

# کوکی لینوکس به زبان

فارسی (ویرایش دو)

## Linux Cookie in Persian v.2

نویسنده : حسام الدین توحید

SKYWAN13@YAHOO.COM



## شناسنامه کتاب

سرشناسه: کوکی لینوکس به زبان فارسی – ویرایش دو

عنوان و نام پدید آور: کوکی لینوکس به زبان فارسی ; حسام الدین توحید

مشخصات نشر: انتشار به صورت اینترنتی طبق لیسانس GPL v3 – ایران 1394

مشخصات ظاهری: 223 ص.: جدول، نمودار، متن.

وضعیت فهرست نویسی: فیا

یادداشت: عنوان اصلی Linux Cookie in Persian v.2

موضوع: سیستم عامل لینوکس ----- سیستم های عامل ( کامپیوتر )

شناسه افزوده: توحید، حسام الدین، 1359 -، مترجم و نویسنده

ناشر: اینترنتی

قیمت: رایگان

سال انتشار: 1394

شماره نسخه ویرایش: **دو**

هر گونه کپی برداری از این کتاب با حفظ حقوق مولف مجاز می باشد، در نشر آن بکوشید.

## مقدمه مولف :

آنچه پیش رو دارید به صورت رایگان و تحت لیسانس GPL v3 به علاقه مندان لینوکس هدیه می گردد . در تهیه این کتاب از سر فصل های درسی گفته شده در دوره های LPic2 و RHCE و RHCSA استفاده شده و لازم می دانم از **مهندس مهدوی فر** به خاطر راهنمایی های مفیدشان و **مرکز آموزشهای پیشرفته دانشگاه شریف** تشکر کافی را داشته باشم. کتابی که پیش روی شماست سعی دارد با بکارگیری قابلیت های مختلف رابط خط فرمان، بدور از پیچیدگی های غیر ضروری و با نگاهی کاربردی، مروری داشته باشد بر راه اندازی و استفاده از چند سرویس مهم و پرکاربرد لینوکس که امید است مطالب ارائه شده بتواند باعث ارتقاء دانش فنی کاربران لینوکس و متخصصین IT شود. خواهشمند است هرگونه نقص در محتوا را به ایمیل نگارنده ارسال فرمائید. هر گونه استفاده نامناسب از محتوای ارائه شده بر عهده کاربر بوده و تمام حقوق این اثر به نویسنده آن تعلق دارد لذا با حفظ حقوق مولف در نشر آن بکوشید.

موفق باشید

**حسام الدین توحید**

فروردین 1394





## فهرست مطالب

### فصل اول – فرآیند کامل Boot شدن لینوکس

- 11 شرحی بر فرآیند کامل Boot شدن لینوکس
- 12 مرحله اول Boot لینوکس – POST
- 14 انتخاب اولین دستگاه قابل Boot
- 15 مرحله دوم Boot لینوکس – MBR
- 16 مرحله سوم Boot لینوکس – GRUB
- 17 مرحله چهارم Boot لینوکس – Kernel
- 19 مرحله پنجم Boot لینوکس – init process
- 21 مرحله ششم Boot لینوکس – Run Leves
- 23 روند ایجاد یک ترمینال

### فصل دوم – معرفی GRUB در لینوکس

- 26 مقدمه ای بر Bootloader
- 27 GRUB چیست؟
- 29 مراحل کار GRUB در هنگام Load کردن سیستم عامل
- 29 مرحله اول Boot Loader
- 30 مرحله یک و نیم Boot Loader
- 30 مرحله دوم Boot Loader
- 30 Load کردن سیستم عامل یا Kernel
- 31 LILO چیست؟
- 33 تعمیر GRUB صدمه دیده
- 35 ویرایش فایل grub در توزیع های مبتنی بر Debian
- 36 تغییر سیستم عامل پیش فرض در Grub
- 37 تغییر مدت زمان نمایش Grub در هنگام بوت
- 37 افزایش زمان نمایش منوی Grub
- 38 پاک کردن Grub و نصب Bootloader ویندوز

- 38 ▪ نرم افزار کمکی Super Grub Disk

## فصل سوم – راه اندازی FTP در لینوکس

- 41 ▪ آشنایی با پروتکل FTP
- 45 ▪ مدهای کاری سرویس دهنده FTP
- 47 ▪ نصب و راه اندازی VSFTP
- 48 ▪ مهمترین مسیرهای ایجاد شده توسط VSFTP
- 50 ▪ مروری بر تنظیمات فایل کانفیگ اصلی VSFTP
- 56 ▪ تغییر مسیر یوزرهای Local بعد از Login به FTP
- 57 ▪ محدود کردن دسترسی یوزرها به FTP
- 57 ▪ Jail کردن یوزرها در FTP
- 58 ▪ بر طرف کردن Error 500
- 59 ▪ ایجاد Mullti Homing در FTP
- 60 ▪ استفاده از دستور ftp به عنوان نرم افزار کلاینتی
- 63 ▪ فرق بین مد باینری و اسکی
- 63 ▪ SCP یک جایگزین امن برای FTP
- 64 ▪ متداول ترین کدهای وضعیت در FTP

## فصل چهارم – راه اندازی SSH در لینوکس

- 68 ▪ آشنایی با پروتکل SSH
- 70 ▪ نصب و راه اندازی سرویس OpenSSH
- 72 ▪ بررسی فایل های موجود در /etc/ssh/
- 73 ▪ پیکربندی سرویس SSH
- 77 ▪ استفاده از ssh جهت اتصال به کامپیوتر در شبکه
- 78 ▪ مبحث SSH Client
- 79 ▪ نصب و راه اندازی SSH Client
- 80 ▪ استفاده از ارتباط بدون پسورد در SSH Client
- 81 ▪ پیکربندی SSH Client و تولید کلید

## فصل پنجم – مدیریت Log و راه اندازی Logrotate

- 86 ▪ پیکربندی و مدیریت لاگها با syslog
- 88 ▪ نصب و راه اندازی سرویس syslog
- 88 ▪ مبنای کاری syslog
- 91 ▪ تنظیم لاگ بر اساس Unix domain socket
- 92 ▪ تنظیم لاگ بر اساس Internet socket
- 93 ▪ log فایل های مهم
- 94 ▪ ابزارهای گزارش گیری (logging)
- 95 ▪ چرخش لاگها با logrotate
- 95 ▪ نصب و راه اندازی سرویس logrotate
- 96 ▪ مهمترین فایل های logrotate
- 98 ▪ بررسی /etc/logrotate.d

## فصل ششم – اشتراک سیستم فایل با NFS

- 102 ▪ مقدمه ای بر NFS
- 103 ▪ مزایای استفاده از NFS
- 104 ▪ ویژگی های NFS.v4
- 105 ▪ RPC های مورد استفاده در NFS
- 106 ▪ نصب و راه اندازی NFS
- 107 ▪ Export دایرکتوری در NFS
- 110 ▪ Export دایرکتوری به سبک NFSv4
- 110 ▪ Export دایرکتوری Home
- 111 ▪ اختصاص پورت های ثابت به NFS
- 112 ▪ دستورات مهم برای NFS
- 113 ▪ استفاده از دایرکتوری Export شده در کلاینتها
- 114 ▪ مبحث Autofs

## فصل هفتم – زمان بندی فرایندها توسط Cron

- 118 ▪ زمان بندی اجرای برنامه ها توسط cron

- 118 ▪ نصب راه اندازی سرویس cron
- 119 ▪ مسیرها و فایل های اضافه شده به سیستم
- 121 ▪ توضیح فایل پیکربندی سرویس cron
- 123 ▪ استفاده از سرویس cron
- 125 ▪ محدود کردن دسترسی یوزرها برای استفاده از cron
- 126 ▪ مثال هایی برای cron
- 131 ▪ اجرای برنامه ها با واسط گرافیکی کاربر
- 131 ▪ زمان بندی اجرای فرامین توسط Anacron
- 132 ▪ تنظیم کردن وظایف Anacron
- 135 ▪ زمان بندی دستورات با at
- 136 ▪ جزئیات at

## فصل هشتم – بررسی LVM در لینوکس

- 140 ▪ LVM چیست
- 142 ▪ مزایای LVM
- 142 ▪ ساختمان LVM
- 143 ▪ شروع کار با LVM
- 145 ▪ راه اندازی LVM به صورت کامندی
- 163 ▪ ایجاد LVM Partition در زمان نصب لینوکس

## فصل نهم – سهمیه بندی دیسک در لینوکس

- 175 ▪ سهمیه بندی دیسک توسط Disk Quota
- 175 ▪ علت استفاده از Disk Quota
- 176 ▪ انواع سهمیه بندی در Disk Quota
- 176 ▪ سطوح هشدار در Disk Quota
- 177 ▪ نصب و راه اندازی Disk Quota
- 179 ▪ اجرا و اعمال Disk Quota

## فصل دهم – راه اندازی سرویس DHCP در لینوکس

- 185 ▪ سرویس Dhcp در لینوکس

- 186 ▪ شروط دریافت IP توسط کلاینت لینوکس
- 187 ▪ Dhcp Failover
- 187 ▪ مراحل Dora
- 189 ▪ مکانیزم Lease Duration
- 190 ▪ مکانیزم Duplicate Address Detection
- 190 ▪ نصب و راه اندازی سرویس Dhcp
- 192 ▪ پیکربندی و تنظیمات سرویس Dhcp
- 198 ▪ شروع به کار سرویس Dhcp
- 198 ▪ آمار IP های واگذار شده
- 199 ▪ مفهوم Dhcp relay agent
- 200 ▪ تنظیمات سرور و کلاینت
- 202 ▪ ساخت یک فایل نمونه

## فصل یازدهم – بررسی Xinetd

- 205 ▪ xinetd در لینوکس
- 206 ▪ مزایا و معایب xinetd
- 207 ▪ بررسی آپشن های موجود در یک فایل xinetd
- 209 ▪ نمونه ای از فایل ایجاد شده در xinetd برای سرویس vsftpd

## ضمیمه یک

- 211 ▪ اضافه کردن Repository به لینوکس

## ضمیمه دو

- 217 ▪ بازیابی پسورد یوزر root در لینوکس



**Linux  
Professional  
Institute**



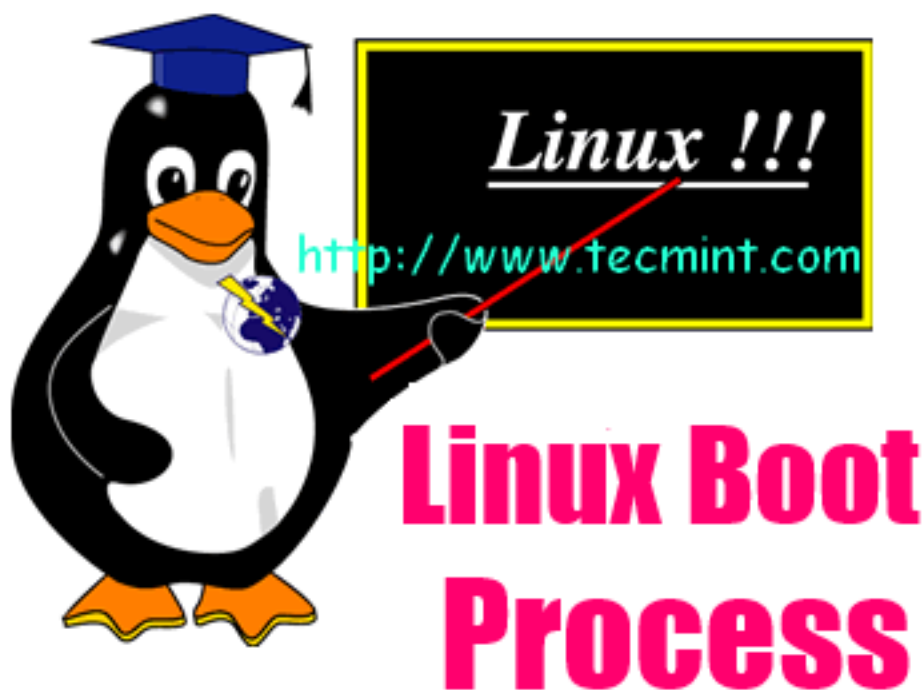
**RHCE**



**Red Hat Certified Engineer**

## فصل اول

### شرحی بر فرآیند کامل Boot شدن لینوکس

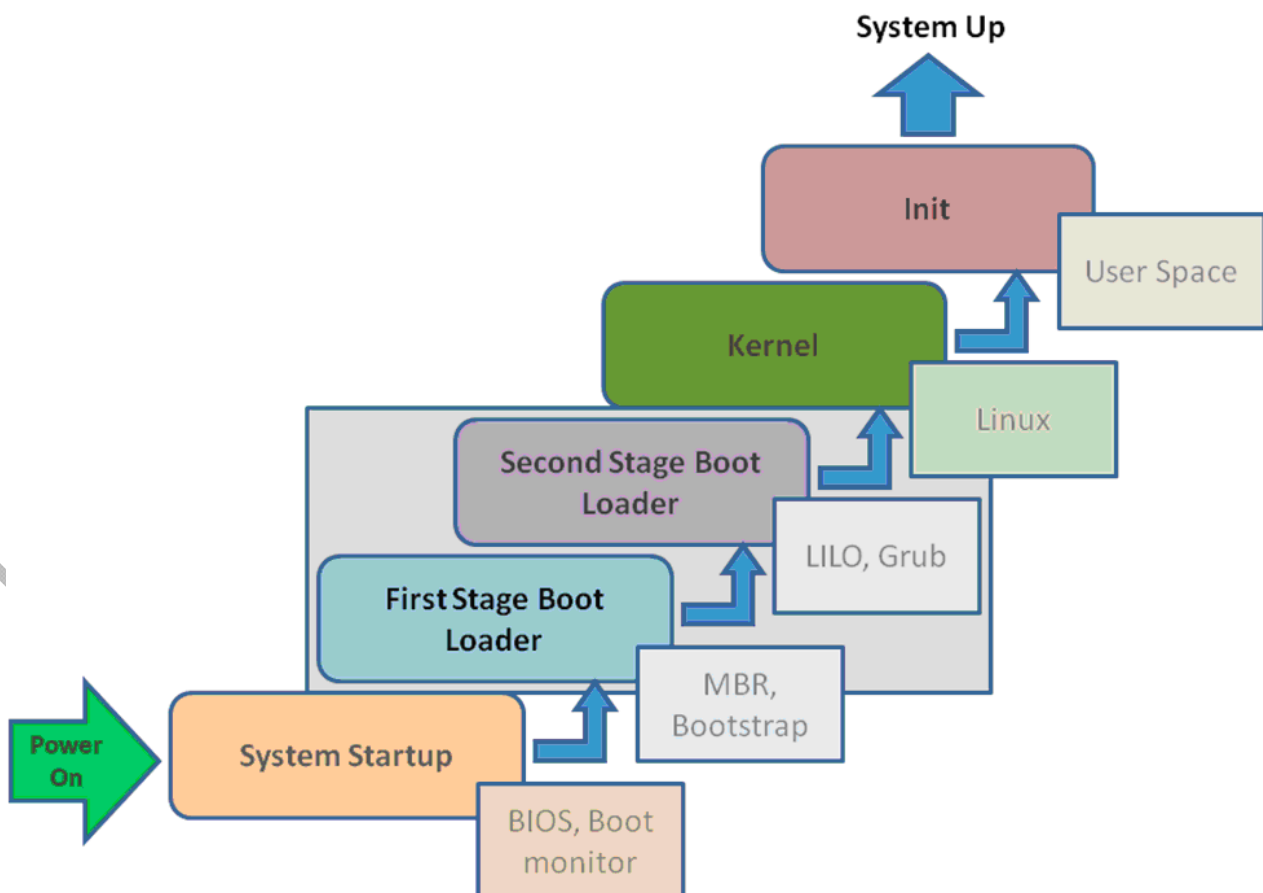


در تهیه فصل اول این کتاب، از مطالب منتشر شده در سایت محترم [itpro.ir](http://itpro.ir) استفاده شده که لازم می دانم از مهندس محمد نصیری مدیر انجمن این سایت به خاطر اجازه استفاده، تشکر لازم را داشته باشم.

## شرحی بر فرآیند کامل Boot شدن لینوکس

در این فصل به این موضوع می پردازیم که سیستم عامل لینوکس چگونه Boot می شود و فرآیند Startup این سیستم عامل از بدو روشن کردن سیستم تا استفاده از خط فرمان داخلی آن چگونه انجام می شود. ما هر روز بارها کامپیوترمان را روشن کرده و برای چند ثانیه تا چند دقیقه شاهد بوت شدن سیستم عامل روی آن هستیم، اما تاکنون از خودمان سؤال کرده ایم در پس این بوت شدن چه اتفاقی می افتد تا سیستم عامل آماده اجرای کار شود؟ دانستن اینکه یک سیستم چگونه Boot می شود هم می تواند به درک شما در برقراری ارتباط بین سخت افزار و نرم افزار و همچنین رفع مشکلات ناشی از Boot نشدن سیستم عامل بسیار کمک کند. در اینجا می خواهیم با جزئیات بیشتری این فرآیند را قسمت به قسمت به شما معرفی کنیم. فرآیند کلی Boot سیستم عامل لینوکس شامل شش مرحله اصلی است که همانطور که در تصویر زیر نیز مشاهده می کنید این مراحل به BIOS، MBR، GRUB، Boot Loader پیشفرض، Kernel، Init و Runlevel تقسیم بندی می شود که در ادامه هر کدام از این مراحل، با ذکر جزئیات بیشتر توضیح داده می شوند.

**نکته:** تمامی مراحل بوت شدن تا وارد شدن به سیستم به همراه تمامی اتفاق های رخ داد (خطاها - هشدارها و موفقیت ها در انجام فرایند) همگی در فایل هایی مانند `var/log/message/` و `var/log/klog/` ثبت می شوند.





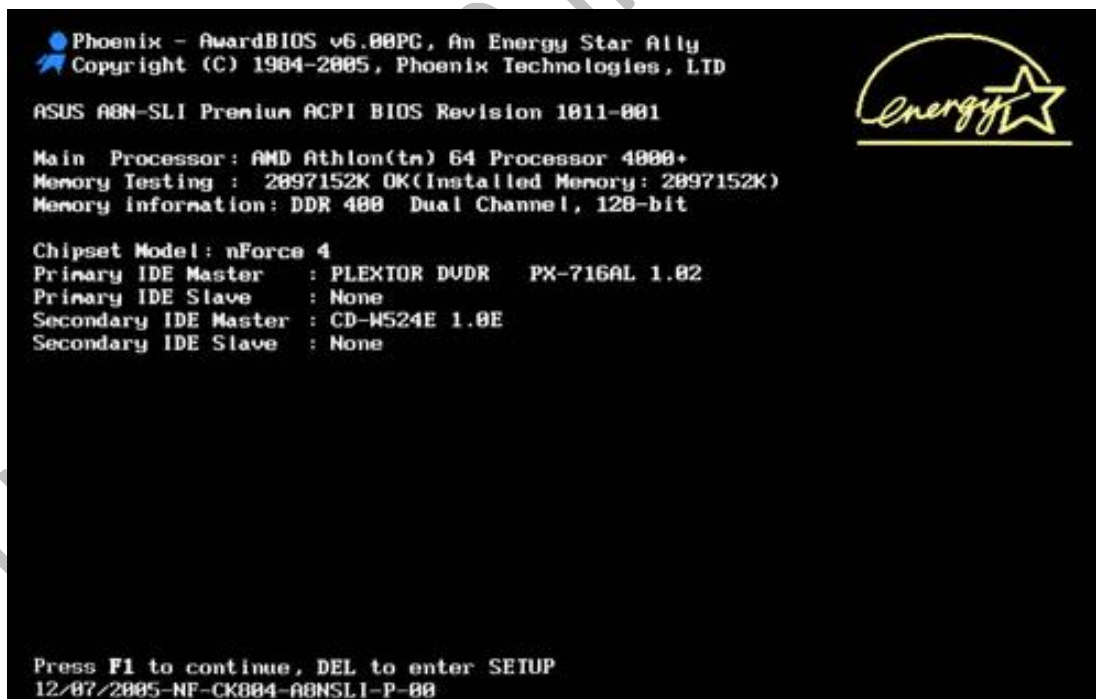
## مرحله اول Boot لاینوکس — POST

اولین فاز فرآیند Boot شدن سیستم فرآیند Startup است. زمانیکه شما دکمه Power سیستم را می زنید یا اینکه سیستم را Restart می کنید ، قدرت برق به SMPS یا Switched Mode Power Supply می رسد که در این قسمت برق تبدیل به AC یا DC می شود. برق DC به تمامی تجهیزات سیستم اعم از Motherboard ، هارد دیسک ها ، CD-ROM ، ماوس و کیبورد و ... داده می شود و آنها را روشن می کند. هوشمند ترین قطعه کامپیوتر شما قطعا پردازنده یا همان CPU شما است ، به محض اینکه برق به این دستگاه برسد این دستگاه شروع به انجام عملیات هایی می کند که در حافظه یا ROM سیستم وجود دارد. اولین دستوری که در این مرحله از طرف CPU صادر می شود این است که کنترل سیستم به دست BIOS سیستم داده شود ، BIOS مخفف کلمات Basic Input/Output System است. دستور CPU به BIOS این است که عملیات POST یا Power On Self-Test را انجام دهد. در این عملیات تجهیزات ضروری سیستم برای فرآیند Boot کاملاً آزمایش می شوند.

بعد از اینکه کنترل به دست BIOS سیستم افتاد دو چیز اتفاق می افتد :

1. فرآیند آزمایش POST یا Power On Self-Test

2. انتخاب اولین دستگاه قابل Boot در سیستم



**نکته:** لاینوکس به محض فعال شدن با یک فریم ورکی ، CPU را آپدیت می کند.

عملیات POST یا Power On Self-Test یک سری فرآیند پردازشی است که در دسترس بودن سخت افزارهای سیستم را بررسی می کند. BIOS لیستی از سخت افزارهایی که در آخرین Boot موفق سیستم فعال بوده اند را همیشه در خود دارد ، برای اینکه بررسی شود که یک سخت افزار برای انجام فرآیند Boot سیستم در دسترس است یا خیر برای هر کدام از این سخت افزارها که در لیست قرار دارند یک پالس الکتریکی فرستاده می شود.

اگر پالس الکتریکی به درستی پاسخ داده شود به معنی در دسترس بودن دستگاه مورد نظر و آماده بودن سخت افزار برای انجام فرآیند Boot است. اما اگر از سخت افزاری پالس دریافت نشود ، سخت افزار مورد نظر به عنوان سخت افزار معیوب یا مشکل دار در نظر گرفته می شود و فرض بر این گذاشته می شود که از سیستم جدا شده است. اینکار برای همه سخت افزارهای موجود در لیست BIOS انجام می شود و لیست جدید بر اساس آزمایش های جدید ایجاد می شود و در حافظه BIOS برای فرآیند Boot بعدی استفاده خواهد شد.

