفاده کنید که در این صورت از شما عددی را برای تغییر نوع فایل سیستم می پرسد که برای تغییر آن به swap باید عدد 82 را وارد کنید (به سویچ l و جدول آن مراجعه شود)

حال با استفاده از دستور mkswap فایل سیستم پارتیشن خودمون رو به swap تغییر می دیم . به عنوان مثال :

```
mkswap /dev/sdb1
```

در مرحله ی بعد اگر همانند قسمت قبلی از دستور file استفاده کنید , خواهید دید که فایل سیستم پارتیشن ما به swap تغییر کرده است . یعنی :

```
file -Ls /dev/sdb1
```

```
root@ubuntu:/home/elab# file -Ls /dev/sdb1
/dev/sdb1: Linux/i386 swap file (new style), version 1 (4K pages), size 131071 p
ages, no label, UUID=6290073b-9a5f-4f47-97eb-8e89943d0964
root@ubuntu:/home/elab#
```

حال برای استفاده از این پارتیشن به عنوان فضای swap , باید آن را به سیستم معرفی کنیم . برای این منظور از دستور swapon به صورت زیر استفاده می کنیم :

```
swapon /dev/sdb1
```

برای مشاهده ی تغییرات می توانید از برنامه ی htop و یا free و یا top و ... استفاده کنید و خواهید دید (به عنوان مثال در این جا) که فضای swap به اندازه ی 512 MB افزایش پیدا کرده است .

برای مشاهده ی پارتیشن هایی که به عنوان swap در نظر گرفته شده اند (و فعال اند) می توان از دستور swapon همراه با سویچ -s استفاده کرد و یا :

```
cat /proc/swaps
```

نکته : بعد از هر بار ریست سیستم , این تغییرات از بین می رود و فضای swap به همان مقدار قبلی خود باز می گردد . برای جلوگیری از این تغییرات به صورت زیر عمل می کنیم :

```
echo "/dev/sdb1 swap swap defaults 0 0" >> /etc/fstab
```

منبع : پژوهشی دات آی آر | Pazhoheshi.ir

نویسنده : E2MA3N

دانلود PDF این مقاله

هر گونه نظر و انتقاد رو در قالب نظر با من در میون بزارید .

E2MA3N ۹۳/۰۹/۰۷ ۰ ۰

LVM چیست ؟ | آموزش دستور fdisk | آموزش فارسی lpic | آموزش فارسی lpic 1 | آموزش فارسی لینوکس

دستور ldconfig | پارتیشن Extended | پارتیشن Logical | پارتیشن Primary | پارتیشن بندی در لینوکس

نظرات (۰)

هیچ نظری هنوز ثبت نشده است

ارسال نظر آزاد است، اما اگر قبلا در بیان ثبت نام کرده اید می توانید ابتدا وارد شوید.

نام *

پست الکترونیک

سایت یا وبلاگ

پیام *

کد امنیتی *

نظر بصورت خصوصی ارسال شود

پست الکترونیک برای عموم قابل مشاهده باشد

ارسال

بلاگ بیان، رسانه متخصصان و اهل قلم