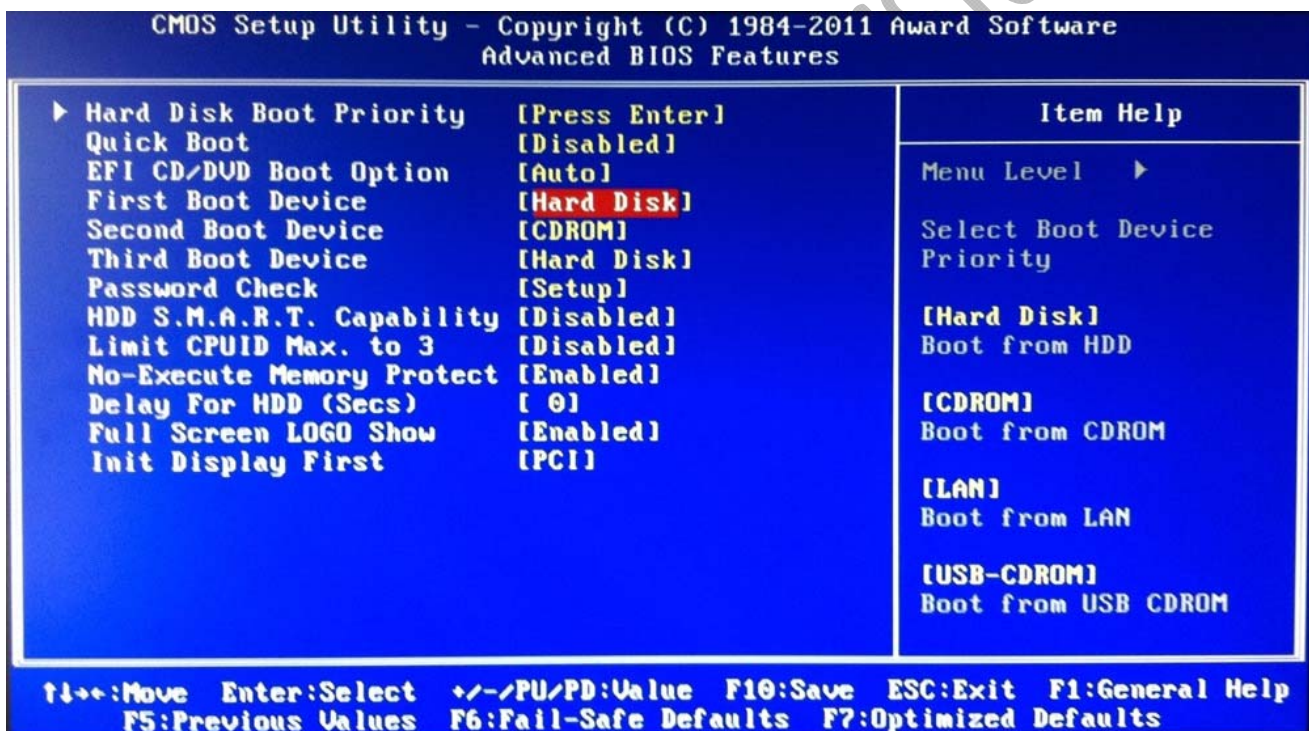
```
GNU GRUB version 0.95 (638K lower / 128960K upper memory)

root (hd0,0)
kernel /boot/vmlinuz-2.6.15-1-686 root=/dev/sda1 ro
initrd /boot/initrd.img-2.6.15-1-686
savedefault
boot

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

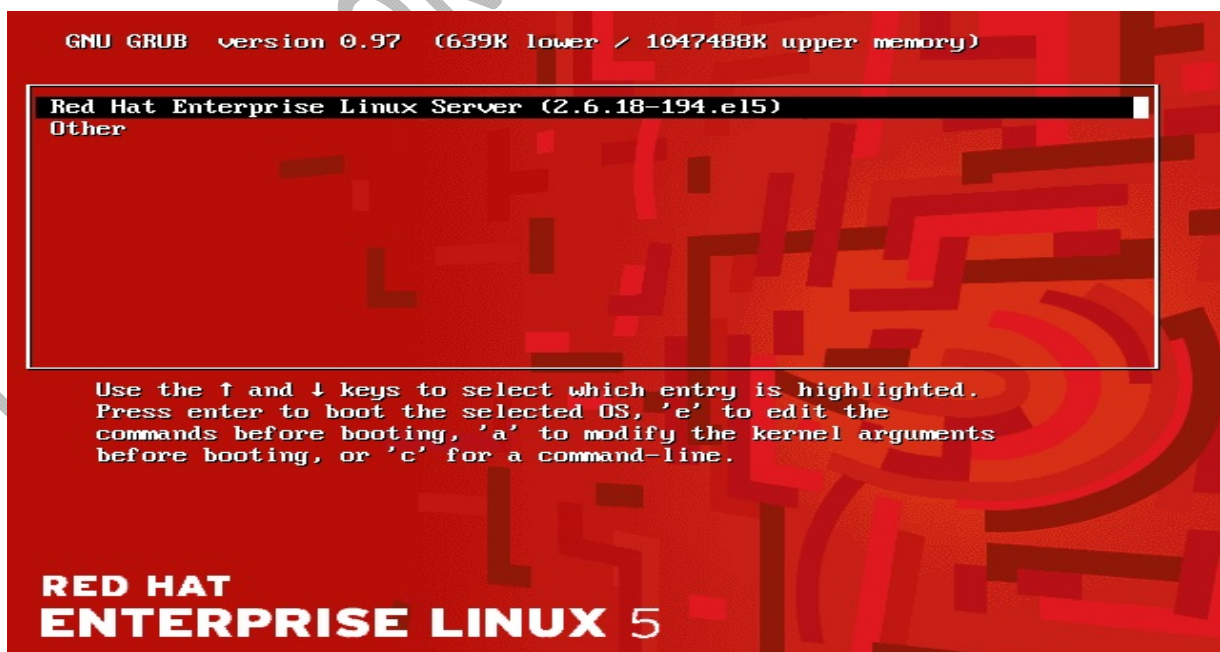
## انتخاب اولین دستگاه قابل Boot

زمانیکه عملیات های مربوط به POST کامل شدند، BIOS لیست دستگاه هایی که در دسترس هستند را خواهد داشت. در اینصورت BIOS به راحتی قادر خواهد بود اولین دستگاهی که در CMOS به عنوان Boot Device معرفی شده است را انتخاب کند. در این مرحله BIOS همیشه اولین دستگاه قابل Boot را انتخاب کرده و سپس کنترل سیستم را مجدداً به پردازنده یا CPU می سپارد. اگر فرض کنیم که اولین دستگاه قابل Boot بر روی سیستم پیدا نشود، بصورت خودکار BIOS به دنبال دومین دستگاه و به همین ترتیب همه دستگاه های قابل Boot را آزمایش می کند تا به نتیجه مورد نظر برسد. اگر BIOS نتواند هیچ دستگاه قابل Boot ای بر روی سیستم پیدا کند به شما پیامی به شکل No boot device found به معنی عدم پیدا کردن دستگاه قابل Boot نمایش خواهد داد.



## مرحله دوم Boot لینوکس - MBR

MBR مخفف Master Boot Record است. زمانی که BIOS مجدداً اختیار یا کنترل سیستم را در اختیار CPU قرار می‌دهد، BIOS سعی می‌کند MBR یا Master Boot Record را از اولین دستگاه قابل Boot بخواند و Load کند، در اینجا فرض را بر این گذاشته ایم که دستگاه Bootable ما هارد دیسک است. MBR یک قسمت کوچک از هارد دیسک‌ها است که فقط اندازه‌ای برابر 512 بایت دارد، **تاکید می‌کنم 512 بایت نه کیلو بایت**، این سکتور از هارد دیسک یا MBR در اولین نقطه شروع اطلاعات در هارد دیسک یا در انتهای هارد دیسک بر اساس نوع سازنده هارد دیسک قرار می‌گیرد. در واقع می‌توان به MBR یک Boot Loader اولیه گفت که وظیفه اصلی آن فراخوان کردن Boot Loader ثانوی سیستم است. MBR پیش زمینه اجرا شدن GRUB را فراهم می‌کند و این امکان را می‌دهد که GRUB به داخل حافظه Load شود و باعث بالا آمدن Kernel شود. از 512 بایت فضایی که به MBR اختصاص یافته در حدود 434 تا 446 بایت آن به Primary Boot Loader اختصاص یافته است، 64 بایت آن به Partition Table و در نهایت 6 بایت آن برای اعتبارسنجی MBR یا MBR Validation و Timestamp آن اختصاص یافته است. خوب تا اینجا کار، MBR به درستی Load شده اما MBR هیچگونه درکی از Kernel و نحوه Load کردن آن ندارد و نمی‌تواند مفهوم فایل سیستم را درک کند، در اینجا است که کار یک Boot Loader دیگر برای درایوهای فایل سیستم و Load کردن Kernel دیده می‌شود و این Secondary Boot Loader چیزی جز GRUB نیست. MBR معمولاً در هارد دیسک‌های لینوکسی در مسیر `/dev/hda` یا `/dev/sda` قرار گرفته است و مهمترین وظیفه آن Load کردن و اجرای GRUB است.



## مرحله سوم Boot لینوکس - GRUB

اگر با سیستم عامل لینوکس کمی کار کرده باشید یا آن را بصورت Dual Boot با ویندوز نصب کرده باشید حتما دیده اید که در مرحله ای از Boot سیستم از شما نوع سیستم عاملی که می خواهید آن را Boot کنید سؤال می کند. در واقع این همان GRUB ای است که در موردش صحبت شد. GRUB یک Boot Loader است با قابلیت Boot کردن چندین سیستم عامل بصورت همزمان که حتی می تواند ویندوز را نیز در کنار لینوکس Boot کند. اگر شما در صفحه انتخاب سیستم عامل هیچ گزینه را انتخاب نکنید GRUB بصورت خودکار سیستم عامل پیشفرض را انتخاب و آن را Boot می کند. تمامی تنظیمات مربوط به GRUB در فایل تنظیمات آن در مسیر `etc/grub.conf` قرار دارد. GRUB محتوی فایل های Image مربوط به سیستم عامل ها و `initrd` است و وظیفه اصلی آن نیز اجرا و Load کردن همین Image ها است. در ادامه نمونه ای از فایل `grub.conf` موجود در سیستم عامل لینوکس RedHat را مشاهده می کنید:

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
1 hiddenmenu
2 title Red Hat Enterprise Linux Server (2.6.18-
3 194.26.1.el5)
4     root (hd0,0)
5     kernel /vmlinuz-2.6.18-194.26.1.el5 ro
6 root=/dev/VolGroup00/root clocksource=acpi_pm divisor=10
7     initrd /initrd-2.6.18-194.26.1.el5.img
8 title Red Hat Enterprise Linux Server (2.6.18-
9 194.11.4.el5)
10    root (hd0,0)
11    kernel /vmlinuz-2.6.18-194.11.4.el5 ro
12root=/dev/VolGroup00/root clocksource=acpi_pm divisor=10
13    initrd /initrd-2.6.18-194.11.4.el5.img
14title Red Hat Enterprise Linux Server (2.6.18-
15194.11.3.el5)
16    root (hd0,0)
    kernel /vmlinuz-2.6.18-194.11.3.el5 ro
    root=/dev/VolGroup00/root clocksource=acpi_pm divisor=10
    initrd /initrd-2.6.18-194.11.3.el5
```

تا اینجای کار هنوز هیچ قسمتی از سیستم عامل لینوکس فعال نشده است ، در واقع همه فرآیندی که مربوط به سیستم عامل می شود در مرحله بعدی انجام می شود که GRUB کارهای خود را انجام می دهد و نوبت به Kernel سیستم عامل لینوکس

می رسد که توسط GRUB در حافظه RAM سیستم بارگذاری میشود . در قسمت بعدی در خصوص ادامه مراحل که Load شدن Kernel و همچنین فرآیند Init است توضیحاتی ارائه میشود.

## مرحله چهارم Boot لینوکس - Kernel

Kernel یا هسته یک سیستم عامل قلب تپنده آن سیستم عامل به شمار می رود و تمامی فرآیندهای سیستم عامل شما را مدیریت می کند. کرنل از هزاران مازول نرم افزاری تشکیل شده است. در مرحله قبلی GRUB به جایی رسید که Image های مربوط به Kernel و initrd را به داخل حافظه RAM سیستم Load کرد و از این مرحله به بعد همگی کارها بر عهده Kernel می باشد. Kernel به محض Load شدن تنظیمات مربوط به حافظه RAM اختصاص یافته به سیستم و همچنین پیکربندی کلیه سخت افزارهای سیستم را انجام می دهد سپس image مربوط به initrd را از حالت فشرده خارج می کند ، در حالت پیشفرض این image در حالت فشرده به شکل zlib یا zImage قرار گرفته اند. بعد از اینکه از حالت فشرده خارج شد با mount کردن آن ، درایورهای ضروری را Load می کند.





Load کردن و Unload کردن ماژول های Kernel لینوکس با استفاده از برنامه هایی به نام insmod و rmmod انجام می شود که در initrd image قرار گرفته اند. سپس Kernel به دنبال هارد دیسک می رود و بررسی می کند که در حالت LVM قرار دارند و یا اینکه RAID شده اند. سپس initrd را Unmount و فضای اختصاص یافته توسط این Image را نیز خالی می کند. Kernel پارتیشن ریشه یا root partition ای که در فایل grub.conf وجود دارد را در حالت فقط خواندنی یا read only به سیستم Mount کرده و در نهایت فرآیند init را اجرا می کند. توجه کنید که Kernel تا زمانی که سیستم شما خاموش نشده باشد بر روی حافظه RAM شما باقی می ماند.

کرنل بعد از اینکه کارهایش را انجام داد می آید پروسه ای را واسطه قرار می دهد و به آن اختیار می دهد تا همه کارها را مدیریت کند این پروسه init نام دارد.

```

295
audit(1216470015.968:3): policy loaded auid=4294967295 ses=4294967295
INIT: version 2.86 booting
        Welcome to Red Hat Enterprise Linux Server
        Press 'I' to enter interactive startup.
Setting clock (utc): Sat Jul 19 05:20:22 MST 2008          [ OK ]
Starting udev:                                           [ OK ]
Loading default keymap (us):                             [ OK ]
Setting hostname rhce-prep.example.com:                 [ OK ]
No devices found
Setting up Logical Volume Management:
    No volume groups found                               [ OK ]

Checking filesystems
/: clean, 4871/263232 files, 72321/263056 blocks
/home: clean, 117/130560 files, 27384/522080 blocks
/var: clean, 1165/130560 files, 65117/522080 blocks
/dev/md0: clean, 12/883872 files, 45604/883456 blocks
/usr: clean, 81733/524288 files, 427747/524120 blocks
/boot: clean, 33/66264 files, 24068/265040 blocks        [ OK ]

Remounting root filesystem in read-write mode:          [ OK ]
Mounting local filesystems:                             [ OK ]
Enabling local filesystem quotas:                       [ OK ]

```

## مرحله پنجم Boot لینوکس - Init Process

اگر دقت کرده باشید بعد از اینکه از صفحه انتخاب سیستم عامل GRUB عبور کردید به یکباره یک سری خطوط به شما نمایش داده می شود و در کنار آنها بعضا کلماتی مثل OK و Failed را مشاهده می کنید که به سرعت از جلوی چشم شما عبور می کند (شکل صفحه قبل)، انگار که سیستم در حال Start کردن سرویس های لازم جهت اجرای سیستم عامل میباشد. این دقیقا درست است، به محض اینکه Kernel شما Load شد یک فرآیند یا Process به نام init توسط Kernel اجرا می شود که اولین Process اجرا شده توسط Kernel است. init مخفف کلمه initialization بوده که در ترجمه فارسی به معنی مقدار دهی اولیه است اما شما در اینجا از init به عنوان پردازشی یاد کنید که همه پردازش های دیگر مورد نیاز سیستم عامل را فراخوانی و اجرا می کند. زمانیکه init اجرا شد به عنوان فرآیند والد یا Parent Process در سیستم عامل لینوکس در نظر گرفته می شود. اولین کاری که init انجام می دهد این است که محتویات فایل تنظیمات خودش یا initialization file ای که در مسیر `etc/inittab/` وجود دارد را بخواند. محتویات این فایل به init می گوید که یک اسکریپت تنظیمات اولیه محیطی یا `environment configuration script` را اجرا کند که در این اسکریپت تعیین مسیرها یا `path` ها، فرآیند `Swapping`، بررسی فایل سیستم و ... انجام می شود. تقریبا می توان گفت اجرای این اسکریپت هر چیزی که سیستم شما نیاز دارد تا فرآیند مقداردهی اولیه یا همان initialization را انجام دهد را شامل می شود، حتی تعیین کردن ساعت سیستم، پورتهای سریال و ... همه در این مرحله انجام می شود. مخزن لینوکس دایرکتوری `init.d` است و تمامی فایل هایی که در دایرکتوریهای `rc` قرار دارند لینک به این دایرکتوری می باشد.

به محض اینکه یک سرویس در لینوکس نصب می شود یک اثر از خودش در `init.d` بر جا می گذارد. در نسخه های قدیمی لینوکس init فقط یک فایل پیکربندی داشت که گاهی به چند صد خط می رسید. یعنی تمام فایل هایی که در دایرکتوریهای `rc` قرار داشت در یک فایل کانفیگ بزرگ جمع شده بودند که این خود یک ضعف و آسیب پذیری به شمار می آمد، چون برای این فایل مشکلی ایجاد میشد سیستم `up` نمی شد. لذا `redhat` سیستمی به نام `upstart` را ایجاد کرد که در اصل همان `init` است منتها `event base` عمل می کند نه `parallel`. در حافظه سیستم هم نام آن `init` است. در اصل آن فایل چند صد خطی بزرگ به تعدادی فایل شکسته شده تا مدیریت و پایداری سیستم تنظیم بشود. `init` اولین نرم افزاری است که در `user mode` کار میکند.

`Init` همچنان به خواندن فایل `etc/inittab/` ادامه می دهد، در ادامه این فایل به `init` اعلام می کند که سیستم قرار است چگونه در هر `run level` پیکربندی شود و همچنین `run level` پیش فرض را نیز تنظیم می کند. `Run level` در واقع پیکربندی فرآیندها یا پردازش های موجود در سیستم است. تمامی سیستم عامل های خانواده `Unix` توانایی این را دارند که



با پیکربندی های پردازشی مختلف اجرا شوند برای مثال ما می توانیم به سیستم عامل لینوکس بگوییم که در حالت تک کاربره یا single user mode اجرا شود که در این حالت اگر init اجرا شود ما می گوییم init در level 1 اجرا شده است ، در زبان دیگر به run level ای که single user mode است run level S نیز گفته می شود. در این حالت ( Single User Mode ) فقط مدیر سیستم می تواند به سیستم متصل شود. معمولاً زمانی از چنین run level ای استفاده می شود که می خواهیم وظایف نگهداری سیستم یا maintenance task های سیستم را بدون ریسک تخریب سیستم یا داده های کاربران انجام دهیم. طبیعتاً در چنین run level ای ما نمی خواهیم هیچگونه سرویسی به کاربران ارائه دهیم و سرویس های کاربری معمولاً در این run level غیرفعال یا Disable هستند. Run Level دیگری که استفاده می شود به نام run level reboot شناخته می شود یا run level 6 که وظیفه shutdown کردن همه سرویس ها و دستورالعمل های وابسته به آن و همچنین restart کردن سیستم را بعهده دارد. همانطور که در مثال زیر مشاهده می کنید شما می توانید با استفاده از دستور who متوجه شوید که در چه run level ای قرار دارید :

```
#who -r
```

```
run-level          2          2015-11-17    09:52          last=S
```

بعد از اینکه run level پیشفرض برای سیستم در نظر گرفته شد ، init با توجه به محتویات دایرکتوری rc ای که به level مربوطه اختصاص یافته است تمامی پردازش ها یا Process های پس زمینه ای که برای اجرا شدن سیستم مورد نیاز هستند را اجرا می کند. Init هر کدام از اسکریپت های kill را که با حرف K شروع می شوند را با یک پارامتر stop اجرا می کند و سپس تمامی اسکریپت های Start را که با S شروع می شوند را اجرا می کند و سرویس ها و برنامه های کاربردی که در run level مربوطه وجود خواهند داشت را اجرا خواهد کرد. نکته جالب در خصوص این اسکریپت ها این است که همانند سرویس های ویندوز شما می توانید آنها را بعد از وارد شدن به سیستم عامل لینوکس بصورت دستی نیز stop و یا start کنید ، اینکار براحتی با استفاده از دستور service httpd stop یا service httpd start در مسیر etc/init.d/httpd/ قابل اجرا است ، البته دقت کنید که در این اینجا ما سرویس httpd را مثال زدیم ، شما می توانید هر سرویس دیگری را به همین شکل start یا stop کنید فقط فراموش نکنید که برای انجام تغییرات سیستمی حتماً باید دسترسی ریشه داشته باشید.

توجه کنید که در زمان startup سیستم معمولاً دو اسکریپت rc2.d و rc3.d اجرا می شوند. در این حالت هیچ سرویسی در حالت stopped قرار نمی گیرد یا حداقل می توان گفت بصورت دائمی در حالت stopped قرار نمی گیرد. هیچکدام از اسکریپت هایی که در پوشه etc/rc<x>.d/ قرار گرفته اند سرویس ها را stop یا start نمی کنند. در عوض تمامی فایل هایی که در پوشه etc/rc<x>.d/ قرار گرفته اند به عنوان یک لینک عمل میکنند که به اسکریپت هایی که در پوشه etc/init.d/ قرار گرفته اند اشاره می کنند. به اینگونه لینک ها symbolic link گفته می شود ، یک symbolic link چیزی بیشتر از یک فایل که به یک فایل دیگر اشاره می کند نیست و زمانی استفاده می شود که شما می خواهید بدون ایجاد کردن و یا حذف کردن یک فایل، اسکریپت آن را اجرا کنید و سرویس ها را start و یا stop کنید.

در نهایت در این قسمت اسکریپت های مربوط به run level مورد نظرتان انتخاب و اجرا خواهند شد و سیستم به سراغ مرحله بعدی می رود. لیست run level های مختلف در سیستم عامل لینوکس را می توانید در زیر با مشخصات آنها مشاهده کنید .

## Run levels

init 0 : Shutdown the system or halt

init 1 : Single-user mode.

init 2 : Multi-user mode with out networking support.

init 3 : Multi-user mode with networking support.

init 4 : Unused.

init 5 : Graphical user interface.

init 6 : Reboot the system.

## مرحله ششم Boot لینوکس - Run Level

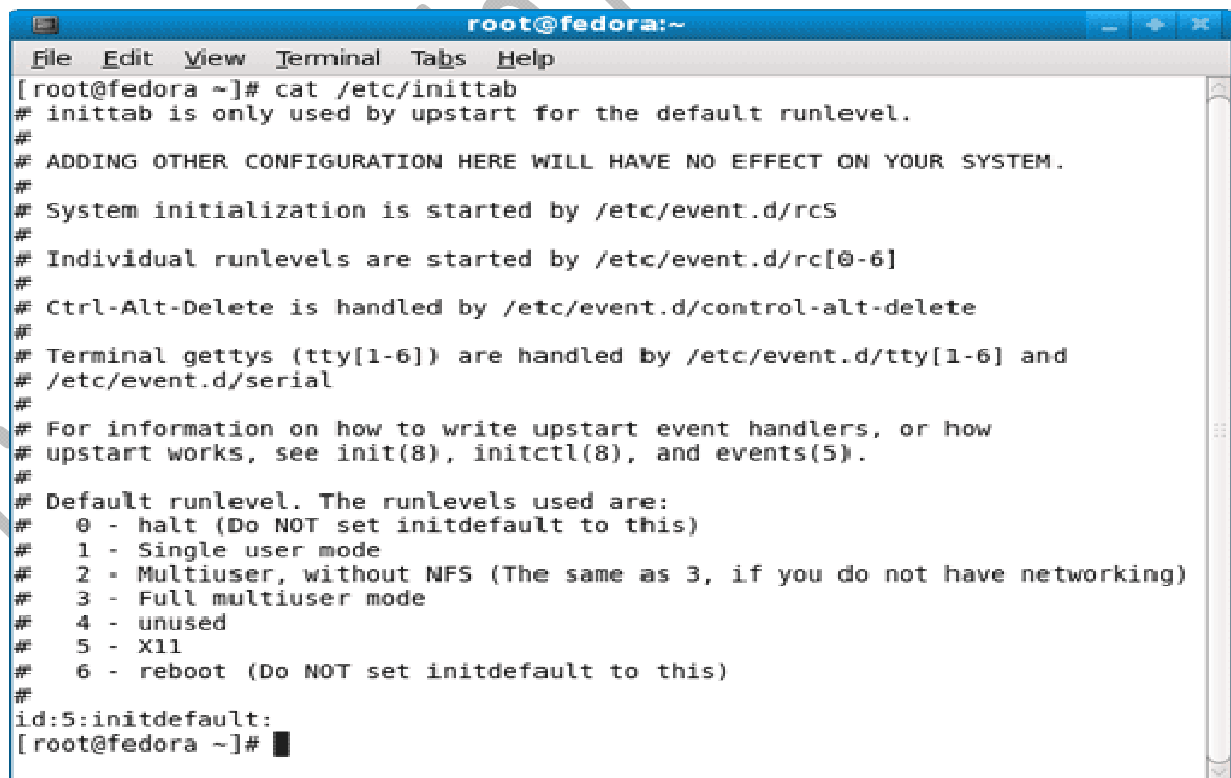
همانطور که در مرحله قبل مشاهده کردید در فرآیند init یک run level انتخاب می شود که بر حسب انتخابی که انجام می شود یک سری اسکریپت از پوشه های مورد نظر اجرا می شوند. startup script های مربوط به init در پوشه /etc/rc.d/ قرار گرفته اند اما اسکریپت هایی که برای run level ها استفاده می شوند در subdirectory هایی به شکل /etc/rc.d/rc0.d/ تا /etc/rc.d/rc6.d/ بر اساس انتخاب init از بین run level های 0 تا 6 اجرا می شوند. در آخر نیز init هر چیزی که در پوشه /etc/rc.d/rc.local/ پیدا می کند را فارق از اینکه در چه run level ای کار می

کند اجرا خواهد کرد. در زیر، پوشه های مربوط به انواع run level script لیست شده است که متناسب با run level انتخاب شده در init اجرا خواهند شد .

## Runlevel Directory

```
/etc/rc.d/rc0.d
/etc/rc.d/rc1.d
/etc/rc.d/rc2.d
/etc/rc.d/rc3.d
/etc/rc.d/rc4.d
/etc/rc.d/rc5.d
/etc/rc.d/rc6.d
```

به این موضوع دقت کنید که اسکریپت های آخری که عنوان کردیم در پوشه rc.local وجود دارد در همه توزیع های لینوکس ممکن است وجود نداشته باشند ، در نهایت همه چیز اگر به خوبی پیش برود شما می توانید صفحه login مربوط به سیستم عامل لینوکس را مشاهده کنید . در زیر نمونه ای از محتویات فایل inittab را مشاهده می کنید.



```
root@fedora:~
File Edit View Terminal Tabs Help
[root@fedora ~]# cat /etc/inittab
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/event.d/rcS
#
# Individual runlevels are started by /etc/event.d/rc[0-6]
#
# Ctrl-Alt-Delete is handled by /etc/event.d/control-alt-delete
#
# Terminal gettys (tty[1-6]) are handled by /etc/event.d/tty[1-6] and
# /etc/event.d/serial
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(8), initctl(8), and events(5).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
[root@fedora ~]#
```

پس از 6 مرحله بالا باید صفحه Login یا به صورت متنی و یا بصورت گرافیکی نشان داده می شود. در اینجا باید نام کاربری پسورد خود را وارد کنید. اگر نام کاربری و پسوردتان درست باشد با ایجاد یک ترمینال توسط برنامه `mingetty` که یکی از 6 ترمینال `tty1` تا `tty6` هستند، یک جلسه یا `Session` به نام یوزر لاگین کرده اختصاص داده می شود. اما روند ایجاد یک ترمینال چگونه است؟

## روند ایجاد یک ترمینال

`Shell` یا پوسته، ابزاری است که دستور ها را از شما می گیرد و آنها را تحلیل می کند یا یک خروجی مناسب یا یک خطا در صورت مشکل داشتن ورودی، نشان می دهد. پروسه ای به نام `fork` وجود دارد که یک نمونه از خود فرایند را عینا ایجاد می کند. وظیفه `fork` ایجاد پروسس `child` از فرایند `parent` میباشد. (از فرایند اصلی یا والد یک فرایند فرزند ایجاد میکند) پس از آخرین گام بالا دستور `/sbin/mingetty` یک ترمینال را ایجاد می کند. `Shell` و `Terminal` دو چیز متفاوت هستند. در اصل پوسته یا شل کدهایی هستند که دستور های ورودی از طریق ترمینال یا کنسول را تحلیل می کند و ترمینال یک ابزاری برای کار با شل است. در لینوکس تعدادی پوسته وجود دارند مانند `bash, sh, csh, ksh, zsh` که لیست نام این شل ها در لینوکس در فایل `shells` که با دستور زیر قابل مشاهده است قرار دارد.

```
#less /etc/shells
```

اما یک شل (هر کدام که باشد) چگونه در اختیار شما قرار می گیرد. همانطور که گفتیم شل بعنوان تحلیل گر دستور های ورودی و ترمینال نیز بعنوان رابطی برای اتصال به شل استفاده می شود. بصورت پیش فرض `bash` بعنوان پوسته پیش فرض لینوکس است. به محض `login` شدن و ایجاد یک ترمینال توسط `mingetty` مثل `tty1`، فرایند `init` نیز یک فرایند دیگر به نام `bash` را می سازد که پوسته `bash` را فعال می کند. بطور خلاصه می توان گفت که :

پس از مرحله ششم از مراحل بوت شدن، صفحه `login` نشان داده می شود و فرایند `init` نیز شل `bash` را از مسیر `bin/bash/` ایجاد و کنترل می کند (توسط عمل `fork` یک نمونه از `bash` می سازد) سپس در صورت درست وارد کردن نام کاربری و پسورد، برنامه `mingetty` یک ترمینال مانند `tty2` را در اختیار شما قرار می دهد. توجه کنید که خود فرایند `mingetty` بعنوان فرزند `bash` و خود `bash` نیز بعنوان فرزند فرایند `init` نی باشد.

اگر بخواهیم از ترمینال خارج شویم یا بهتر اینکه بخواهیم جلسه ایجاد شده را از بین ببریم باید از دستور `logout` یا `exit` استفاده کرد. به محض اجرای این دستور ها جلسه کاری تمام می شود و دوباره صفحه `login` نشان داده می شود.

**نکته:** در حین بوت شدن ممکن است اعمال دیگری مانند اجرای دستور `fsck` یا اعمال دیگری انجام شود ولی 6 مرحله بالا اساس راه اندازی یک سیستم عامل لینوکس و یا حتی هر سیستم عامل یونیکسی دیگری است.

Linux Cookie in Persian v.2

## فصل دوم

### معرفی GRUB در لینوکس



## مقدمه ای بر Bootloader

برای اینکه بخواهیم از سیستم عامل استفاده کنیم، یک سری برنامه نرم افزاری وجود دارد که باید ابتدا اجرا شوند تا محیط سیستم عامل را بتوانید ببینید. یکی از این نرم افزارها به عنوان **boot strap loader** یا **Bootloader** معروف است. در واقع کاری که **Bootloader** انجام می دهد همانطور که از نامش هم پیداست، این است که سیستم عامل اصلی کامپیوتر را **load** می کند. نرم افزارهای سیستم عامل در هارد دیسک، **CD**، **DVD**، حافظه های فلش **USB** و امثال اینها ذخیره می شوند، اما این نرم افزارها توسط **CPU** یا پردازنده کامپیوتر قابل دسترس نیستند زیرا **CPU** فقط و فقط می تواند برنامه هایی را اجرا کند که در **ROM** سیستم قابل دسترس هستند. برنامه هایی که در **RAM** وجود دارند با استفاده از **Boot Loader** هایی مثل **BIOS**، **SLOF**، **OpenBoot**، **OpenBIOS**، **BOOTMGR**، **NTLDR**، **Syslinux**، **GRUB** و **LILO** قابل دسترس خواهند شد.

یکی از واژه های جالبی که برای افراد تازه کار با محیط لینوکس همیشه کمی ترس در بر دارد واژه **Boot Loader** است. دلیل آن هم کاملاً مشخص است، افرادی که به تازگی با محیط لینوکس آشنا شده اند و با آن کار می کنند اکثراً از محیط سیستم عامل ویندوز به لینوکس مهاجرت کرده اند، ممکن است در دنیای ویندوزی به هیچ عنوان اسم **Bootloader** را نیز نشنیده باشند. کاربران ویندوز فقط متوجه می شوند که ویندوز آنها **Boot** می شود و از روندی که در پس زمینه **Boot** شدن این سیستم عامل طی میشود هیچ اطلاعی ندارند و برایشان این فرآیند کاملاً نامحسوس است.

برخی اوقات ممکن است به مشکلاتی در **Boot** سیستم بر بخورند و با استفاده از **Windows Recovery Console** مشکلات را برطرف کنند و در همین حد برایشان کفایت می کند. اما وقتی صحبت از لینوکس می شود یعنی همه چیز قابل بررسی و مشاهده است، فرآیند **Boot** شدن سیستم عامل و به ویژه نحوه **Boot** شدن، و نرم افزاری که اینکار را برای آنها انجام می دهد مهم می شود و اینجاست که تازه کاران باید با واژه **Bootloader** و نحوه عملکرد آن در لینوکس بیشتر آشنا شوند. (مثل من) در این فصل ما می خواهیم یک درک ابتدایی از عملکرد **GRUB Bootloader** که اصلی ترین **Bootloader** لینوکسی است پیدا کنیم.

اگر تا کنون کمی با سیستم عامل لینوکس کار کرده و حداقل با ساختار پارتیشن بندی، فایل سیستم و ساختار آن، دایرکتوری های پیشفرض لینوکس و دستورات خط فرمان اولیه لینوکس آشنایی پیدا کرده باشید، گام بعدی این است که دانش خود را فراتر بگذارید و بدانید که **GRUB** که **Bootloader** اصلی لینوکس است چگونه کار می کند و اصلاً منوی **Boot** که در زمان روشن کردن سیستم مشاهده می کنید چگونه ایجاد می شود و اگر به مشکلات **Boot** در لینوکس برخورد کردید چگونه آنها را برطرف کنید، اینکار با استفاده از تغییرات منوی **GRUB** انجام می شود و شما باید با این محیط و به ویژه کار با محیط های چند سیستم عاملی در **GRUB** آشنایی پیدا کنید.

## GRUB چیست ؟

GRUB مخفف کلمه های Grand Unified Bootloader است که می تواند سیستم عامل هایی مانند Linux ، Mach4 ، vSAT ، DOS و بسیاری دیگر از سیستم عامل ها را boot کند. GRUB توانایی Load کردن انواع هسته یا Kernel را با انواع مختلف قالب های باینری دارد. امکانات و تنظیمات مختلفی در فایل config این Bootloader وجود دارد که اجازه Multiboot کردن سیستم عامل ها را به سیستم می دهد. همچنین انتخاب پارتیشن سیستمی که قرار است سیستم عامل و هسته آن یا Kernel از آن Load شوند نیز بر عهده GRUB می باشد. GRUB از چندین قالب اجرایی پشتیبانی می کند و همچنین نیازی به انجام داده ترجمه ها و تبدیل های هندسی هم ندارد. GRUB یک محیط command prompt دارد که تا حد زیادی شبیه به bash shell لینوکس است و این اجازه را به کاربران می دهد که یک سیستم عامل نصب شده را از روی CD-ROM و یا USB بوت کند.

Boot Loader یک کد نرم افزاری است که قبل از اجرای سیستم عامل اجرا می شود و کمک می کند که سیستم عامل در سیستم به درستی اجرا و در محیط عملیاتی قرار بگیرد. Boot Loader ها معمولاً روشی های مختلفی برای Boot کردن Kernel سیستم عامل ها دارند و همچنین امکاناتی را فراهم می کنند تا در صورت بروز مشکل برای فرآیند Boot سیستم عامل از طریق دستورات یا ارسال یک سری مقادیر به Kernel مشکلات مربوطه را برطرف کند تا به درستی انجام شود . اگر فرض را بر این بگذارید که یک کامپیوتر با سیستم عامل Redhat Enterprise Linux روشن شود ، سیستم عامل Redhat Enterprise Linux ای که روی سیستم نصب شده است ابتدا توسط Boot Loader از محل نصب به حافظه سیستم منتقل می شود تا شروع به کار کند. برنامه Boot Loader یک کامپیوتر بر روی هارد دیسک آن کامپیوتر ذخیره شده و وظیفه اصلی آن Load کردن Kernel سیستم عامل و فایل های ضروری مورد نیاز برای Boot سیستم عامل به داخل حافظه RAM سیستم می باشد. اما توجه کنید که همه سیستم عامل ها از Boot Loader های مشابهی استفاده نمی کنند برای مثال RedHat Enterprise Linux از Boot Loader های مختلفی برای معماری های سخت افزاری کامپیوتر مختلف استفاده می کند ، در ادامه برخی از این معماری ها و Boot Loader های مورد استفاده در آنها در این سیستم عامل را مشاهده می کنید:

معماری سیستم	نوع Boot Loader مورد استفاده
AMD32, AMD64,	GRUB
Intel x86, EMT64T	GRUB
Intel Itanium	ELILO
IBM eServer System i	OS/400
IBM eServer System p	YABOOT



GRUB در واقع Boot Loader پیشفرض لینوکس برای معماری های AMD32 ، AMD64 ، Intel x86 و Intel 64T است. GRUB این امکان را به شما می دهد که سیستم عامل مورد نظرتان را در هنگام Boot سیستم انتخاب کنید و از همه مهمتر اینکه می توان در زمان Boot شدن Kernel سیستم عامل، به آن یک سری آرگومان ارسال کنید. GRUB به نسبت Boot Loader قدیمی تر که به نام LILO وجود داشت ساختار پیچیده تری دارد اما طبیعتا پیشرفته تر و قابل اعتماد تر از LILO است.

تا اینجا کار متوجه شدیم که کار Boot Loader پیدا کردن Linux Kernel و سایر فایل های ضروری آن و Load کردن آن در حافظه است ، سپس همین Boot Loader هسته یا Kernel را Start می کند تا بتواند پردازش های خودش را انجام دهد. علاوه بر این که Boot Loader وظیفه کنترل کردن چگونگی Boot سیستم را نیز بر عهده دارد ، اگر شما یک سیستم Dual Boot داشته باشید ، این Boot Loader است که به شما امکان انتخاب سیستم عامل مورد نظرتان را در هنگام Startup می دهد. زمانیکه کامپیوتر Boot می شود BIOS سیستم که در حافظه ROM سیستم که یک حافظه فقط خواندنی است قرار دارد از رکورد اول هارد دیسک شما یا همان Master Boot Record یا MBR برنامه Boot Loader را فراخواهی می کند. البته برخی از نسخه های لینوکس وجود دارند که مستقیما اینکار را از حافظه Flash انجام می دهند و به سمت BIOS ارسال می کنند . به این موضوع توجه کنید که لینوکس ذاتا به گونه ای طراحی شده است که نیازی به Boot Loader نداشته باشد و مستقیما از طریق سکتورهای هارد دیسک Boot شود.

### اما استفاده کردن از Boot Loader در لینوکس مزایای زیر را برای ما فراهم می کند :

- امکان Boot کردن چندین سیستم عامل یا Multi Boot
- امکان ارسال پارامتر یا آرگومان به کرنل سیستم عامل که در هنگام Boot برای موارد رفع اشکال کاربردی است.
- امکان Load کردن Kernel های متفاوت بصورت Interactive ، زمانیکه می خواهید Kernel را بروز رسانی کنید کاربرد دارد .

تمامی تنظیمات مربوط به GRUB در فایل تنظیمات آن در مسیر `/etc/grub.conf` و یا در مسیر `/boot/grub/grub.conf` قابل دسترس است. GRUB محتوی فایل های Image مربوط به سیستم عامل و `initrd` است و وظیفه اصلی آن نیز اجرا و Load کردن همین Image ها می باشد.

در ادامه نمونه ای از فایل `grub.conf` موجود در سیستم عامل لینوکس RedHat را مشاهده می کنید:

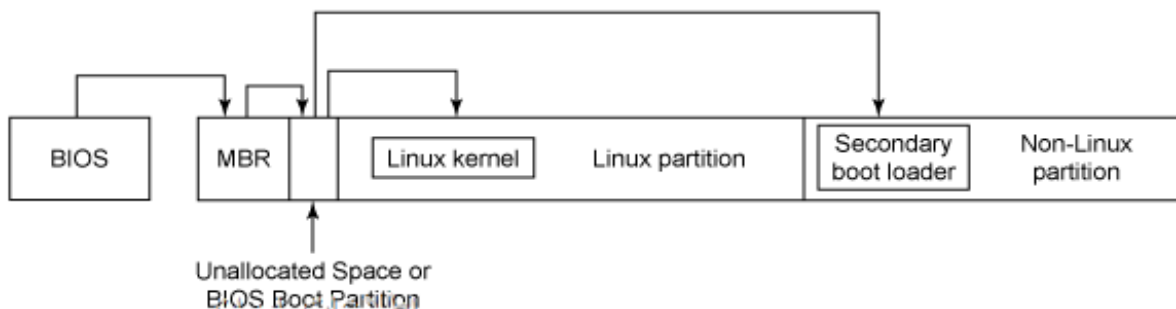
```

default=0 سیستم عامل پیش فرض را نشان میدهد
timeout=5 زمان پیش فرض صفحه نمایش بوت
1 splashimage=(hd0,0)/grub/splash.xpm.gz تصویر صفحه لود گراب و سیستم عامل را مسیر دهد میکند
2 hiddenmenu
3 عنوان سیستم عامل را نشان میدهد title Red Hat Enterprise Linux Server (2.6.18-194.26.1.el5)
4 آدرسی است که گراب اطلاعات خودش را از آن به کرنل می دهد
5     root (hd0,0)
6     kernel /vmlinuz-2.6.18-194.26.1.el5 ro root=/dev/VolGroup00/root
7 clocksource=acpi_pm divisor=10
8     initrd /initrd-2.6.18-194.26.1.el5.img
9 title Red Hat Enterprise Linux Server (2.6.18-194.11.4.el5)
10    root (hd0,0)
11    kernel /vmlinuz-2.6.18-194.11.4.el5 ro root=/dev/VolGroup00/root
12clocksource=acpi_pm divisor=10
13    initrd /initrd-2.6.18-194.11.4.el5.img
14title Red Hat Enterprise Linux Server (2.6.18-194.11.3.el5)
15    root (hd0,0)
16    kernel /vmlinuz-2.6.18-194.11.3.el5 ro root=/dev/VolGroup00/root
    clocksource=acpi_pm divisor=10
    initrd /initrd-2.6.18-194.11.3.el5

```

## مراحل کار GRUB در هنگام Load کردن سیستم عامل

در خصوص GRUB مختصر توضیحی ارائه شد حال در خصوص مراحل کاری آن توضیحاتی ارائه می شود و اینکه در چند مرحله و چگونه یک سیستم عامل توسط این Boot Loader شروع به کار می کند. بصورت کلی اکثر Boot Loader ها دو یا بیشتر از دو مرحله کاری دارند که در GRUB هم به همین شکل است، اولین مرحله از کار GRUB قرار دادن یا Load کردن خود GRUB داخل حافظه RAM است:



## مرحله اول Boot Loader

اولین گام این است که BIOS از داخل Master Boot Record یا MBR برنامه Boot Loader اصلی سیستم را به داخل حافظه RAM منتقل می کند. برنامه Boot Loader فضایی کمتر از 512 بایت در MBR را به خود اختصاص داده و این فضا اینقدر کم است که امکان اجرای دستورات پیچیده مورد نیاز برای Load کردن یک سیستم عامل را در خود

ندارد به همین دلیل Boot Loader سیستم فعالیتی را انجام می دهد که توسط آن مرحله یک و نیم ( در اصطلاح فنی ) اجرا میشود.

## مرحله یک و نیم Boot Loader

مرحله یک و نیم در واقع به این دلیل نامگذاری شده که یک مرحله میانی است که بین مرحله اول و مرحله دوم قرار گرفته و برخی از سخت افزارها برای درست اجرا کردن Boot Loader به این مرحله میانی نیاز دارند. اگر بخواهیم تخصصی تر صحبت کنیم ، مرحله یک و نیم زمانی اتفاق می افتد که پارتیشن boot/ در فاصله ای بیشتر از سیلندر 1024 هارد درایو قرار گرفته باشد یا اینکه شما از حالت LBA استفاده می کنید. مرحله یک و نیم ( GRUB Stage 1.5 ) در 30 کیلوبایت اول هارد دیسک شما بلافاصله بعد از MBR و قبل از اولین پارتیشن قرار گرفته است ، این فضا به گونه ای بهینه سازی شده که درایورهای فایل سیستم و ماژول های آنها را در خود نگه دارد . بعد از فعال شدن مرحله یک و نیم سیستم آماده می شود که مرحله دوم را شروع کند و محل انجام شدن مرحله دوم در اینجا کاملاً مشخص است ، معمولاً در فایل سیستم ، مرحله دوم را از boot/grub/ شروع می شود.

## مرحله دوم Boot Loader

در این مرحله ، در واقع Boot Loader ثانوی سیستم یا Secondary Boot Loader اجرا می شود و GRUB تازه شروع به کار می کند. Secondary Boot Loader در محلی از هارد دیسک شما در یکی از بوت سکتورهای اولین پارتیشن اکتیو سیستم قرار گرفته است. در اینجاست که میتوانید منوهای GRUB را مشاهده می کنید و محیط خط فرمانی آن را نیز ببینید ، البته در صورتیکه Boot Loader شما از نوع GRUB باشد ، اگر در این قسمت شما چندین سیستم عامل را بصورت Dual Boot در کنار هم داشته باشید امکان انتخاب کردن آنها را خواهید داشت. در این قسمت است که شما می توانید به Kernel سیستم عامل آرگومان یا دستور ارسال کنید.

## Load کردن سیستم عامل یا Kernel

در این مرحله بعد از اینکه در مرحله قبل سیستم عامل مورد نظر خود را انتخاب کردید ، GRUB تصمیم می گیرد که چه سیستم عاملی را Start و سپس Kernel مورد نظر آن را به همراه initrd به داخل حافظه RAM منتقل کند. در این قسمت است که کنترل کامل سیستم به دست سیستم عامل سپرده می شود و GRUB از کار خارج می شود . سابقاً سیستم عامل های توزیع لینوکس از یک Boot Loader دیگر به نام LILO استفاده می کردند که با معرفی شدن GRUB دیگر بسیار کم از LILO استفاده می شود و تقریباً می توان گفت GRUB بصورت کامل جایگزین LILO شده

است. از امکانات مهمی که GRUB به ما ارائه می دهد قابلیت محافظت با استفاده از رمز عبور ، امکان ارسال پارامترهای Boot مختلف و انجام عملیات لود سیستم بدوم هنگ کردن و ایجاد مشکل برای فایل تنظیمات GRUB و همچنین امکان همزمان Boot کردن سیستم عامل ویندوز در کنار لینوکس را می توان نام برد.

از طرفی GRUB همانطور که قبلا نیز اشاره شد قابلیتی به نام Logical Block Addressing یا LBA را پشتیبانی می کند که به شما این امکان را می دهد که فایل های boot سیستم را بسیار ساده تر پیدا کنید . روش Boot ای که سیستم عامل Redhat Enterprise Linux استفاده می کند به روش Direct Loading Method معروف است ، این به دلیل آن است که Boot Loader بدون نیاز به مرحله میانی سیستم عامل و Kernel را Load می کند. در حالتی دیگر از Boot که به Chain Booting یا بوت کردن زنجیره ای معروف است ، MBR به اولین سکتور از پارتیشن اشاره می کند که شامل سیستم عامل نیز می شود. MBR با استفاده از فایل هایی که در این قسمت از GRUB وجود دارد ، هم از حالت Direct و هم از حالت Chain پشتیبانی می کند. اینکار باعث می شود که تقریباً هر نوع سیستم عاملی قابل Boot باشد

## LILO چیست ؟

LILO مخفف کلمات Linux Loader است و همانطور که از نامش هم پیداست Boot Loader ای است که ویژه سیستم عامل لینوکس طراحی و پیاده سازی شده است. LILO در واقع کدی است که BIOS در هنگام Startup سیستم در داخل حافظه کامپیوتر Load میکند. همانند GRUB بوت لودر LILO هم می تواند سیستم عامل ها را از یک منبع خارجی مثل فلاش مموری یا هارد دیسک اکسترنال boot کند. کاری که LILO در چنین حالتی انجام می دهد این است که Master Boot Record یا MBR را روی دستگاه خارجی نصب می کند و با این حالت براحتی در زمان بوت سیستم Kernel سیستم عامل را پیدا می کند و آن را داخل حافظه RAM سیستم Load می کند و سیستم عامل را بالا می آورد . LILO بصورت پیشفرض به عنوان Boot Loader اصلی لینوکس ارائه می شد تا اینکه به مرور زمان با GRUB جایگزین شد. برخلاف GRUB در LILO شما توانایی Boot کردن از طریق شبکه را ندارید اما در حالت کلی LILO ساده تر و دارای پیچیدگی های کمتری نسبت به GRUB است ، اگر بخواهیم بصورت خلاصه تفاوت های این دو Boot Loader را معرفی کنیم به شکل زیر بیان میشود:

- GRUB یک Boot Loader است که می توان در لینوکس ، DOS و سایر سیستم عامل ها استفاده کرد در حالیکه LILO در واقع Boot Loader اصلی سیستم عامل لینوکس به حساب می آید.
- هم GRUB و هم LILO توانایی Boot کردن سیستم عامل از حافظه های خارجی را دارند اما GRUB توانایی بوت کردن سیستم عامل در شبکه را دارد در حالیکه LILO قابلیت Boot کردن سیستم عامل در شبکه را ندارد.

- زمانیکه فایل تنظیمات به مشکل می خورد، LILO بایستی مجددا نصب شود تا بتواند MBR را تعمیر کند اما تنظیمات پیشفرض GRUB آن را در حالت خط فرمان به اجرا در می آورد که بصورت دستی یا خودکار تنظیمات را اصلاح کنیم.
- GRUB دارای پیچیدگی های بیشتری نسبت به LILO است.
- LILO امروزه با GRUB جایگزین شده است و دیگر بصورت پیشفرض در لینوکس استفاده نمی شود.
- GRUB برای سیستم عامل های مختلف قابل استفاده است اما LILO برای سیستم عامل های محدودی قابل استفاده می باشد.
- مهمترین تفاوت بین لیلو و گراب در نحوه دریافت اطلاعات پیکربندی است. فایل پیکربندی لیلو با اجرای دستور lilo بصورت ثابت ذخیره می گردد. این اطلاعات یا در رکورد بوت اصلی (MBR) دیسک یا در رکورد بوت پارتیشن لینوکس ذخیره می شود. اطلاعات پیکربندی که توسط دستور lilo استفاده می شود، عموماً در فایل etc/lilo.conf ذخیره می شود. در زیر یک مثال از این فایل پیکربندی را مشاهده می نمایید.

```
boot=/dev/hda # boot loader to MBR
root=/dev/hda1 # root partition
install=/boot/boot.b
map=/boot/map
delay=50 # 5 second delay before auto-boot
image=/vmlinuz # kernel
label=linux # name to refer to entry
read-only
image=/vmlinuz.old # backup entry
label=old
read-only
```

در مثال بالا امکان انجام بوت از روی دو هسته مختلف یعنی vmlinuz و vmlinuz.old وجود دارد. هنگام بوت شما می توانید یکی از این هسته ها را انتخاب نمایید. با فشردن کلید TAB در اعلان لیلو، گزینه های موجود لیست می شوند. در صورتی که هسته جدیدی را با ویرایش فایل etc/lilo.conf اضافه کرده اید و یا هر تغییر دیگری در فایل پیکربندی لیلو اعمال کرده اید، باید پس از آن دستور lilo را اجرا نمایید تا این تغییرات اعمال شوند.



## تعمیر GRUB صدمه دیده

بسیاری از کاربران لینوکس به دلیل مشکلاتی مانند وابسته بودن به نرم افزارهای خاص یا ناآشنایی با سیستم لینوکس هنوز ویندوز را به همراه لینوکس بر روی ماشین خود نصب میکنند؛ از آنجایی که ویندوز سیستم عاملی ناپایدار است، هر چند وقت یکبار نیاز به نصب مجدد دارد، این نصب مجدد باعث می شود تا ویندوز MBR دیسک سخت شما را پاک کرده و اطلاعات خود را بر روی آن بنویسد و از آنجایی که ویندوز بوت لودری چند سیستم عامله ندارد، این دوباره نویسی MBR باعث پاک شدن اطلاعات مورد نیاز بوت لودر لینوکس شما می شود، در نتیجه زمان بوت سیستم دیگر به لینوکس دسترسی نخواهید داشت. در این قسمت قصد دارم بازیابی بوت لودر GRUB را شرح بدهم.

شما لینوکس خود را کنار ویندوز نصب می کنید، ولی پس از مدتی ویندوز شما دچار عیب شده و دیگر اجرا نمی شود، برای اجرا نشدن ویندوز دلایل بسیاری وجود دارد و دلایل بیشتری برای نصب مجدد ویندوز؛ حال شما ویندوز را مجدداً نصب می کنید، اما از آنجایی که ویندوز یک سیستم عامل منفرد بوده و بوت لودری ندارد که چند سیستم عامل را همزمان اجرا کند بنابراین اطلاعات موجود در سکتورهای بوت سیستم شما را رونویسی می کند. بعد از راه اندازی مجدد سیستم دیگر از بوت لودر لینوکس و گزینه های مربوط به لینوکس خبری نیست و سیستم ویندوز شما مستقیماً اجرا می شود. در چنین وضعیتی اغلب، کاربران تازه کار، لینوکس خود را مجدداً نصب می کنند که این کار علاوه بر اتلاف وقت بسیار، باعث از دست دادن مقدار زیادی اطلاعات و تنظیماتی می شود که شما در مدت کار با لینوکس ایجاد کرده بودید. بهترین راه برای تعمیر Bootloader، تعمیر گراب میباشد.

برای تعمیر سیستم کفایت بوت لودر گراب خود را از ابتدا نصب کنید، این کار بسیار ساده و در اندک زمانی ممکن است. برای تعمیر بوت لودر خود تنها به یک دیسک زنده ی لینوکس نیاز دارید.

پس ابزار کار ما یک دیسک زنده است فرقی ندارد از کدام توزیع استفاده میکنید مهم این است که سیستم از طریق یک دیسک زنده بوت شود. پس از بوت سیستم از طریق دیسک زنده یک پنجره ی خط فرمان (ترمینال) باز کرده و در آن مراحل زیر را طی کنید:

1. با وارد کردن دستور su به حالت کاربر ریشه یا root بروید.
2. اگر چند توزیع لینوکس نصب کرده اید باید بدانید که آخرین بار کدام یک از این توزیع ها، GRUB را نصب کرده است. برای نمونه اگر ابتدا Redhat را نصب کرده اید و بعد Debian را، پس GRUB موجود در سیستم شما را Debian نصب کرده است. خوب در این حالت باید بدانید Debian در کجای سیستم شما نصب شده است. اگر نمی دانید از طریق همان دیسک سخت تمام پارتیشن های خود را Mount کنید و سپس به جستجوی پارتیشنی بروید که دیان در آن نصب شده است. فرض کنیم دیان در پارتیشن sda6 نصب شده، پس sda6 را

mount می کنیم. البته اغلب دیسک های زنده تمام پارتیشن های موجود در هارد دیسک را به صورت پیش فرض در زمان بالا آمدن سیستم mount می کنند.

3. حالا با دستور chroot ریشه ی سیستم را تغییر می دهیم.

**#chroot /mnt/sda6**

4. مسیر /mnt/sda6 درواقع همان پارتیشنی است که سیستم لینوکس در آن نصب شده است. با استفاده از این روش شما پارتیشن / خود را از دیسک زنده به مسیری که وارد کرده اید تغییر داده اید.

5. حال باید گراب را اجرا کنید.

**#!/sbin/GRUB**

6. در مقابل شما صفحه ای قرار می گیرد همراه با یک اعلان (Prompt) که آماده ی دریافت دستورات است.

GNU GRUB version ۰.۹۵ (۶۴۰ / ۳۰۷۲K lower / ۳۰۷۲K upper memory) [ Minimal BASH like line editing is supported. For the first word, TAB lists possible command completions. Anywhere else TAB lists the possible completions of a device/filename. ] **GRUB>**

در این مرحله لازم است بدانید GRUB را در کجا نصب کرده اید؟ اگر برای سیستم خود پارتیشن boot ایجاد کرده اید،

پارتیشن گراب همان پارتیشن بوت است، اگر چنین پارتیشنی ایجاد نکرده بودید پارتیشن بوت شما همان پارتیشن لینوکس

است. برای مطمئن شدن می توانید در پارتیشن های خود boot/grub/ را جستجو کنید.

7. حالا محل نصب GRUB را به برنامه معرفی کنید:

**>root (hd0)**

**نکته مهم:** گراب برای نمایش پارتیشن های دیسک سخت از مدل یونیکس استفاده می کند، در مدل یونیکسی hda همان hd0 می شود و hda6 معادل hd5 است در واقع GRUB علی رغم استاندارد لینوکس پارتیشن ها را از صفر آغاز می کند. گراب نام درایوها را در فایل boot/grub/device.map به صورت جدولی که شامل نام گرابی برای مثال (hd0) و نام لینوکسی برای مثال (/dev/hda)، لیست میکند.

**پارتیشن ها در لینوکس :** hda1 , hda2 , hdb1 , hdb2 و یا sda1 , sda2 , sdb1 , sdb2

**پارتیشن ها در GRUB :** hd0,0 – hd0,1 – hd0,2 – hd0,3 – hd0,4

8. می توانیم برای اینکه بفهمیم /boot در کدام درایو و پارتیشن قرار دارد از دستور df /boot استفاده کنیم.

**#df /boot/**

9. مرحله ی آخر دستور نصب است، برای این کار تایپ کنید:

**>setup (hd0)**

**>quit**

گراب شما نصب شده است، سیستم را reboot کنید، در این بارگذاری سیستم خواهید دید که GRUB شما نصب شده است و می توانید بدون مشکل از کار با سیستم لینوکس لذت ببرید.

**توجه:** در برخی دیسک های زنده جدیدتر دیگر نیازی به اجرای مرحله chroot کردن نیست و شما می توانید مستقیماً با اجرای دستور Grub و انتخاب root به نصب بوت لودر گراب پردازید. بعد از boot سیستم حتماً یک پشتیبان از فایل گراب تهیه کنید تا در صورت بروز مشکل بتوانید از آن استفاده کرده و تنظیمات گراب را به حالت قبل بازگردانید. در لینوکس های Debian Base :

```
#cp /etc/default/grub /home/skywan13/grub.backup
```

و در توزیع های Redhat Base :

```
#cp /etc/grub.conf /home/skywan13/grub.backup
```

## ویرایش فایل grub در توزیع های مبتنی بر Debian

برای ویرایش فایل پیکربندی گراب، می بایست از نرم افزارهای ویرایشگر متن استفاده نمود. به صورت پیش فرض ویرایشگر پر قدرت vi در اکثر توزیع های لینوکس نصب می باشد که می توان از آن استفاده نمود. همچنین برای ویرایش فایل گراب، می بایست دسترسی root داشته باشید. جهت تغییر در پیکربندی فایل گراب دستور زیر را در ترمینال وارد کنید.

```
# vi /etc/default/grub
```

فایل گراب به صورت پیش فرض، در اکثر توزیع های مبتنی بر Debian، دارای محتویات زیر می باشد.

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'
GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```



```
# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true
# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"
# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

## تغییر سیستم عامل پیش فرض در Grub

چنانچه پیش از یک سیستم عامل بر روی کامپیوتر خود نصب کرده اید، می توانید با تغییر محتویات فایل گراب سیستم عاملی را که به طور پیش فرض بعد از روشن شدن سیستم راه اندازی می شود را تغییر دهید. بدین منظور در خط ششم محتویات گراب (`GRUB_DEFAULT=0`) مقدار صفر را می بایست تغییر دهید. عدد صفر بیانگر «اولین گزینه ی منو از بالا» است. این نکته را در نظر بگیرید که در گراب شمارش از صفر شروع می شود.

### GRUB\_DEFAULT=0

چنانچه می خواهید یک سیستم عامل مشخص (و نه متغیر) هنگام بوت شدن رایانه به صورت پیش فرض بالا بیاید، می بایستی که عدد صفر را تغییر دهید. چنانچه به طور مثال هنگام بوت شدن رایانه، سیستم عامل مورد نظر شما در گزینه ی سوم قرار دارد، عدد «صفر» را می بایست به «دو» تغییر داد.

و چنانچه می خواهید آخرین سیستم عاملی که وارد آن شده اید، پیش فرض منوی گراب برای ورود بعدی به صورت اتوماتیک باشد. می بایستی که عدد صفر را با کلمه saved جایگزین کنید:

### GRUB\_DEFAULT=saved

**اخطار:** اگر از dmraid استفاده می کنید، پارامتر default را به کلیدواژه 'saved' تغییر ندهید. در صورت انجام این کار قادر به بوت کردن سیستم خود نخواهید بود.

## تغییر مدت زمان نمایش Grub در هنگام بوت

برای تغییر این مدت زمان می‌بایست محتویات این خط (`GRUB_TIMEOUT=10`) را به مقدار دلخواه تغییر دهید. این زمان بر اساس ثانیه است. چنانچه این مقدار را برابر صفر قرار دهید، در هنگام بوت، منوی گراب نشان داده نخواهد شد ولی می‌توان هنگام بوت با فشردن دکمه‌ی shift آن را دید.

**GRUB\_TIMEOUT=10**

## افزایش زمان نمایش منوی Grub

اگر شما نیاز داشته باشید تا برای تغییر گزینه بوت یا انتخاب یک هسته متفاوت به داخل منوی گراب وارد شود، باید بلافاصله بعد از اجرای گراب در هنگام بوت سیستم دکمه‌ی `ESC` را فشار دهید. به طور پیش‌فرض شما باید دکمه‌ی `ESC` را خیلی سریع فشار دهید. برای افزایش این زمان شما باید فایل گراب را ویرایش کرده و زمان وارد شده در قسمت `timeout` را که برحسب ثانیه است را تغییر دهید.

**GRUB\_HIDDEN\_TIMEOUT=0**

همچنین شما می‌توانید این منو را همیشه در هنگام اجرای گراب داشته باشید. برای این کار، با برداشتن `#` از ابتدای `'hiddenmenu'` آن را از حالت توضیحات خارج کنید. ممکن است این گزینه وجود نداشته باشد لذا می‌توانید خودتان آن را اضافه کنید.

### hiddenmenu

اوبونتو از یک ابزار به نام `update-grub` برای ایجاد تغییرات در گراب استفاده می‌کند. `update-grub`، به طور خودکار تمام هسته‌هایی را که شما در شاخه `boot` دارید شناسایی کرده و بر روی هر کدام تنظیمات مختلف `global` را اعمال می‌کند. هر وقت که شما هسته لینوکس را با استفاده از مخازن به روز می‌کنید یا تغییری در فایل پیکربندی آن اعمال می‌کنید برای تصحیح تنظیمات گراب حتماً از این دستور استفاده کنید، در غیر این صورت تغییرات اعمال نخواهند شد. در خط اول فایل کانفیگ گراب این نکته تذکر داده شده است.

### #update-grub

توجه داشته باشید که شما می‌توانید با فشار دادن `'e'` در هنگام نمایش منوی گراب و ویرایش خط هسته، پارامترهای هسته را به صورت موقت تغییر بدهید. همانطور که می‌دانید این تغییرات فقط مختص عملیات بوت جاری می‌باشد و در راه‌اندازی‌های بعد اعمال نمی‌گردد.

## پاک کردن Grub و نصب Bootloader ویندوز

اگر به هر دلیلی دیگه نمی‌خواهید که از گراب و لینوکس استفاده کنید و می‌خواهید گراب را پاک کرده و از همان روش معمولی بوت شدن ویندوز استفاده کنید، سیستم رو با DVD ویندوز بوت کنید و وقتی به مرحله شروع نصب رسید، به جای اینکه فرملن شروع نصب را بدهید، دکمه R را بزنید تا وارد Recovery بشوید. سپس از دستور زیر را اجرا کنید و به پیغام های هشدار، جواب مثبت بدهید تا بوت لودر ویندوز جایگزین گراب بشه :

**fixmbr**

## نرم افزار کمکی Super Grub Disk

خب تاکنون با مدیریت گراب و فرامین آن آشنا شدید اما برای مدیریت و تعمیر گراب در صورت داشتن چند سیستم عامل یک راهکار فوق سریع نیز وجود دارد آنهم استفاده از یک دیسک کم حجم بنام SuperGrub است که از لینک زیر می‌توانید آنرا دانلود و فایل ISO آنرا رایت و جهت تعمیر گراب، سیستم را با آن بوت کنید.

[ftp://linux.iingen.unam.mx/Linux/SuperGrubDisk/super\\_grub\\_disk\\_0.9797.iso](ftp://linux.iingen.unam.mx/Linux/SuperGrubDisk/super_grub_disk_0.9797.iso)

Linux Cookie in Persian v.2

فصل سوم

## راه اندازی FTP در لینوکس



**FTP SERVER**

## آشنائی با پروتکل FTP

مقدمه:

امروزه از پروتکل های متعددی در شبکه های کامپیوتری استفاده می گردد که صرفاً "تعداد اندکی از آنان به منظور انتقال داده طراحی و پیاده سازی شده اند. اینترنت نیز به عنوان یک شبکه گسترده از این قاعده مستثنی نبوده و در این رابطه از پروتکل های متعددی استفاده می شود. برای بسیاری از کاربران اینترنت همه چیز محدود به وب و پروتکل مرتبط با آن یعنی HTTP است، در صورتی که در این عرصه از پروتکل های متعدد دیگری نیز استفاده می گردد. **File Transfer Protocol** نمونه ای در این زمینه است.

### پروتکل FTP چیست؟

تصویر اولیه اینترنت در ذهن بسیاری از کاربران، استفاده از منابع اطلاعاتی و حرکت از سایتی به سایت دیگر است و شاید به همین دلیل باشد که اینترنت در طی سالیان اخیر به سرعت رشد کرده و متداول شده است. بسیاری از کارشناسان این عرصه اعتقاد دارند که اینترنت گسترش و عمومیت خود را مدیون سرویس وب می باشد. فرض کنید که سرویس وب را از اینترنت حذف نمائیم. برای بسیاری از ما این سوال مطرح خواهد شد که چه نوع استفاده ای را می توانیم از اینترنت داشته باشیم؟ در صورت تحقق چنین شرایطی، یکی از عملیاتی که کاربران قادر به انجام آن خواهند بود دریافت داده، فایل های صوتی، تصویری و سایر نمونه فایل های دیگر با استفاده از پروتکل FTP است.

### ویژگی های پروتکل FTP

پروتکل FTP، اولین تلاش انجام شده برای ایجاد یک استاندارد به منظور مبادله فایل بر روی شبکه های مبتنی بر پروتکل TCP/IP است که از اوایل سال 1970 مطرح و مشخصات استاندارد آن طی RFC 959 در اکتبر سال 1985 ارائه گردید. پروتکل FTP دارای حداکثر انعطاف لازم و امکان پذیر به منظور استفاده در شبکه های مختلف با توجه به نوع پروتکل شبکه می باشد.

پروتکل FTP از مدل کلاینت - سرور تبعیت می نماید. برخلاف HTTP که یک حاکم مطلق در عرصه مرورگرهای وب و سرویس دهندگان وب است، نمی توان ادعای مشابهی را در رابطه با پروتکل FTP داشت چون هم اینک مجموعه ای گسترده از سرویس گیرندگان و سرویس دهندگان FTP وجود دارد.

برای ارسال فایل با استفاده از پروتکل FTP به یک سرویس گیرنده FTP (در سمت کلاینت) نیاز است. ویندوز دارای یک برنامه سرویس گیرنده FTP از قبل تعبیه شده بوده ولی دارای محدودیت های مختص به خود می باشد. در این رابطه نرم

افزارهای متعددی تاکنون طراحی و پیاده سازی شده است که **Smart FTP**، **FTP Explorer**، **WSFTP Professional** و نمونه هائی در این زمینه می باشند .

پروتکل FTP را می توان به عنوان یک سیستم پرس وجو نیز تلقی نمود چراکه سرویس گیرندگان و سرویس دهندگان گفتگوی لازم به منظور تأیید یکدیگر و ارسال فایل را انجام می دهند. علاوه بر این، پروتکل فوق مشخص می نماید که سرویس گیرنده و سرویس دهنده، داده را بر روی کانال گفتگو ارسال نمی کنند. در مقابل، سرویس گیرنده و سرویس دهنده در خصوص نحوه ارسال فایل ها بر روی اتصالات مجزا و جداگانه (یک اتصال برای هر ارسال داده) با یکدیگر گفتگو خواهند کرد (نمایش لیست فایل های موجود در یک دایرکتوری نیز به عنوان یک ارسال فایل تلقی می گردد). پروتکل FTP امکان استفاده از سیستم فایل را مشابه پسته یونیکس و یا خط دستور ویندوز در اختیار کاربران قرار می دهد. سرویس گیرنده در ابتدا یک پیام را برای سرویس دهنده ارسال و سرویس دهنده نیز به آن پاسخ خواهد داد و در ادامه ارتباط غیرفعال می گردد. وضعیت فوق با سایر پروتکل هائی که به صورت تراکنشی کار می کنند متفاوت می باشد (نظیر پروتکل HTTP). برنامه های سرویس گیرنده زمانی قادر به شبیه سازی یک محیط تراکنشی می باشند که از مسائلی که قرار است در آینده محقق شوند، آگاهی داشته باشند. در واقع، پروتکل FTP یک دنباله statefull از یک و یا چندین تراکنش است.

سرویس گیرندگان، مسئولیت ایجاد و مقداردهی اولیه درخواست ها را برعهده دارند که با استفاده از دستورات اولیه FTP انجام می گردد. دستورات فوق، عموماً "سه و یا چهار حرفی می باشند (مثلاً" برای تغییر دایرکتوری از دستور `cwd` استفاده می شود). سرویس دهنده نیز بر اساس یک فرمت استاندارد به سرویس گیرندگان پاسخ خواهد داد (سه رقم که به دنبال آن از `space` استفاده شده است به همراه یک متن تشریحی). سرویس گیرندگان می بایست صرفاً" به کد عددی نتیجه استناد نمایند چراکه متن تشریحی تغییر پذیر بوده و در عمل برای اشکال زدائی مفید است (برای کاربران حرفه ای). پروتکل FTP دارای امکانات حمایتی لازم برای ارسال داده با نوع های مختلف می باشد. دو فرمت متداول، اسکی برای متن (سرویس گیرنده با ارسال دستور `TYPE A`، موضوع را به اطلاع سرویس دهنده می رساند) و `image` برای داده های باینری است (توسط `TYPE I` مشخص می گردد). ارسال داده با فرمت اسکی در مواردی که ماشین سرویس دهنده و ماشین سرویس گیرنده از استانداردهای متفاوتی برای متن استفاده می نمایند، مفید بوده و یک سرویس گیرنده می تواند پس از دریافت داده آن را به فرمت مورد نظر خود ترجمه و استفاده نماید. مثلاً" در نسخه های ویندوز از یک دنباله `carriage return` و `linefeed` برای نشان دادن انتهای خط استفاده می گردد در صورتی که در سیستم های مبتنی بر یونیکس صرفاً" از یک `linefeed` استفاده می شود. برای ارسال هر نوع داده که به ترجمه نیاز نداشته باشد، می توان از ارسال باینری استفاده نمود.

اتخاذ تصمیم در رابطه با نوع ارسال فایل ها در اختیار سرویس گیرنده است (برخلاف HTTP که می تواند به سرویس گیرنده نوع داده ارسالی را اطلاع دهد). معمولاً" سرویس گیرندگان ارسال باینری را انتخاب می نمایند و پس از دریافت

فایل ، ترجمه لازم را انجام خواهند داد . ارسال باینری ذاتاً " دارای کارآئی بیشتری است چراکه سرویس دهنده و سرویس گیرنده نیازی به انجام تراکنش های *on the fly* نخواهند داشت . ارسال اسکی گزینه پیش فرض انتخابی توسط پروتکل FTP است و در صورت نیاز به ارسال باینری ، سرویس گیرنده می بایست این موضوع را از سرویس دهنده درخواست نماید یک اتصال پروتکل TCP/IP ( نسخه شماره چهار ) شامل دو نقطه مجزا می باشد که هر نقطه از یک آدرس IP و یک شماره پورت استفاده می نماید .

برقراری ارتباط بین یک سرویس گیرنده و یک سرویس دهنده منوط به وجود چهار عنصر اطلاعاتی است :

آدرس سرویس دهنده، پورت سرویس دهنده، آدرس سرویس گیرنده و پورت سرویس گیرنده . در زمان برقراری یک ارتباط ، سرویس گیرنده از یک شماره پورت استفاده می نماید . این شماره پورت می تواند متناسب با نوع عملکرد برنامه سرویس گیرنده به صورت اختیاری و یا اجباری باشد . مثلاً " برخی برنامه های سرویس گیرنده به منظور ارتباط با سرویس دهنده ، نیازمند استفاده از یک شماره پورت خاص می باشند ( نظیر برنامه های سرویس گیرنده وب و یا مرورگرهای وب که از پورت شماره 80 به منظور ارتباط با سرویس دهنده وب استفاده می نمایند) . در مواردی که الزامی در خصوص شماره پورت وجود ندارد از یک شماره پورت موقتی و یا *ephemeral* استفاده می گردد. این نوع پورت ها موقتی بوده و توسط IP stack ماشین مربوطه به متقاضیان نسبت داده شده و پس از خاتمه ارتباط ، پورت آزاد می گردد . با توجه به این که اکثراً IP Stacks بلافاصله از پورت موقت آزاد شده استفاده نخواهند کرد ( تا زمانی که تمام *pool* تکمیل نشده باشد ) ، در صورتی که سرویس گیرنده مجدداً درخواست برقراری یک ارتباط را نماید ، یک شماره پورت موقتی دیگر به وی تخصیص داده می شود .

پروتکل FTP منحصرراً از پروتکل TCP استفاده می نماید( هرگز از پروتکل UDP استفاده نمی شود) . معمولاً " پروتکل های لایه Application ( با توجه به مدل مرجع OSI ) از یکی از پروتکل های TCP و یا UDP استفاده می نمایند ( به جزء پروتکل DNS ) . پروتکل FTP نیز از برخی جهات شرایط خاص خود را دارد و برای انجام وظایف محوله از دو پورت استفاده می نماید . این پروتکل معمولاً " از پورت شماره 20 برای ارسال داده و از پورت 21 برای گوش دادن به فرامین / استفاده می نماید . توجه داشته باشید که برای ارسال داده همواره از پورت 20 استفاده نمی گردد و ممکن است در برخی موارد از پورت های دیگر استفاده شود .

اکثر سرویس دهندگان FTP از روش خاصی برای رمزنگاری اطلاعات استفاده نمی نمایند و در زمان *login* سرویس گیرنده به سرویس دهنده ، اطلاعات مربوط به نام و رمز عبور کاربر به صورت متن معمولی در شبکه ارسال می گردد . افرادی که دارای یک *Packet sniffer* بین سرویس گیرنده و سرویس دهنده باشند ، می توانند به سادگی اقدام به سرقت نام و رمز عبور نمایند . علاوه بر سرقت رمزهای عبور ، مهاجمان می توانند تمامی مکالمات بر روی اتصالات FTP را شنود و محتویات داده های ارسالی را مشاهده نمایند . پیشنهادات متعددی به منظور ایمن سازی سرویس دهنده FTP مطرح می گردد



ولی تا زمانی که رمزنگاری و امکانات حفاظتی در سطح لایه پروتکل IP اعمال نگردد (مثلاً رمزنگاری توسط IPsec) نمی بایست از FTP استفاده گردد خصوصاً اگر بر روی شبکه اطلاعات مهم و حیاتی ارسال و یا دریافت می گردد. همانند بسیاری از پروتکل های لایه Application، پروتکل FTP دارای کدهای وضعیت خطاء مختص به خود می باشد (همانند HTTP) که اطلاعات لازم در خصوص وضعیت ارتباط ایجاد شده و یا درخواستی را ارائه می نماید. زمانی که یک درخواست (GET, PUT) برای یک سرویس دهنده FTP ارسال می گردد، سرویس دهنده پاسخ خود را به صورت یک رشته اعلام می نماید. اولین خط این رشته معمولاً شامل نام سرویس دهنده و نسخه نرم افزار FTP است. در ادامه می توان دستورات GET و یا PUT را برای سرویس دهنده ارسال نمود. سرویس دهنده با ارائه یک پیام وضعیت به درخواست سرویس گیرندگان پاسخ می دهد.

FTP، یک پروتکل ارسال فایل است که با استفاده از آن سرویس گیرندگان می توانند به سرویس دهندگان متصل و صرفنظر از نوع سرویس دهنده اقدام به دریافت و یا ارسال فایل نمایند. پروتکل FTP به منظور ارائه خدمات خود از دو حالت متفاوت استفاده می نماید: Active Mode و Passive Mode. مهمترین تفاوت بین روش های فوق جایگاه سرویس دهنده و یا سرویس گیرنده در ایجاد و خاتمه یک ارتباط است.

## مدهای کاری سرویس دهنده FTP

### ActiveMode

Active Mode، روش سنتی ارتباط بین یک سرویس گیرنده FTP و یک سرویس دهنده می باشد که عملکرد آن بر اساس فرآیند زیر است:

تمام FTP های دنیا در دو mode کار می کنند. یا Active هستند یا Passive. که البته به طور پیش فرض FTP ها در حالت اکتیو کار می کنند.

کلاینتیک یک پورت رندوم باز کرده و از طریق آن یک ارتباط با پورت 21 سرویس دهنده FTP برقرار می نماید و روی پورت 21 احراض هویت می شود. پورت 21، پورتهی است که سرور به آن گوش فرا می دهد تا از صدور فرامین آگاه و آنان را به ترتیب پاسخ دهد. کلاینت برای برقراری ارتباط با سرور از یک پورت تصادفی و موقتی (بزرگتر از 1024) استفاده می نماید (پورت x).

کلاینت شماره پورت لازم برای ارتباط سرویس دهنده با خود را از طریق صدور دستور PORT N+1 به وی اطلاع می دهد (پورت x+1)

سرور یک ارتباط را از طریق پورت 20 خود با پورت مشخص شده کلاینت (پورت x+1) برقرار می نماید. در فرآیند فوق، ارتباط توسط کلاینت آغاز و پاسخ به آن توسط سرور و از طریق پورت x+1 که توسط کلاینت مشخص شده است، انجام می شود. در صورتی که کلاینت از سیستم ها و دستگاه های امنیتی خاصی نظیر فایروال استفاده کرده باشد، می بایست تهدیدات لازم به منظور ارتباط کامپیوترهای میزبان راه دور به کلاینت پیش بینی تا آنان بتوانند به هر پورت بالاتر از 1024 کلاینت دستیابی داشته باشند. بدین منظور لازم است که پورت های اشاره شده بر روی ماشین کلاینت open باشند. این موضوع می تواند تهدیدات و چالش های امنیتی متعددی را برای سرویس گیرندگان به دنبال داشته باشد.

### Passive Mode

در Passive Mode، که به آن "مدیریت و یا اداره سرویس گیرندگان FTP" نیز گفته می شود از فرآیند زیر استفاده میگردد:

کلاینت دو پورت را فعال می نماید (پورت x و x+1)، ارتباط اولیه از طریق پورت x کلاینت با پورت 21 سرور آغاز می گردد کلاینت از این پورت احراض هویت انجام میدهد. سرور یک پورت را فعال (y) و به کلاینت شماره پورت را اعلام می نماید.

در ادامه کلاینت یک اتصال از طریق پورت  $x+1$  با پورت  $x$  سرور برقرار می نماید. در فرآیند فوق ، کلاینت دارای نقش محوری است و فایروال موجود بر روی کلاینت می تواند درخواست های دریافتی غیرمجاز به پورت های بالاتر از 1024 را به منظور افزایش امنیت بلاک نمایند . در صورتی که بر روی کامپیوترهای سرور نیز فایروال نصب شده باشد ، می بایست پیکربندی لازم به منظور استفاده از پورت های بالاتر از 1024 بر روی آن انجام گردد . باز نمودن پورت های فوق بر روی سرور می تواند چالش های امنیتی خاصی را به دنبال داشته باشد. متأسفانه تمامی کلاینتهای FTP از Passive Mode حمایت نمی کنند. اگر یک کلاینت بتواند به یک سرور login کند ولی قادر به ارسال داده بر روی آن نباشد ، نشان دهنده این موضوع است که فایروال و یا Gateway برای استفاده از PassiveMode به درستی پیکربندی نشده است .

### مقایسه بین اکتیو و پسیو به زبان ساده :

در حالت Active باید فایروال کلاینتها پیکربندی شود ولی در حالت Passive فایروال سمت سرور کانفیگ میشود. مشکل Active در همین است که باید فایروال سمت کلاینت توسط خود کاربر پیکربندی شود که این می تواند خود مشکل آفرین باشد. اما در حالت Passive تنظیمات فایروال توسط مدیر شبکه انجام می شود بنابراین لازم است به سرور اجازه داده شود که به اتصالات هر پورت بالاتر از 1024 پاسخ دهد . ترافیک فوق ، معمولاً "توسط فایروال سرور بلاک می گردد . در چنین شرایطی امکان استفاده از Passive Mode وجود نخواهد داشت .

حالت پسیو دارای سرعت بالاتر و overhead کمتری است و نداشتن مشکلات فایروالی هم جزو محسنات این مد به حساب می آید . با توجه به مستندات درج شده در RFC 1579 ، استفاده از Passive Mode به دلایل متعددی به Active Mode ترجیح داده می شود . تعداد سرویس دهندگان موجود بر روی اینترنت به مراتب کمتر از سرویس گیرندگان می باشد . با استفاده از امکانات موجود می توان سرویس دهندگان را پیکربندی تا بتوانند از مجموعه پورت های محدود و تعریف شده ای با در نظر گرفتن مسائل امنیتی ، استفاده نمایند .

معروف ترین و بهترین این سرویس دهنده ها از نظر ردهت VSFTP می باشد که در این فصل به طور اجمالی مورد بررسی قرار می گیرد.

## نصب راه اندازی VSFTP

معروفترین در بین سرویس دهندگان خدمات فایل VSFTP و ProFTP می باشد که VSFTP از امنیت بالاتری برخوردار بوده و به شدت Stable می باشد و از مهمترین قابلیت های آن می توان به Multi Homing بودن آن اشاره کرد. این قابلیت اجازه می دهد چندین Ftp Daemon روی یک سرور اجرا شود که هر کدام از اینها دارای تنظیمات مختص به خود می باشد. مبنای احراز هویت VSFTP فایل `/etc/passwd` است. یعنی به صورت پیش فرض بانک جداگانه ای برای یوزرها ندارد بلکه از یوزرهای Local پشتیبانی میکند. تنها عیب VSFTP راه اندازی مشکل و سرعت پائین آن به نسبت ProFTP است. VSFTP به صورت پیش فرض به صورت Standalone بالا می آید ولی قابلیت این را دارد که تحت نظر Xinetd ارائه سرویس کند. اکثر Hosting ها از ProFTP استفاده می کنند چون سرعت احراز هویت بالاتری دارد و علت آن هم نداشتن ماژول امنیتی بر روی آن است که باعث می شود به راحتی مورد حمله هکرها قرار گیرد. مهمترین مزیت آن فقط سرعت بالای احراز هویت و اتصال آن می باشد و البته بسیاری از فیچرهای VSFTP را هم ندارد. ProFTP به صورت پیش فرض از فایل `/etc/passwd` استفاده نمی کند بلکه برای احراز هویت یوزرها از فایل جداگانه ای بهره می برد.

جهت نصب این سرویس دهنده میتوان از سرویس Yum و یا اگر پکیج آن از قبل موجود باشد می توان از دستور rpm استفاده کرد. ولی توصیه می شود در صورت امکان از Yum برای نصب نرم افزارها استفاده کنید زیرا نیازمندی های لازم را دانلود و نصب می کند. این قابلیت در rpm وجود ندارد لذا جهت نصب از دستورات زیر استفاده می کنیم . ابتدا باید از نصب بودن پکیج vsftp اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

```
#rpm -qa | grep vsftp
```

- در صورت نصب نبودن ، در سیستم های ردهت جهت نصب vsftpd از yum استفاده می کنیم :

```
#yum -y install vsftp
```

- بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج vsftp بر روی سیستم نصب شده است یا خیر

```
#rpm -qa | grep vsftp
```

- سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm -ql vsftpd
```

- و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm -qi vsftpd
```

- سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
#chkconfig vsftpd on
```

```
#service vsftpd restart
```

و در انتها سرویس را reset می کنیم :

## مهمترین مسیرهای ایجاد شده توسط VSFTP

با نصب این سرویس چندین شاخه و مسیر جدید به سیستم اضافه می شود که مهمترین آنها در زیر مختصراً توضیح داده شده است :

**/etc/vsftpd/**  
**/etc/logrotate.d/vsftpd.log**  
**/etc/pam.d/vsftpd**  
**/etc/rc.d/init.d/vsftpd**  
**/var/ftp/pub/**  
**/usr/sbin/vsftpd/**

[:/etc/vsftpd/](#)

در زیر این دایرکتوری چهار فایل مهم پیکربندی این سرویس قرار دارد که به ترتیب شرح داده می شود :

**Vsftpd.conf**  
**User\_list**  
**Ftpusers**  
**Vsftpd\_conf\_migrate.sh**

- **vsftpd.conf**: این فایل اصلی ترین فایل پیکربندی سرویس vsftpd می باشد.
- **user\_list**: این فایل شامل لیست یوزرهایی است که به آنها دسترسی یا عدم دسترسی به ftp داده می شود به شرط آنکه گزینه **userlist\_deny** ، در فایل پیکربندی اصلی را مقدار دهی کرده باشیم.
- **ftpusers**: هر یوزری که نام آن در این فایل قرار بگیرد به آن یوزر اجازه login به ftp داده نمی شود. در اصل این فایل یک blacklist می باشد.
- **vsftpd\_conf\_migrate.sh**: برای migrate کردن و جابه جایی بین دو ftp از این اسکریپت استفاده میشود.

و در ادامه :

[:/etc/logrotate.d/vsftpd.log](#)

فایل کانفیگ rotate لاگ این سرویس در این آدرس قرار دارد.

[:/etc/pam.d/vsftpd](#)

Pam یک مکانیزم امنیتی است که برای کنترل سرویس ها به کار می رود. اگر بخواهیم مکانیزم احراز هویت vsftpd در اختیار pam باشد باید در این فایل تنظیمات لازم را اعمال کنیم.

### [:/etc/rc.d.init.d/vsftpd](#)

اسکرپت اجرای سرویس در این مکان قرار دارد . vsftpd به صورت standalone اجرا میشود و در زیر مجموعه init قرار دارد.

### [:/var/ftp/pub/](#)

این مسیر برای قرار دادن فایل ، و دایرکتوری جاری یوزرهایی که لاگین می کنند به کار می رود. به این مسیر ftproot گفته می شود. تمام یوزرهای anonymous به طور پیش فرض وارد این دایرکتوری می شوند و کاربران local هم بعد از ورود به ftp به دایرکتوری home مربوطه هدایت خواهند شد مگر اینکه ما این مسیر را تغییر دهیم .

### [:/usr/sbin/vsftpd](#)

فایل دستور vsftpd در این مسیر قرار دارد .

## مروری بر تنظیمات فایل کانفیگ اصلی VSFTP

Syntax این فایل بدین گونه است که هر چیزی که درون آن نوشته می شود باید بدون فاصله باشد. کلا برای اتصال به vsftpd دو نوع یوزر داریم. این یوزرها یا یوزر local سیستم هستند یا یوزرهای anonymous. کانفیگ کلی vsftpd بدون آپشن خاصی 13 خط می باشد که در زیر مهمترین آنها توضیح داده شده است. برای کسب اطلاعات تکمیلی به man vsftpd.conf مراجعه شود. در این فصل کاربران anonymous به اختصار anon نامیده می شوند.

#vi /etc/vsftpd/vsftpd.com

**anonymous\_enable=YES**

Yes بودن این گزینه به کاربران anonymous اجازه می دهد که از طریق محیط گرافیکی بدون پسورد وارد دایرکتوری pub سرور FTP شوند. یوزرهای anon اجازه chroot را ندارند.

**anon\_root=/opt/dir\_anon**

اگر بخواهیم کاربران anon به محض ورود به FTP به دایرکتوری مشخصی هدایت شوند در جلوی این گزینه مسیر مورد نظر را وارد می کنیم.

**anon\_upload\_enable=YES**

Yes بودن این آپشن اجازه می دهد یوزرهای anon بتوانند در FTP فایل آپلود کنند.

**anon\_mkdir\_enable=YES**

با yes قرار دادن مقدار این خط یوزرهای anon می توانند در FTP دایرکتوری ایجاد کنند.

**anon\_max\_rate=4000**

مقدار این خط نرخ حداکثر سرعت دانلود و آپلود یوزرهای anon را مشخص می کند. این نرخ بر اساس بایت می باشد.

**no\_anon\_passwd=YES**

با فعال کردن این خط، برنامه FTP در محیط **cli** از یوزرهای anon پسورد نمی خواهد.

**anon\_umask=???**

مجوزهای پیش فرض ایجاد فایل و دایرکتوری، برای کاربران anonymous را تعیین می کند.

**anon\_other\_write\_enable = NO**

اگر این خط برابر YES باشد کاربران anon به غیر از آپلود و ایجاد دایرکتوری مجاز به انجام عملیات نوشتن حذف و تغییر نام خواهند شد. به طور کلی این کار توصیه نمی شود اما برای تکمیل مطالب این بخش آورده شده است.

**local\_enable=YES**

با فعال کردن این گزینه یوزرهای local ای که داخل فایل passwd هستند این اجازه را پیدا می کنند با یوزر و پسورد خود به FTP لاگین کرده و وارد دایرکتوری خانگی خودشان بشوند. یوزرهای local اجازه changroot را دارند که این برای سیستم یک خطر امنیتی محسوب شده و باید این قابلیت را غیرفعال کرد.

**write\_enable=YES**

این گزینه به یوزرهای local اجازه آپلود فایل در FTP را می دهد. این آپشن زمانی کاربرد دارد که فایل را تحت FTP بسازیم نه تحت bash.

**local\_umask=022**

این umask پرمیژن پیش فرضی است که برای فایل های آپلود شده توسط کاربران local استفاده می شود که پیش فرض 775 می باشد.

**local\_max\_rate=10000**

مقدار این خط نرخ حداکثر سرعت دانلود و آپلود یوزرهای local را مشخص می کند. این نرخ بر اساس بایت می باشد.

**local\_root=/var/tmp**

اگر بخواهیم کاربران local به محض ورود به FTP به دایرکتوری مشخصی هدایت شوند در جلوی این گزینه مسیر مورد نظر را وارد می کنیم.

**dirmessage\_enable=YES**

این گزینه فقط مختص یوزرهایی است که تحت cli به FTP لاگین می کنند. ما می توانیم برای هر دایرکتوری یک message ایجاد کنیم تا به محض ورود کاربر به آن دایرکتوری پیام مورد نظر برای کاربر به نمایش داده شود. برای ایجاد پیام به داخل دایرکتوری مورد نظر رفته و یک فایل به نام **message** ایجاد می کنیم و پیام مورد نظرمان را درون آن ذخیره می کنیم. با yes بودن این خط ، vsftpd به محض ورود کاربر به دایرکتوری از قبل مشخص شده ، ابتدا انجا را چک می کند تا ببیند آیا چنین فایل وجود دارد یا خیر . در موجود بودن پیام داخل آن را برای کاربر نمایش می دهد.



**xferlog\_enable=YES**

لاگ های vsftpd به دو صورت ذخیره می شوند. یا با فایل vsftpd.log درون خود vsftpd ذخیره می شود که باید این گزینه NO باشد. یا با YES قرار دادن این خط کاری می کنیم log آن توسط log سرور در دایرکتوری مربوطه ذخیره شود.

**listen=YES**

اگر این گزینه yes باشد FTP در مد standalone و تحت نظر init کار می کند. اگر بخواهیم این سرویس تحت نظر xinetd اداره شود باید این گزینه را برابر no قرار داده و در زیر xinetd یک فایل کانفیگ به نام vsftpd بسازیم. در انتهای این فصل نمونه ای از این فایل آورده شده است.

**max\_per\_ip=20**

این خط مشخص می کند چه تعداد کانکشن می تواند از هر IP به سرور متصل شود.

**max\_client=100**

این خط بیان می کند در یک زمان حداکثر چند یوزر می توانند هم زمان به FTP متصل شوند. (عدد 100 مثال است)

**chown\_uploads=YES****chown\_username=نام یوزر**

اگر بخواهیم مالک (owner) فایل های که یوزرهای anon آپلود می کنند یوزر دیگری باشد تا خاصیت اجرا را از یوزر anon بگیریم خط اول را برابر yes قرار داده و در خط دوم نام یوزری که می خواهیم owner فایل ها باشد را وارد می کنیم. مثل یوزری که قابلیت لاگین را از آن گرفته باشیم.

**ftp\_banner=welcome to FTP****banner\_file=/opt/ftp/ftp.txt**

هر متنی که در جلوی عبارت خط اول نوشته شود برای یوزرهایی که با محیط cli لاگین کرده اند نمایش درمی آید. باید دقت داشت در جلوی این عبارت نمی توان بیش از یک خط نوشت. اگر متن ما بیش از چند خط بود باید آن را در یک فایل جداگانه نوشته و آدرس آن را در جلوی خط دوم وارد کرد.

**pam\_service\_name=VSFTPD****tcp\_wrappers=YES**

این دو خط مکانیزم های امنیتی کنترل سرویس FTP را مشخص می کند. اگر بخواهیم رنجی از IP را Block کنیم تا به FTP دسترسی نداشته باشند باید از مکانیزم امنیتی tcp\_wrappers استفاده شود. البته باید از قبل IP های مجاز و غیر مجاز را درون فایل های /etc/hosts.allow و /etc/hosts.deny وارد کنیم .

**userlist\_enable=YES**

**userlist\_file=/etc/vsftpd/user\_list**

اگر مقدار این خط yes باشد محتویات فایلی که در خط دوم، مسیر دهی شده خوانده می شود و فقط به یوزرهای که در این فایل ثبت شده اند اجازه دسترسی به FTP داده می شود.

**userlist\_deny=YES**

**userlist\_file=/etc/vsftpd/user\_list**

اگر مقدار این خط yes باشد محتویات فایلی که در خط دوم مسیر دهی شده خوانده می شود و اسامی وزرهایی که در این فایل قرار دارند نمی توانند به FTP دسترسی داشته باشند.

**nopriv\_user=ali**

اگر اجازه آپلود را به یوزرهای anon بدهیم باید قابلیت اجرا را از آنها بگیریم. برای این کار یک یوزر ساخته و اجازه اجرای فایل را از آن می گیریم. با وارد کردن نام یوزر مربوطه در اینجا، از این به بعد سطح دسترسی این یوزر به یوزرهای anon اعمال می شود. این کار برای امن سازی FTP لازم است.

**chroot\_local\_user=YES**

اگر این خط برابر YES باشد از تمام یوزرهای local قابلیت change root گرفته می شود.

**chroot\_list\_enable=YES**

**chroot\_list\_file=/etc/vsftpd/chroot\_list**

این خطوط برای jail کردن یوزرها به کار می رود. اگر قابلیت change root را از یوزری بگیریم اصطلاحاً میگوییم یوزر را jail کرده ایم. اگر بخواهیم بعضی از یوزرها local قابلیت change root نداشته باشند خط اول را برابر YES قرار داده و در خط دوم آدرس لیست یوزرهای انتخابی را وارد می کنیم. **این فیچر مخصوص یوزرهای local می باشد .**

**deny\_email\_enable=YES**

**banned\_email\_file=/etc/vsftpd/banned\_emails**

وقتی کاربر با مرورگر خود به یک FTP وصل می شود در صورتی که یوزر local نباشد به صورت یوزر anon لاگین کرده و به صورت پیش فرض وارد مسیر `/var/ftp/` می شود. اگر دقت کرده باشید بسیاری از سایتهای ftp موقع ورود از شما یوزر و پسورد نمی خواهد در صورتی که برای ورود، یوزر و پسورد anon نیاز است. توضیحی که برای این اتصال بدون پسورد وجود دارد این است که همه مرورگرها یک یوزر و پسورد پیش فرض داخلی برای احراز هویت دارند که در چنین مواقعی از آن استفاده میکنند. به طور مثال پسورد داخلی فایرفاکس [mozilla@example.com](mailto:mozilla@example.com) است. حال اگر این پسورد را در فایل `banned_email` وارد کنیم هیچ کاربری نمی تواند با مرورگر فایرفاکس به FTP متصل شود. این کار را برای محدود سازی اتصال با مرورگرهای خاص است. به این کار `banned` کردن ایمیل گفته می شود.

**port\_enable=YES**

**pasv\_enable=YES**

خط اول مشخص می کند FTP ما در Active mode کار کند و خط دوم حالت Passive

**idle\_session\_timeout=300**

این خط بیان میکند در صورتی که کاربر غیر فعال بود بعد از چند ثانیه ارتباط او توسط سرور قطع شود.

**delete\_failed\_uploads=YES**

اگر این گزینه فعال (YES) باشد تمامی آپلودهای failed شده پاک خواهند شد.

**download\_enable=YES**

اگر این گزینه فعال (YES) باشد تمامی درخواست های دانلود رد خواهند شد.

**listen\_port=21**

به طور پیش فرض پورت این سرویس ۲۱ است که برای امنیت بیشتر می توان این پورت را تغییر داد. البته همزمان باید در فایل کانفیگ این سرویس در `Xinetd` و فایل `/etc/services` تغییراتی را اعمال نمود.

**listen\_address=192.168.1.1**

اگر چندین کارت شبکه روی سرور داشته باشیم می توانیم یکی از آنها را به سرویس ftp اختصاص دهیم. حتی اگر درخواست ها زیاد باشد می توان دو یا چندین کارت شبکه را به این امر اختصاص داد. اگر این پارامتر مقداری نشود تمام کارت های شبکه برای این کار استفاده می شوند.

**allow\_anon\_ssl = YES**

اگر YES باشد کاربران anon مجاز به استفاده از ارتباطات امن SSL می باشند.

**ascii\_download\_enable=YES**

اگر YES باشد انتقال داده به صورت اسکی خواهد بود.

**force\_anon\_logins\_ssl = YES**

در صورت فعال بودن ssl\_enable و این گزینه کاربران anon مجبور به برقراری یک اتصال امن SSL برای ارسال رمز عبور خواهند بود.

**force\_dot\_file=YES**

در صورتی که این خط مقدار YES داشته باشد حتی اگر دستور فهرست کردن دایرکتوری ها بدون سویچ a باشد باز هم فایل های که ابتدایشان نقطه دارند نشان داده نخواهد شد (فایل های مخفی)

**ls\_recurse\_enable=YES**

اگر این آپشن فعال باشد اجازه اجرای دستور ls -R را دارید. فقط یک مشکلی که دارد این است که بکارگیری این آپشن در سایت های که حجم فایل بالایی بر روی آنها وجود دارد باعث هدر رفتن منابع سیستم می گردد.

**dirlist\_enable = NO**

اگر NO باشد به هیچ کدام از دستورات directory list اجازه اجرا داده نمی شود.

**hide\_ids=YES**

اگر این گزینه فعال باشد همه اطلاعات کاربر و گروه ، در لیست دایرکتوری را نمایش می دهد.

**no\_anon\_password=YES**

هنگامی که فعال باشد از کاربر anon درخواست یوزر و پسورد نمی کند.

**dual\_log\_enable=YES**

اگر این گزینه را فعال کنیم دو نوع لاگ برای ما تهیه می کند. یکی xferlog که لاگ پیش فرض است و یکی هم لاگ vsftpd را ثبت می کند.

**force\_dot\_files=YES**

اگر بخواهیم فایل های مخفی برای کاربران به نمایش در بیاید این گزینه را برابر YES قرار می دهیم.  
جهت اطلاعات بیشتر لطفا man vsftpd.conf را مطالعه بفرمائید.

## تغییر مسیر یوزرهای Local بعد از login به FTP

یوزرهای anon به صورت پیش فرض بعد از login به مسیر /var/ftp/ هدایت می شوند ولی یوزرهای Local سیستم بعد از Login به دایرکتوری Home خود وارد می شوند.

ما می توانیم کار کنیم که یوزرهای Local به جای اینکه به دایرکتوری Home خودشان وارد شوند به /var/ftp/ هدایت شوند و یا قابلیت ساخت دایرکتوری، فایل، آپلود و دانلود را هم به آنها داده ، یا آنها را بنا بر سیاست های کاری محدود کنیم. برای چنین کاری خطوط زیر را در فایل اصلی پیکربندی اصلاح می کنیم:

**#vi /etc/vsftpd/vsftpd.conf**

```
Anonymous_enable=NO
Local_enable=YES
Write_enable=YES
Local_umask=002
Dirmessage_enable=YES
Local_root=/var/ftp/pub
```

نکته : معمولا سیستم های گرافیکی، کانکشن FTP را cash میکنند.

نکته مهم : اگر گزینه ای را بخواهیم غیر فعال کنید بهتر است به جای کامنت کردن ، در جلوی آن **NO** را بنویسیم .

## محدود کردن دسترسی یوزرها به FTP

اگر بخواهیم تعداد خاصی از یوزرهای Local به FTP دسترسی داشته باشند باید اسامی آنها را داخل یکی از فایل های ftpusers و یا user\_list وارد کرده و در فایل کانفیگ تغییراتی را اعمال کنیم. اگر بخواهیم از بین تعداد زیادی از یوزرها فقط بعضی اجازه دسترسی به FTP داشته باشند باید گزینه userlist\_deny=??? را در فایل کانفیگ برابر NO قرار دهیم. در صورت NO بودن این گزینه فقط یوزرهای موجود در فایل user\_list می توانند به FTP دسترسی داشته باشند.

YES بودن گزینه userlist\_deny=??? چندان هم منطقی نیست. اگر این گزینه برابر YES باشد و اسامی تعریف شده در فایل user\_list در ftpusers هم موجود باشند آنوقت YES بودن این گزینه بی معنی می شود چون از طریق بررسی فایل ftpusers به یوزرها اجازه دسترسی داده می شود. در این حالت هر دو فایل جهت اعطا حق دسترسی مورد بررسی قرار می گیرند. اما اگر این مقدار برابر NO باشد دیگر فایل ftpusers مورد بررسی قرار نمی گیرد.

## Jail کردن یوزرها در FTP

وقتی کاربری به FTP لاگین می کند نباید بتواند به دایرکتوری، به غیر از ریشه ای که به آن وارد شده برود. به طور مثال اگر تعریف کرده باشیم یوزر به محض ورود به دایرکتوری /var/ftp/pub هدایت شود، نباید بتواند به سمت دایرکتوری بالایی تغییر دایرکتوری بدهد. به جلوگیری از چنین کاری Jail کردن یوزر گفته می شود.

با این کار قابلیت chroot را از یوزرها سلب کردیم. برای Jail کردن یوزرها آپشن زیر باید مقدار **YES داشته باشد**

قابلیت chroot غیر فعال می شود.

**chroot\_local\_user=YES**

اگر هم بخواهیم قابلیت chroot برای بعضی از یوزرها فعال شود کافی است اسم آنها را در یک فایل قرار داده و در خط زیر آدرس دهی کنیم.

**chroot\_list\_enable=YES**

**chroot\_list\_file=/etc/vsftpd/chroot\_list**

## بر طرف کردن Error 500

در centos سری 6 ممکن است در هنگام login یوزرهای local ، به آنها error 500 نشان داده شود . ممکن است نتوانند به دایرکتوری Home خود بروند یا اجازه write پیدا نکنند. یا فرضا ما یکسری از قابلیت ها را فعال می کنیم ولی در عمل کار نکنند. دلیل آن هم به خاطر عدم پیکربندی متغیرهای بولین Selinux است. این متغیرها را با دستور زیر می توان مشاهده کرد :

### #getsebool -a

یکی از مهمترین این متغیرها **allow\_ftpd\_anon\_write=off** می باشد. مثلا اگر در فایل کانفیگ اجازه رایت به یوزرهای anon داده شده باشد، تا این متغیر on نشود اجازه write به یوزرهای anon داده نمی شود. یکی دیگر از مهمترین متغیرها، **ftp\_home\_dir = off** می باشد. این متغیر به یوزرهای local اجازه می دهد تا بتوانند از طریق FTP وارد دایرکتوری home خود بشوند. فعال بودن این متغیر است که باعث می شود که error 500 برای یوزرها نمایش داده شود و از رفتن به دایرکتوری خانگی آنها ممانعت به عمل آید. با دستور setsebool میتوان مقدار این متغیرها را تغییر داد:

### #setsebool -P ftp\_home\_dir=1

در *Selinux* برای هر سرویس مقدار زیادی متغیر وجود دارد که باید بعد از راه اندازی هر سرویس متغیرهای آن را پیکربندی کنیم. بزرگترین اشتباه آن است که *Selinux* را خاموش کنیم.

*Selinux* در سه مد کار می کند:

1. enforcing : این مد بالاترین درجه امنیت در *Selinux* را دارا میباشد. اگر این مد را فعال کنیم تمام مازول های

امنیتی سیستم enable می شود.

2. permissive : در این مد *Selinux* فعال نیست و چیزی را **denay** نمی کند اما از همه چیز log برداری میکند .

3. disable : با فعال کردن این مد *Selinux* کاملا غیر فعال می شود.

فایل کانفیگ *Selinux* در مسیر **/etc/selinux/config** قرار دارد. برای تغییر در مدهای آن این فایل را باز کرده و در مقابل کلمه **SELINUX** مد مربوطه را وارد می کنیم و برای اعمال شدن آن حتما باید سیستم را یکبار ریست کنیم . از دستورات زیر هم جهت تغییر مد آن می توان استفاده کرد :

### #setenforce 0

### #echo 0 > /selinux/enforce

با دستور زیر هم می توان از وضعیت *Selinux* و مدهای کاری آن کسب اطلاع کرد :

### #setstatus

## ایجاد Mullti Homing در FTP

به طور معمول بر روی هر سیستم فقط یک سرویس دهنده FTP راه اندازی می شود در حالی که با Vsftpd می توان چندین سرویس دهنده مستقل FTP را روی یک سرور راه اندازی کرد. فرض کنید یک سرور داریم که هم زمان به اینترنت و شبکه داخلی سرویس می دهد و می خواهیم یک FTP به کاربران اینترنتی و یک FTP دیگر به کاربران داخلی سرویس بدهد. در چنین شرایطی از خاصیت Mullti Homing استفاده می کنیم. برای هر سرور باید یک فایل کانفیگ جداگانه با نام منحصر به فرد، در زیر دایرکتوری `/etc/vsftpd` ایجاد کنیم. برای هر کارت شبکه یک آدرس اختصاصی تنظیم کرده و در هر فایل کانفیگ یکی از آنها را وارد میکنیم. مهمترین گزینه ای که در فایل کانفیگ باید آورده شود `listen_address` است که به کارت شبکه مختص به سرویس اشاره دارد.

طبق توصیه اکید ردهت، مکان ذخیره لاگ هر کدام از این سرورها باید با دیگری فرق داشته باشد. قبلا گفته شد که `vsftpd` به دو صورت `log` برداری میکند که `xferlog` نحوه پیش فرض لاگ گرفتن این سرویس می باشد.

می توانیم به یک کارت شبکه سرور دو `IP` اختصاص داده و هر `IP` را مختص یک سرویس دهنده فایل قرار دهیم به این کار **Virtual Host** گفته می شود. برای این کار باید از فایل کانفیگ اصلی یک کپی با یک نام دلخواه ایجاد کرده و تنظیمات مربوطه به آدرس `IP` مورد نظر و پورت دلخواه را درون آن وارد کنیم.

### #vi /etc/vsftpd/vsftpd2.conf

```
listen=YES
local_enable=NO
anonymous_enable=YES
write_enable=YES
anon_max_rate=YES
anon_root=/opt
listen_address=192.168.1.1
listen_port=2020
```

### #vsftpd /etc/vsftpd/vsftpd2.conf

با این دستور فقط فایل `vsftpd2.conf` ریست می شود و بقیه `ftp` ها به کارشان ادامه می دهند. چون نیازی نیست همه آنها با هم ریست شوند پس بهتر است فقط فایل کانفیگ مربوطه را ریست کنیم.



## استفاده از دستور ftp به عنوان ابزار کلاینتی

نرم افزار پیش فرض کلاینتی اکثر توزیع های لینوکس ابزار ftp می باشد که برای کپی، انتقال، rename، حذف، ساختن یک فایل و همچنین تغییر سطح دسترسی می توان از آن استفاده کرد. برای جلوگیری از سرقت اطلاعات بسیار بهتر است که همواره از sftp یا همان secure ftp استفاده کنید که انتقال امن را فراهم می آورد. اگر FTP به صورت امن راه اندازی نشود اطلاعات را به صورت clear Text رد و بدل می کند. توصیه می شود بجای ftp از دستور lftp که دارای خصوصیات پیشرفته تر و در عین حال کار با آن نیز آسانتر است، استفاده کنید. دستور lftp یک دستور تعاملی است یعنی یک چیزی به آن می دهیم و یک چیزی به ما بر می گرداند و برای هر کاری باید یک دستور به آن بدهیم. دستور ftp به طور پیش فرض نمی توان به صورت anon به سرور متصل شد بلکه باید حتما نام یوزر local را وارد کنیم، این دقیقا بر عکس دستور lftp می باشد، چون lftp به صورت پیش فرض با یوزر anon و بدون پسورد به سرور متصل می شود که البته به جای پسورد باید یک کاراکتر دلخواه را وارد کنیم. lftp قادر است همزمان عمل احراز هویت و برقراری اتصال را انجام دهد ولی ftp قادر به انجام آن نیست.

نکته: اگر اول کامندی از علامت ! استفاده کنیم یعنی این دستور را روی سرور اجرا نکن بلکه باید آن را روی سیستم local اجرا کند.

نکته: زمانی که به سرور لاگین می کنیم یکسری کد به همراه پیامهایی به نمایش در می آید. این کدها از قبل تعریف شده هستند و برای ثبت log استفاده می شوند.

برای اتصال به یک سرویس دهنده فایل با استفاده از دستور ftp به شیوه زیر عمل کنیم:

```
$ftp ftp.example.com
username
password
```

و شیوه همزمان آن در lftp:

```
$lftp admin@ftp.example.com
```

و یا

```
$lftp -u admin ftp.example.com
```

به جای ftp.example.com می بایست hostname سرور مربوطه و یا نام یکی از دامنه های مستقر بر روی آن را بنویسیم و برای ورود، اطلاعات اکانت کاربری ftp متعلق به سرور مقصد را وارد نمائید. با دستور ftp نمی توان همزمان هم احراز هویت و هم اتصال برقرار کرد اما در lftp می توان با یک دستور هم لاگین کرده و احراز هویت کنیم. به طور مثال، مراحل زیر را مشاهده می فرمائید:

Trying 87.51.34.132...

Connected to ftp.freebsd.org.

220 ftp.beastie.tdk.net FTP server (Version 6.00LS) ready.

Name (ftp.freebsd.org:vivek): ftp

331 Guest login ok, send your email address as password.

Password:

230 Guest login ok, access restrictions apply.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp>

از این پس، به سبب اینکه از پروتکل ftp بهره می گیرید، اعلان پرامپت شما مطابق زیر خواهد بود:

ftp>

برای نمایش فایل ها و فولدر ها می توانید از دستور زیر استفاده نمایید:

ftp>ls

برای مثال احتمالا، اطلاعاتی شبیه به اطلاعات زیر را دریافت می کنید:

229 Entering Extended Passive Mode (|||60692|)

150 Opening ASCII mode data connection for '/bin/ls'.

total 10

drwxrwxr-x 2 0 5 512 Jul 19 2007 .snap

drwx----- 2 0 0 2048 Jul 19 2007 lost+found

drwxr-xr-x 3 1006 1006 512 Sep 21 2009 pub

drwxr-xr-x 3 1006 1006 512 Jun 5 2007 sup

drwxr-xr-x 4 1006 0 512 Sep 18 2009 www

226 Transfer complete.

ftp>

دقت نمایید که ستون آخر نمایش دهنده نام فایل ها و فولدر ها می باشد.

برای ورود به یک فولدر دیگر می توانید از دستور زیر استفاده کنید ، NOT دستور cd ، کامند lcd است.

ftp> cd folder-name

برای دریافت یک فایل می توانید از دستور get مطابق مثال زیر استفاده کنید. اگر بخواهیم فایل های دانلودی در یک مسیر

مشخص ذخیره شوند باید در محیط دستور ftp ابتدا به آن مسیری که روی سیستم local قرار دارد lcd کرده و سپس اقدام

به دانلود فایل ها نمائیم و اگر از سوئیچ -c استفاده کنیم در هر مرحله از دانلود یا آپلود ارتباط ما قطع شود در ارتباط بعدی

از باقی مانده کار شروع به دانلود یا آپلود می کند :

ftp> get -c resume.pdf

و مطابق ذیل مشاهده خواهید کرد که فایل مربوطه دانلود می گردد:

local: resume.pdf remote: resume.pdf

229 Entering Extended Passive Mode (|||55093|)

150 Opening BINARY mode data connection for 'resume.pdf' (53077 bytes).

100% |53077 12.58 KiB/s 00:00 ETA

226 Transfer complete.

53077 bytes received in 00:04 (12.57 KiB/s)

اگر در همین زمان می خواهید، محل دایرکتوری خود در سیستم Local و مبدا را تغییر دهید، دستور زیر مفید خواهد بود:

ftp>lcd /path/to/new/dir

مثل:

ftp>lcd /tmp

حتی می توانید با دستور زیر محل دایرکتوری خود در سرور اصلی مشخص نمایید:

ftp>lpwd

برای دریافت چندین فایل می توانید از دستور زیر استفاده نمایید:

ftp>mget \*

و یا:

ftp>mget \*.jpg

برای حذف یک فایل:

ftp>delete fileName

ftp> delete output.jpg

و اما دستور زیر که شاید برای خیلی ها تازگی داشته باشد؛ اگر می خواهید فایلی را در سرور از طریق shell آپلود نمایید،

یعنی به سروری که متصل شده اید منتقل کنید، کافی است دستور زیر را استفاده کنید:

ftp> put FileName

مثلا می خواهید فایل logo.jpg را از کامپیوتر محلی خود به سرور از طریق shell انتقال دهید:

ftp> put logo.jpg

و برای آپلود چندین فایل:

ftp>mput \*

ftp>mput \*.pl

اضافه کردن یک دایرکتوری:

ftp>mkdirdirName

حذف کردن یک دایرکتوری:

ftp>rmdirdirName

و در نهایت، برای خروج از ftp می توانید دستورات زیر را بکار ببرید:

ftp> quit

## فرق بین مد باینری و اسکی

کلا در دنیا دو نوع کلی فایل وجود دارد 1-اسکی 2-باینری. فایل های اسکی فایل های text base می باشند مثل فایل های php,asp,html,pdf و کلا هر فایلی که بتوان محتوای آن را خواند فایل اسکی است، به غیر از این، تمام فایل ها باینری هستند مثل عکس، فیلم، آهنگ و ...

دستور ftp می تواند در دو مد اسکی و باینری فایل ها را منتقل کند. اگر مد انتقال فایل با فایل دریافتی هم خوانی نداشته باشد فایل ها در مقصد برای باز شدن دچار مشکل خواهند شد. پس اگر فایل باینری باشد باید در مد باینری و اگر اسکی باشد باید در مد اسکی نقل و انتقال صورت پذیرد. برای تغییر مد کافی است کلمه ascii را تایپ کنیم. به همین سادگی مد ترانسفر از باینری به اسکی تغییر می کند. نکته: دستور ftp مانند سرویس دهنده آن در دو مد active و passive ارتباط برقرار می کند.

ftp>ascii

## SCP یک جایگزین امن برای ftp

از دیدگاه شبکه، سرویس FTP سرویس امنی نیست، زیرا نام کاربری، کلمه عبور و داده ها همگی بدون هیچ گونه رمزنگاری بر روی شبکه مبادله می شوند. شکل امن این سرویس SFTP و SCP هستند، که به عنوان جزئی از بسته Openssh در دسترس بوده و به شکل پیش فرض در سیستم های CentOS و Redhat نصب می باشد. به خاطر داشته باشید که SCP برخلاف FTP قابلیت پشتیبانی از بارگیری بی نشان (Anonymous Download) را دارا نیست. فرمان SCP در لینوکس، قالبی همانند فرمان cp را داراست. اولین پارامتر فایل مبدا و دومین پارامتر فایل مقصد را مشخص می کند. در هنگام کپی کردن یا گذاشتن فایل ها در سرویس دهنده SSH، کاربر باید توسط scp وارد سرویس دهنده شود که برای این کار باید نام سرویس دهنده، نام کاربری و کلمه عبور را با موفقیت به عنوان آرگومان های ورودی به آن ارسال کند. پس از این، فایل موردنظر با پیشوندی از نام کاربری و سرویس دهنده که با یک @ از یکدیگر جدا شده اند، در سمت سرویس دهنده پردازش می شود. قالب مربوط به این موضوع بدین شکل است:

username@servername:filename

username@servername:directoryname

به طور مثال فرض کنید نیاز به کپی کردن فایل /etc/syslog.conf بر روی سرویس دهنده ای با آدرس 192.168.1.100 و نام کاربری Peter داریم. بدین منظور از قالب

etc/syslog.conf/:peter@192.168.1.100

استفاده می کنیم. در صورت تمایل به کپی برداری از کل شاخه /etc قالب فوق بدین شکل تغییر می یابد.

/etc/:1.100 .Peter@192.168

## متداولترین کدهای وضعیت FTP

متداولترین کدهای وضعیت FTP به همراه مفهوم هریک در جدول زیر نشان داده شده است .

کدهای وضعیت سری 100	
110	Restart reply
120	Service ready in x minutes
125	Connection currently open, transfer starting
150	File status okay, about to open data
کدهای وضعیت سری 200	
200	Command okay
202	Command not implemented, superfluous at this site
211	System status/help reply
212	Directory status
213	File status
214	System Help message
215	NAME system type
220	Service ready for next user.
221	Service closing control connection. Logged off where appropriate
225	Data connection open; no transfer in progress.
226	Closing data connection. Requested action successful
227	Entering Passive Mode
230	User logged in, continue
250	Requested file action okay, completed
257	"PATHNAME" created.
کدهای وضعیت سری 300	
331	User name okay, need password.
332	Need account for login
350	Requested file action pending further information.
کدهای وضعیت سری 400	
421	Service not available, closing control connection.

425	Can't open data connection
426	Connection closed; transfer aborted.
450	Requested file action not taken. File not available - busy etc..
451	Request aborted: error on server in processing.
452	Requested action not taken. Insufficient resources on system
کدهای وضعیت سری 500	
500	Syntax error, command unrecognized
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands
504	Command not implemented for that parameter.
530	Not logged in.
532	Need account for storing files
550	Requested action not taken. File unavailable
552	Requested file action aborted. Exceeded storage allocation
553	Requested action not taken. File name not allowed
مفهوم برخی از کدهای متداول	
226	دستور بدون هیچگونه خطائی اجراء گردید.
230	زمانی این کد نمایش داده می شود که یک سرویس گیرنده رمز عبور خود را به درستی درج و عملیات login با موفقیت انجام شده باشد.
231	کد فوق نشاندهنده دریافت username ارسالی سرویس گیرنده توسط سرویس دهنده می باشد و تائیدی است بر اعلام وصول Username (نه صحت آن) .
501	دستور تایپ شده دارای خطاء گرامری است و می بایست مجدداً "دستور تایپ گردد".
530	عملیات login با موفقیت انجام نشده است. ممکن است Username و یا رمز عبور اشتباه باشد.
550	فایل مشخص شده در دستور تایپ شده نامعتبر است.

Linux Cookie in Persian v.2

## فصل چهارم

### راه اندازی SSH در لینوکس



# ssh



# آشنایی با پروتکل SSH

مقدمه :

در سال 1995 یک دانشجوی دانشگاه هلسینکی به نام Tatu Ylönen پس از آنکه اطلاعات مهمی مثل رمز و نام های کاربری در شبکه دانشگاه مورد Sniff قرار گرفت به فکر ایجاد یک شبکه امن افتاد که این فکر در نهایت منجر به ایجاد یک Shell امن شد که جایگزینی برای telnet , rlogin , rcp , rsh , ftp شد. البته این دستورات هنوز در لینوکس وجود دارند اما با وجود SSH دیگر کسی از آنها استفاده نمی کند و تمامی اینها با دستورات scp و ssh جایگزین شدند.

SSH مخفف Secure Shell است. SSH یک پروتکل ارتباطی امن بر پایه TCP/IP بین سرویس دهنده و سرویس گیرنده است که با رمز گذاری داده ها از افشای اطلاعات در طول مسیر جلوگیری کرده و یک کانال امن در سیستم عامل سرور برای دستیابی به خط فرمان ، برای کلاینت ایجاد می کند.

کلمه Shell ممکن است این تصور را ایجاد کند که SSH یک مفسر فرمان است اما این کاملاً اشتباه می باشد بلکه SSH یک پروتکل ارتباط امن می باشد. الگوی اولیه رمزنگاری در سرویس SSH که در سال 1995 ارائه گردید در زمان خودش ابزاری مناسب محسوب می شد ولی با گذشت زمان محدودیت هایی در استفاده از آن پدیدار گشت که به جهت رفع این محدودیت ها نسخه دوم این سرویس ارائه شد. همواره سعی کنید تا با تنظیم عبارت Protocol در فایل های پیکربندی سیستم، خود را ملزم به استفاده از نسخه دوم این سرویس کنید. استفاده از SSH محدودیتهایی نظیر لایسنس و پرداخت هزینه را در بردارد ، لذا برای رفع این محدودیت گروه OpenBSD شروع به ارائه موازی نسخه جدیدی به صورت رایگان نمود که نام این محصول را OpenSSH گذاشتند. با خرید لایسنس شرکت Tectia که ارائه کننده SSH تجاری می باشد علاوه بر پشتیبانی از کاربر ، از صفحه مدیریت تحت وب SSH بهره مند می شوید ولی چنین مزایایی در OpenSSH وجود ندارد. راز محبوبیت پروتکل SSH کدگذاری شبکه ، ایجاد تونل امن و پشتیبانی از انواع متد های دیگر ارتباط امن می باشد. مزایای استفاده از SSH نسبت به شیوه های قدیمی ارتباط از راه دور به شرح زیر می باشد:

## 1. رمز گذاری داده ها : (Encryption Data)

همانطور که گفته شد نیاز به یک اتصال امن بین سرور و کلاینتها و جلوگیری از Sniff ، مهمترین دلیل استفاده از SSH می باشد.

## 2. بررسی یکپارچگی داده ها (Data Integrity)

این خاصیت برای جلوگیری از حمله های Insertion and Replay Attacks بسیار مفید می باشد. لازم به ذکر است رمز گذاری داده ها بدون استفاده از Session ID نمی تواند از حملات replay attacks جلوگیری کند. از نسخه دو این پروتکل بعد این قابلیت اضافه شد تا پکتها در مسیر ارسال جایگزین و یا شبیه سازی نشوند. در این نوع

حمله هکر دیتای تبادلی در نشست را مانیتور نمی کند بلکه مثل نرم افزارهای Keylogger خروجی صفحه کلید را مانیتور نموده و با مقایسه پکتهای تایپ شده با ترافیک جاری SSH متوجه کارکترهای خاص تایپ شده می شود.

### 3. قابلیت فشرده سازی : (Compression)

این پروتکل علاوه بر رمز نگاری ، اطلاعات ارسالی را فشرده می کند که این کار در ارتباطات کم سرعت بسیار مفید خواهد بود.

### 4. عدم اتصال به سرور جعلی : (Prevent Impersonation of host)

در یک اتصال SSH هنگام اتصال به سرور، هویت سنجی صورت می گیرد و اگر یک ماشین با مشخصات سرور در مسیر کلاینت قرار گرفته باشد امکان میزبانی کلاینت و یا بالعکس را ندارد. در حالی که در پروتکل های قدیمی تر مثل Telnet این اتفاق اجتناب ناپذیر است. این نوع حمله به حمله مرد میانی موسوم است. (Man-In-The-Middle-Attack or MITM Attack).

### 5. لاگ فایل : (Log Access)

SSH امکان فعال و یا غیر فعال شدن فرایند تهیه لاگ فایل ها را دارد. با فعال شدن این امکان در مواقع بروز مشکل مدیر سیستم بعد از بروز خطا اولین موردی که برای رفع مشکل بررسی میکند لاگ فایل ها می باشد.

### 6. امکان استفاده از X11 Applications

SSH این قابلیت را دارد که برنامه های دیگر مثل نرم افزارهای گرافیکی را کد گذاری کند. به این قابلیت Port Forwarding هم می گویند. از این قابلیت برای Tunneling استفاده می شود.

### 7. موجود بودن کامندهای موسوم به r-Command

تمام امکانات کامندهای موسوم به r-Command در SSH وجود دارد. به عنوان مثال از سرور 1 به سرور 2 دستور date را اجرا می کنیم :

```
skywan13@localhost ~]$ ssh userx@x.x.x.x date
```

```
:ssh userx@x.x.x.x 's password
```

```
Tue Sep 21 18:11:28 IRDT 2014
```

علاوه بر کاربردهای رایج این پروتکل ، انعطاف پذیری بر حسب نیاز موجب محبوبیت این پروتکل در بین کاربران و متخصصان کامپیوتر شده است.

## نصب و راه اندازی سرویس OpenSSH

عموما به شکل پیش فرض سرویس Openssh در زمان نصب سیستم عامل نصب می شود. همچنین از آنجا که scp و ssh جزئی از یک برنامه هستند، هر دو از یک فایل پیکربندی استفاده کرده و توسط سرویس SSH مدیریت می شوند. بسته های rpm سرویس SSH را به راحتی می توان از منابع این نوع بسته ها در Internet تهیه کرد. معتبرترین مرجع جهت تهیه بسته های مربوط به این سرویس سایت ssh.com است که در آن شما قادر خواهید بود نسخه های تجاری و غیر تجاری سرویس SSH را به راحتی تهیه کنید. این سرویس از طریق کامپایل کد منبع آن و ابزار apt-get در سیستم های مبتنی بر debian نیز به راحتی قابل تهیه و استفاده است. همچنین کاربران فدورا با استفاده از yum می توانند این بسته را نصب کنند هرچند به صورت پیش فرض این بسته روی اکثر توزیعهای لینوکسی نصب شده می باشد. SSH با حروف بزرگ به طور کلی به پروتکل SSH اطلاق میشود و ssh با حروف کوچک به نرم افزار سمت کلاینت گفته می شود که برای اتصال به سرور به کار می رود. پیش نیاز نصب SSH پکیج های zlib و OpenSSL است که در صورت استفاده از Yum این نیازمندیها به صورت اتوماتیک نصب خواهند شد. SSH تحت نظر init اداره می شود.

ابتدا باید از نصب بودن پکیج Openssh اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم:

```
#rpm -qa | grep openssh
```

در صورت نصب نبودن، در سیستم های ردهت جهت نصب openssh از yum استفاده می کنیم:

```
#yum -y install openssh
```

بعد از نصب، باید اطمینان حاصل کنیم که آیا پکیج Openssh بر روی سیستم نصب شده است یا خیر لذا دوباره با دستور زیر از سیستم query می گیریم:

```
#rpm -qa | grep openssh
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم:

```
#rpm -ql openssh
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم:

```
#rpm -qi openssh
```

سپس با دستور chkconfig مشخص می کنیم در تمام runlevel هایی فعال باشد:

```
#chkconfig openssh on
```

و در انتها سرویس را reset می کنیم:

```
#service sshd restart
```

با دستور netstat از غیرقابل استفاده بودن پورت SSH توسط سایر برنامه های سیستم اطمینان حاصل میکنیم:

```
# netstat -an | grep 435
```

نکته : با نصب openssh چهار پکیج روی سیستم نصب می شوند که این دو پکیج مهمترین آنها هستند :

**openssh-server      openssh-client**

همانطور که گفته شد SSH یک پروتکل کلاینت سروری است و در هر دو طرف یعنی سرور و کلاینت لینوکسی باید پیکر بندیهای لازم صورت پذیرد.

در سمت سرور در مسیر `/etc/ssh/` و در سمت کلاینت لینوکسی در مسیر `/home/user/.ssh/` باید تغییراتی اعمال شود. از فایل های مسیر `/etc/ssh/` برای پیکربندی سرور SSH استفاده می شود که این تنظیمات global بوده و به همه کلاینتها و یوزرها اعمال می شود. اما تغییراتی که در `/home/user/.ssh/` انجام می دهیم برای تحت تاثیر قرار دادن عملکرد کامندهای ssh و scp است و فقط در سیستم کلاینت اعمال شده و local می باشد.

در این فصل ابتدا تنظیمات سمت سرور را مرور کرده سپس به تنظیمات سمت کلاینت خواهیم پرداخت.



## بررسی فایل های موجود در /etc/ssh

در زیر این دایرکتوری فایل های پیکربندی و الگوریتم های این پروتکل قرار دارد که مهمترین آنها در زیر توضیح داده شده است :

sshd\_config  
ssh\_config  
ssh\_host\_key  
ssh\_host\_dsa\_key  
ssh\_host\_rsa\_key  
moduli

**sshd\_config :** این فایل مهمترین فایل این دایرکتوری است . با این فایل سرویس SSH را پیکربندی می کنیم. تنظیمات این فایل global بوده و همه یوزرها اعمال می شود.

**ssh\_config :** با این فایل کلاینتهای ssh مورد پیکربندی قرار می گیرند. اگر در این فایل تغییراتی اعمال کنیم به همه یوزرها اعمال می شود.

**ssh\_host\_key :** این فایل کلید SSH ورژن یک می باشد که از الگوریتم خاصی برای رمز نگاری استفاده نمی کند.

ssh\_host\_dsa\_key  
ssh\_host\_rsa\_key

این فایل ها کلیدهای SSH ورژن 2 می باشند که از الگوریتمهای dsa و rsa برای رمز نگاری استفاده می کنند.

**moduli :** اطلاعات dh که در معاوضه کلیدها بین طرفین اسفاده می شود در این فایل قرار می گیرد.

## پیکربندی سرویس SSH

در این قسمت به تشریح بعضی از قسمت‌های فایل پیکربندی سرور SSH می‌پردازیم. برای اطلاع دقیق از تمامی آپشن‌های این سرویس به `man sshd_conf` رجوع کنید:

### Port 445

پورت پیش فرض سرویس SSH پورت 22 tcp می‌باشد. SSH با این پورت روی تمام کارت‌های شبکه به حالت `listen` می‌رود. هر گاه زمانی احساس کردید افرادی قصد نفوذ به سیستم شما، از طریق پورت شناخته شده ای مثل 22 را دارند، می‌توانید با تغییر آن، به پورتهایی که تداخلی با برنامه‌های کاربردی موجود در سیستم ندارد، از این امر پیشگیری کنید. این کار را می‌توان تنها یک پیشگیری اولیه محسوب کرد، زیرا برنامه‌هایی در شبکه جهت تشخیص پورت‌هایی که هم اکنون در حال اجرای سرویس `ssh` هستند، نیز وجود دارند.

### AddressFamily any

این خط می‌تواند سه مقدار داشته باشد. مقدار این خط مشخص کننده این است که از چه ورژن IP پشتیبانی کند. مقادیر این خط:

- `inet`: مشخص کننده IP ورژن 4 است.
- `int6`: مشخص کننده IP ورژن 6 است.
- `any`: یعنی از هر دو ورژن IP پشتیبانی کند.

### LitenAddress 192.168.1.1

`Ssh` به طور پیش فرض با تمام IPها ارتباط برقرار می‌کند و اجازه ارتباط هم می‌دهد در اینجا می‌توانیم کارت شبکه خاصی را به عنوان کارت اختصاصی SSH آدرس دهی کنیم. این خط مربوط به IP ورژن 4 است.

### ListenAddress ::

در IP ورژن 6 مقدار این خط معنی همه را می‌دهد.

**HostKey /etc/ssh/ssh\_host\_rsa\_key**

**HostKey /etc/ssh/ssh\_host\_dsa\_key**

با این آپشن‌ها مسیر کلیدهای `rsa` و `dsa` را معین می‌کنیم که البته نیازی به تعویض این مسیرها نیست.

**KeyRegenerationInterval 1h**

سرور SSH به طور پیش فرض هر یک ساعت کلیدهای تولید شده روی سرور را تولید مجدد می کند تا از حملات Capture data و یافتن الگوریتم جلوگیری کند.

**ServerkeyBits 2048**

طول کلیدهای رمزنگاری تولید شده به طور پیش فرض 1024 بیت است که برای امنیت بیشتر بهتر است بر روی 2048 تنظیم شود.

**SyslogFacility AUTHPRIV**

این خط مشخص میکند log هر کاربری که با ssh به سرور لاگین کرده و یوزر پسورد وارد کند ثبت شود.

**LogLevel INFO**

این خط می تواند دو مقدار INFO و DEBUG داشته باشد که سطوح log برداری را مشخص می کنند.

**PermitRootLogin no**

به صورت پیش فرض یوزر root اجازه Remote access از طریق ssh را دارد که باید این اجازه از آن گرفته شود. با قراردادن no این اجازه را از آن می گیریم.

**LoginGraceTime 2m**

این زمان مشخص می کند کلاسی که session برقرار کرده و صفحه لاگین را در اختیار دارد 120 ثانیه وقت دارد یوزر پسورد را تایپ و به سرور لاگین نماید اگر در این مدت اقدامی صورت ندهد کانکشن آن قطع شده و صفحه لاگین بسته می شود.

**MAXAuthTrise 6**

اگر به صورت پیش فرض یوزری 6 مرتبه پسورد را اشتباه وارد کند ، ssh صفحه لاگین را از او گرفته و کانکشن را قطع می نماید.

**Allow and Deny****Allow Users****Deny Users****Allow Groups**

## Deny Groups

در حالت پیش فرض تمامی یوزرها می توانند با استفاده از یوزرشان در سرور به آن ssh زده و لاگین کنند. اگر بخواهیم به یوزر یا گروهی اجازه دسترسی به ssh را داده یا دسترسی آنها را محدود کنیم باید در جلوی این آپشن ها یا نام یوزرها و گروه ها را جدا از هم وارد کرده و یا در لیستی جداگانه وارد و در اینجا آدرس فایل مربوطه را وارد کنیم.

## AuthorizedkeysFiles .ssh/authorized\_keys

این فایل جهت احراز هویت keybase مورد استفاده قرار می گیرد که بر روی دایرکتوری Home هر کلاینت لینوکس قرار دارد. این خط مسیر این فایل را مشخص می کند.

## PasswordAuthentication yes

## PubkeyAuthentication yes

در SSH دو نوع احراز هویت وجود دارد، یکی بر اساس Password و دیگری بر اساس Keybase می باشد. احراز هویت پیش فرض از نوع پسورد است. خط اول احراز هویت را بر اساس پسورد و خط دوم احراز هویت بر اساس کلید را فعال می کند.

## PermitEmptyPassword no

بعضی از اکانتها دارای پسورد نیستند برای جلوگیری از لاگین چنین اکانت هایی مقدار این خط را برابر no قرار می دهیم.

## UsePam yes or no

سرویس SSH می تواند توسط دو مکانیزم امنیتی Pam و tcp\_wrapper کنترل می شود. فعال کردن این گزینه باعث می شود کنترل امنیتی این سرویس تحت اختیار Pam قرار بگیرد که در این صورت باید فایل زیر را مورد پیکربندی قرار دهیم. /etc/security/access.conf و اگر بخواهیم با مکانیزم tcp\_wrapper مورد کنترل قرار بگیرد باید فایل /etc/host.allow را کانفیگ کنیم.

جهت کار کردن با این دو فایل حتما باید راهنمای آنها را مورد مطالعه قرار دهیم. اگر بخواهیم بفهمیم کدام سرویس توسط tcp\_wrapper کنترل می شود از این دستورات استفاده می کنیم.

**#whereis sshd**

**#ldd /usr/sbin/sshd**

دستور ldd تمامی ماژول هائی که یک سرویس از آن استفاده می کند را نشان می دهد. خروجی این دستور لیست تمام ماژول های sshd را نشان می دهد. سومین ماژول نشان داده شده در خروجی، ماژول libwrap.so است. هر سرویسی که این ماژول را داشته باشد یعنی توسط tcp\_wrapper کنترل میشود.



**Accept ENV**

این عبارت متغیرهایی را نشان می دهد که سرور ssh اجازه Export کردن آنها بر روی کلاینت را دارد.

**X11Forwarding yes**

این امکان را به ما می دهد تا بعد از اتصال به سرور SSH به صورت گرافیکی بتوانیم سرور را پیکربندی کنیم. بعد از اتصال ، ما با یک صفحه مشکی رنگ مواجه می شویم ، اما می توانیم از دستوراتی استفاده کنیم که دارای یک محیط گرافیکی ساده ای هستند مثل دستورات `system-config-network` , `rconf` , `setup`. این دستورات کنسول گرافیکی خاصی را بر روی سرور اجرا می کنند. بهتر است این گزینه بر روی `no` تنظیم شود.

**Banner /etc/ssh/banner**

می توانیم پیامی طراحی کرده و در مسیر گفته شده قرار دهیم تا در موقع اتصال هر یوزر به SSH این بنر به نمایش در بیاید.

**Printmotd yes**

اگر این خط برابر `yes` باشد می توان پیامی طراحی و در مسیر `/etc/motd` قرار داد تا در موقع اتصال آن را نمایش دهد.

**Protocol 2**

Ssh دارای دو ورژن 1 و 2 می باشد ورژن 1 دارای آسیب پذیری های `mitm` است و نباید به هیچ عنوان از آن استفاده شود.

**ClientAliveInterval 600****ClientAliveCountMax 0**

می توان یک حالت `Idle TimeOut` برای یوزرها ایجاد کرد تا اگر بین کار آنها فاصله ای بیافتد و عملیاتی صورت نگیرد باعث `Logout` در یوزر شود با اضافه کردن این دو خط و مقادیر مورد نظر این امکان فعال میشود.

**HostBasedAuthentication no**

تا وقتی که این ویژگی فعال باشد یک یوزر از هاست خودش می تواند به هاست دیگری در شبکه هم لاگین کند .

**IgnoreRhosts yes**

فعال بودن این خط باعث می شود یوزرها نتوانند به فایل های `shosts` و `rhosts`. دسترسی داشته باشند.

**نکته:** در حالت پیش فرض یوزرها امکان دسترسی به دایرکتوری هایی به جز دایرکتوری اصلی خود را دارند و می توانند به دایرکتوری هایی مانند bin,etc,..... دسترسی پیدا کنند. با استفاده از سیستم عامل هایی که بر پایه chroot هستند و یا ابزاری مانند rssh می توان ssh را در برابر دیگر یوزرها ایمن کرد

## استفاده از ssh جهت اتصال به یک کامپیوتر در شبکه

استفاده از ssh بسیار شبیه telnet است. جهت اتصال به یک ماشین دیگر در شبکه یا یک سرور SSH تحت یک کاربر دلخواه از آن ماشین از سوئیچ -ا استفاده می کنیم. در اینجا چند مثال جهت اتصال به یک سرویس دهنده SSH ارائه شده که به بررسی آنها می پردازیم:

با استفاده از این فرمان تحت کاربر root به کامپیوتر 192.168.1.1 در شبکه متصل می شویم.

```
# ssh 192.168.1.1
```

حال اگر بخواهیم تحت یک کاربر دیگر این اتصال را انجام دهیم، از یکی از قالب های زیر می توانیم استفاده کنیم:

```
# ssh -l ali 192.168.1.1
# ssh ali@192.168.1.1
```

و اگر بخواهیم بروی پورتی غیر از پورت 22 عملیات Login را انجام دهیم، دستور فوق بدین شکل تغییر پیدا می کند:

```
# ssh -P 435 ali@192.168.1.1
```

و در صورتی که مایل باشیم، دستوراتی مانند system-config-network که محیط گرافیکی دارند را اجرا کنیم از سوئیچ -X بهره میبریم:

```
# ssh -X ali@192.168.1.1
```

در اولین مرتبه ای که از طریق کلاینت لینوکسی با سروری ارتباط ssh برقرار می کنیم یک هشدار دریافت خواهیم کرد، دلیل آن هم این است که سرور مقابل می خواهد بر روی سیستم ما یک کلید آپلود کند.

یکی از امکانات جالب ssh قابلیت ورود و اجرای فرامین منفرد در یکی از سیستم های شبکه است. برای این کار کافی است فرمان مورد نظر را در یک جفت کوتیشن، در انتهای فرمان ssh قرار دهیم. در مثال زیر یک کاربر قصد دارد به نسخه کرنل موجود بروی سرویس دهنده 192.168.1.1 پی برد که برای این کار فرمان `uname -a` را بر روی سرویس دهنده اجرا می کند و بلافاصله خروجی آن به نمایش در می آید:

```
# ssh ali@192.168.1.1 "uname -a"
```

```
Linux yadi 2.6.8-1.521 #1 Mon Aug 10:10:17 EDT 2004 i686 i686 i386
```

## مبحث SSH Client

در این قسمت ، می خواهیم تنظیمات ssh client را بررسی کنیم. زمانی که برای اولین بار از طریق ssh به یک سرور یا ماشینی متصل می شویم، پیامی مبنی بر اینکه ماشین ما توسط سیستم مقصد شناخته شده نیست را دریافت خواهیم کرد. در همین زمان درخواستی جهت ذخیره سازی یک نسخه از کلیدهای شناسایی، ssh سرور مقصد بر روی کامپیوتر خودمان دریافت می کنیم که با تائید آن یک RSA key fingerprint که همان کلید Public سرور مقصد است بر روی سیستم Local آپلود می شود. این رویه را در زیر می توانید مشاهده کنید :

روال ذخیره سازی کلید:

```
# ssh 192.168.1.1
```

```
The authenticity of host 192.168.1.1 (192.168.1.1)' can't be established
```

```
RSA key fingerprint is 5d:d2:f5:21:fa:07:64:0d:63:1b:3b:ee:a6:58:58:bb
```

```
Are you sure you want to continue connecting (yes/no)? Yes
```

```
Warning : Permanently added '192.168.1.1' (RSA) to the list of known hosts.
```

```
root@192.168.1.1' password
```

```
Last login: The Nov 13 11:17:36 2014 from 192.168.1.1
```

```
No mail
```

این کلید در دایرکتوری home کاربر در پوشه ای به نام ssh. که یک پوشه مخفی است ذخیره می شود. در این پوشه فایل به نام known\_host وجود دارد که حاوی کلید های Public سرورها یا کلاینتهایی است که ما به آنها کانکشن زده ایم. ssh سیستم local توسط public key دریافتی از سرور و private Key خودش اطلاعات را رمزنگاری کرده و به سمت سرور ارسال می کند. pubkey سرور مثل قفل میباید که کلید آن privkey سرور است و با آن قفل دیتا رمز شده را باز میکند.

اگر سیستم عامل سرور یا سرویس SSH را بر روی سرور (server ssh) مجددا نصب کنیم و یا ip سرور را تغییر دهیم، کلیدهای pub تولید شده در سمت سرور با کلیدهای ذخیره شده در known\_hosts کامپیوترهای سرویس گیرنده تطابق نخواهند داشت و از این رو ارتباط ssh بین کلاینتها و سرور برقرار نشده و کاربر پیغام خطایی مانند زیر دریافت می کند که در آن احتمال بروز حمله از طریق هکرها هشدار داده می شود. علت اصلی این پیام ، عدم همخوانی کلید pub سرور با کلید pub موجود بر روی کلاینت است.

پیغام های خطای سرویس دهنده ssh

```
#ssh ali@192.168.1.1
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED @
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY

Someone could be eavesdropping on you right now (man-in-the-middle-attack)

It is also possible that the RSA host key has just been changed

The fingerprint for the RSA key sent by the remote host is

d:d2:f5:21:fa:07:64:0d:63:1b:3b:ee:a6:58:bb5

Please contact your system administrator

Add correct host key in /root/.ssh/known\_hosts:2

RSA host key for 192.168.1.10 has changed and you have requested strict checking

Host key verification failed

اگر اطمینان دارید که بروز پیام به دلیل نصب مجدد سرویس یا سیستم عامل سرور و یا تغییر IP سرور است، کافی است فایل known\_host را ویرایش کرده و خطوط مربوط به کلید قبلی سرویس دهنده SSH را از آن حذف کنیم. پس از این کار با اتصال مجدد به سرویس دهنده SSH مجدداً پیغامی مبنی بر ذخیره سازی کلید جدید در فایل ~/.ssh/known\_hosts دریافت می کنید و از این پس جلسات کاری مربوط به سرویس SSH بدون مشکل انجام خواهد شد چون عدم همخوانی کلیدها با دانلود کلید جدید و ذخیره آن در فایل known\_host بر طرف گردیده است. به ازای ارتباط با هر سرور یک محتوای کلید pub در فایل known\_host ذخیره می شود.

اگر بخواهیم دو کلید pub را با هم مقایسه کنیم یکی از آنها را از درون فایل known\_host به یک فایل جدید منتقل کرده سپس توسط دستور diff آن دو را با هم مقایسه می کنیم.

# diff file1 file2

## نصب و راه اندازی SSH Client

همانطور که گفته شد ssh یک نرم افزار انحصاری است که بابت قابلیت های آن باید لایسنس خریداری کرد. اما بسته openssh رایگان بوده و محدودیت های ssh را ندارد. لذا جهت استفاده از ssh در کلاینتهای لینوکسی هم از openssh استفاده می شود.

ابتدا باید از نصب بودن پکیج openssh-clients اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep openssh-clients

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب openssh-clients از yum استفاده می کنیم :

#yum -y install openssh-clients

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج Openssh بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

#rpm -qa | grep openssh

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm -ql openssh-clients
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm -qi openssh-clients
```

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
#chkconfig sshd on
```

کانفیگ سرور SSH زمانی سودمند است که کلايتها بخوانند به آن کانکشن بزنند اما کلاینت را برای این تنظیم میکنیم تا بتواند از سروری که تنظیمات آن پیش فرض نیست استفاده کند. فایل پیکربندی ssh Client در مسیر /etc/ssh/ssh\_config قرار دارد. و در انتها با دستور زیر چک میکنیم ssh روی پورت 22 و یا هر پورت دیگری به حالت listen رفته باشد.

```
#netstat -ntpl | grep 22
```

## استفاده از ارتباط بدون پسورد در SSH Client

در مواردی برای اجرای مجموعه ای از دستورالعمل های متوالی در قالب یک اسکریپت لازم است که ، امکان کپی کردن فایل ها از طریق SCP را بدون وارد کردن کلمه عبور داشته باشیم. یا جهت مصون ماندن از خطر لو رفتن پسورد توسط نرم افزارهای مخرب نخواهیم پسوردی جهت احراز هویت وارد کنیم. به این نوع از احراز هویت Key Base Authentication گفته می شود.

در زمان استفاده از این امکان در سرویس ssh لازم نیست هیچ گونه نگرانی برای فاش شدن کلمه عبور و یا از کار افتادن اسکریپت مورد نظرمان با تغییر کلمه عبور داشته باشیم. به راحتی جهت انجام این کار می توان سرویس SSH را پیکربندی کرد تنها باید کلید این نوع از ارتباط را به سرور معرفی کنیم. بدین وسیله سرویس دهنده ها قادر خواهند بود به کمک این کلیدهای از پیش نصب شده یکدیگر را تایید کرده، به تبادل دیتا بپردازند. ریسک امنیتی که در این روش وجود دارد این است که امکان دسترسی به یک حساب کاربری بر روی سرورتنها از طریق وارد کردن نام کاربری صورت می گیرد که برای کم کردن خطر احتمالی این کار باید از حساب های کاربری غیر مدیریتی در سرور استفاده کنیم، تا در صورت فاش شدن حساب کاربری مربوط به SSH، امکان اعمال نفوذ در کارهای مدیریتی سیستم میسر نباشد.

در روش اول کلاینت ، pubkey سرور را دریافت و اطلاعات را با pubkey سرور و privkey خودش رمز کرده و برای سرور ارسال می کند. اما در این روش عکس این عمل را انجام می دهد. یعنی کلاینت یک pubkey و یک privkey تولید کرده و pubkey را بر روی سرور آپلود میکند تا تمام کارها به عهده خودش باشد. در این شیوه دیگر نیازی نیست

برای احراز هویت پسورد وارد کنیم بلکه پسورد ما کلیدی است که بر روی سرور قرار داده داریم. در زیر شیوه این کار به طور کامل توضیح داده شده است.

## پیکربندی SSH Client و تولید کلید

در اینجا به بررسی مرحله‌ای که جهت تبدیل کردن یک کامپیوتر به سرویس گیرنده SSH، بدون درخواست کلمه عبور انجام میشود می پردازیم دستورات و مرحله‌ای که در ادامه شرح داده می شود بر روی سیستم کلاینت اعمال می گردد.

در ssh برای تولید کلید از دو الگوریتم rsa و dsa استفاده می شود. از الگوریتم dsa برای امضاء دیجیتال استفاده می شود اما rsa هم برای امضاء دیجیتال و هم برای رمزنگاری کاربرد دارد. dsa سریع تر بوده ، ولی امنیت کم تری دارد ولی rsa کندتر بوده و نسبت به dsa از امنیت بیشتری برخوردار می باشد. الگوریتم پیش فرض برای تولید کلید ، الگوریتم rsa است .

**Step one** ابتدا در کلاینت لینوکس یک جفت کلید رمزنگاری SSH که همان کلیدهای pub و priv هستند را برای حساب کاری که قرار است از آن جهت کپی کردن فایل ها استفاده شود، ایجاد می کنیم. این کار توسط فرمان ----- ssh-keygen صورت می گیرد که نحوه انجام آن در زیر نشان داده شده است. دقت کنید زمانی که درخواست وارد کردن یک کلمه عبور از شما می شود تنها کلید Enter را فشار دهید، و هیچ کلمه ای را وارد نکنید ، البته با سوئیچ -p می توان پسورد آن را بعد از تولید کلید عوض کرد. دقت کنید در هر مسیری که باشید کلید تولید شده در همانجا ذخیره می شود، توصیه می شود کلیدها را در پوشه ssh ذخیره کنید :

```
#cd /home/skywan13/.ssh
#ssh-keygen -b 2048
# ssh-keygen
Generating public/private dsa key pair
Enter file in which to save the key:(root/.ssh/id_dsa/)
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in.root/.ssh/id_rsa
Your public key has been saved in.root/.ssh/id_rsa.pub/
The key fingerprint is: e:73:59:83:96:93:4a:50:33:aa1
```

**Step tow** بعد از اتمام مراحل کار فایل های مربوط به کلیدهای رمزنگاری ساخته شده در پوشه ssh در شاخه خانگی کاربر جاری ذخیره می شوند. فایل id\_dsa.pub به عنوان کلید عمومی است که با سرویس دهنده مقصد به اشتراک گذاشته می شوند. و فایل id\_rsa به عنوان کلید priv مورد استفاده قرار می گیرد. بعد از اتمام مراحل تولید کلید باید به سرور مورد نظر ، کلید pub ساخته شده را معرفی کنیم. این انتقال باید به صورت امن صورت پذیرد لذا با دستور scp آن را به سرور مورد نظر منتقل می کنیم :

```
#cd /home/skywan13/.ssh
#scp id_rsa.pub h.tohid@192.168.1.1:/home/h.tohid/.ssh
```

**Step tree** حال باید به سرور مقصد بفهمانیم چه طور از این کلید کپی شده استفاده کند. چون سرور هنوز متوجه نمی شود که فایل کپی شده، pubkey ما می باشد. لذا برای مطلع کردن سرور باید محتوای فایل id\_rsa.pub را داخل فایل جدیدی به نام authorized\_keys قرار دهیم. سرور کلیدهای pub یوزرها را فقط از این فایل می خواند.

```
#cat id_rsa.pub > authorized_keys
```

سپس باید در سرور، وارد فایل پیکربندی SSH شده و دو خط زیر را تغییر دهیم. اگر هر دو خطوط yes باشند یعنی یوزر هم از طریق کلید و هم از طریق پسورد اجازه احراز هویت دارد. روش ارتباط با کلید امن تر می باشد و بهتر است خط پسورد مقدار no داشته باشد.

```
#vi /etc/ssh/sshd_config
PubkeyAuthentication yes
PasswordAuthentication no
RSAAuthentication yes
#service sshd restart
```

**نکته مهم:** می توانیم برای کپی به جای استفاده از scp از دستور زیر برای این کار استفاده کنیم که بسیار بهتر است، زیرا زمانی که عمل کپی را انجام می دهد، در سرور فایل کپی شده را با نام authorized\_keys ذخیره کرده و فایل را در دایرکتوری یوزر مورد نظر و در پوشه /ssh قرار می دهد، لذا دیگر نیازی نیست که با ssh به سرور مورد نظر وصل شده و محتویات فایل id\_rsa.pub را درون فایل authorized\_keys قرار دهیم.

```
#ssh-copy-id -i ~/.ssh/id_rsa.pub h.tohid@192.168.1.1
```

می توانیم برای ایجاد امنیت بیشتر در زمانی که کلید را می سازیم یک پسورد اختصاصی به آن بدهیم تا احراز هویت کاربر با دو لایه امنیت صورت پذیرد. اگر هر دو آپشن گفته شده در فایل کانفیگ yes باشند، 3 مرتبه اول پسورد یوزر را می پرسد و اگر اشتباه تایپ شود برای احراز هویت پسورد کلید را می پرسد. با اتمام این کار، از این به بعد سرور از بابت هر کانکشنی که بر مبنای ssh کار کند از یوزر مربوطه پسورد نمی خواهد زیرا احراز هویت بر اساس کلید انجام میشود.

در زیر مروری دوباره ای خواهیم داشت بر دستورات تایپ شده در ssh Client :

#### Local System:

```
#cd /home/skywan13/.ssh
#ssh-keygen -t rsa -b 2048
#scp id_rsa.pub h.tohid@192.168.1.1:/home/h.tohid/.ssh
#ssh h.tohid@192.168.1.1
or
#ssh-copy-id -i ~/.ssh/id_rsa.pub h.tohid@192.168.1.1
```

#### Remote System:

```
#cd /home/h.tohid/.ssh
#cat id_rsa.pub > authorized_keys
#vi /etc/ssh/sshd_config
PubkeyAuthentication yes
Password Authentication no
RSAAuthentication yes
```

```
#service sshd restart
```

#### نکات کاربردی :

1. با دستور زیر می توانید پسورد کلید را بر روی سیستم کلاینت Cash کرد :

```
#ssh-add
```

2. این دستور نشان می دهد بر روی سیستم کلاینت چه کلیدهایی موجود است :

```
#ssh-add -l
```

3. و این دستور Cash را پاک میکند :

```
#ssh-add -d
```

4. در صورت برقرار نشدن ارتباط ssh با سرور، از دستور زیر برای رفع مشکل استفاده می کنیم :

```
#ssh -vv h.tohid@192.168.1.1
```

5. کلاینتها می توانند بدون OpenSSH server هم کار کنند. اگر کلاینتها نیازی ندارند به اینکه کلاینتی به آنها

remote login بزنند و فقط می خواهند یک طرفه به منبعی وصل شوید می توانید OpenSSH server را از

روی سیستم پاک کنید :

```
#chkconfig sshd off
```

```
#yum erase openssh-server
```

```
#netstat -ntlp | grep 22
```



Linux Cookie in Persian v.2

## فصل پنجم

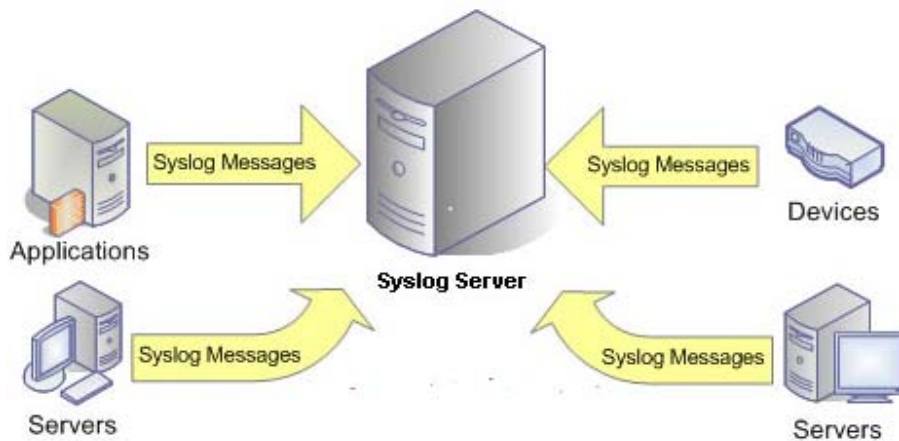
### مدیریت Log و راه اندازی Logrotate



## پیکربندی و مدیریت لاگها با Syslog

مقدمه:

در سیستم عامل لینوکس سرویس ها ، نرم افزار ها و خود هسته در هر لحظه رویداد هایی مانند خطاها و تغییر در روند سرویس یا هر چیزی را ، در غالب فایل هایی متنی ثبت می کند که به این کار Logging یا ثبت رویداد گفته می شود. مدیران این فایل ها را مرتب و در هنگام بروز مشکل و حوادث امنیتی آنها را بررسی می کنند. مستند کردن این فایل ها یکی از وظایفی است که مدیران شبکه در شرکت ها انجام می دهند. هنگامی که یک سرویس Start یا Stop می شود و یا هر تغییر و خطایی رخ می دهد حتی هنگامی که یک عمل با موفقیت انجام می شود یک پیام در فایل Log مرتبط با آن سرویس ثبت خواهد شد. syslog سرور بر روی هر دو پلتفرم ویندوز و لینوکس قابل راه اندازی است.



فایل های Log در مسیر دایرکتوری `/var/log/` قرار دارند و مرتبط با هر سرویس مانند `sshd` یا `dhcpcd` یک فایل Log وجود دارد. لینوکس های ردهت تا ورژن 5 از ابزاری به نام `syslogd` که مخفف `System Log Daemon` است برای ثبت رویداد ها استفاده می کند و از ورژن 6 به بعد از `rsyslog` برای این کار استفاده می کند. این برنامه با سرویس ها و نرم افزار های در ارتباط بوده و آنها رویداد های خود را به این ابزار می دهند. `syslogd` رویداد ها را جمع آوری کرده و در فایل های Log خود آنها ثبت می کند.

فایل های Log فایل های متنی هستند و می توان با دستور های `cat` , `less` , `vim` , `vi` و دستور های `head` , `tail` آنها را مشاهده کرد اما پیشنهاد می کنم از دستور های `head` , `tail` و `less` استفاده کنید. از ابزار های دیگر مرتبط با Log ها در لینوکس نرم افزار `logwatch` است که در بیشتر توزیع های لینوکسی وجود دارد و در دایرکتوری `etc/log.d/` و تحت فایل `logwatch.conf` قابل پیکربندی می باشد.

از اعمال مرتبط با Log ها، `Log Rotate` یا گردش Log است. وقتی که اندازه فایل های Log زیاد می شود بایستی از آنها یک پشتیبان تهیه کرد و یا اینکه دنباله Log کردن را در یک فایل جدید ادامه داد و فایل قدیمی را آرشیو کرد. این

اعمال بصورت خودکار و در غالب Rotate کردن انجام می شود پیکربندی عملیات Rotate به کنترل حجم و بازخوانی ساده تر فایل ها کمک می کند.

بهتر است که در هنگام پارتیشن بندی یک پارتیشن مجزا برای دایرکتوری `var/log/` در نظر بگیریم چونکه رشد اندازه فایل های Log بسیار بالاست و در نظر گرفتن پارتیشن مجزا خارج از دایرکتوری `/` از بروز مشکل جلوگیری می کند. یکی دیگر از موضوعاتی که قابل بحث است ذخیره رویداد ها بصورت محلی و راه دور می باشد. محلی بودن ثبت رویداد کاملاً واضح است و رویداد در خود آن ماشین ذخیره می شوند اما راه دور به معنی است که یک سیستم را بعنوان Log Server انتخاب کرده و تمام ماشین ها رویداد هایشان را به این سرور ارسال کنند. توصیه می شود برای حفظ محرمانگی، داده ها تحت ssh مبادله شوند و بهتر است که ثبت رویداد را هم بصورت محلی (یعنی در خود همان ماشین) و هم بصورت راه دور (یعنی در یک سرور مجزا) انجام دهید.

logging راه دور یک قابلیت امنیتی فوق العاده است. با قراردادن log هایتان در سیستم راه دور، می توانید از رخنه ها و نفوذهای امنیتی که به راحتی می توانند فایل های log را تغییر دهند، جلوگیری کنید.

دو سرویس یا دایمون (daemon) وجود دارد که گزارش گیری را کنترل می کند `klogd` و `syslogd`. `klogd` فقط با پیغامهای کرنل و `syslogd` با دیگر پیغامهای سیستم مانند برنامه های کاربردی سرو کار دارد. شما می توانید رفتار این دو ابزار را با ویرایش فایل `etc/syslog.conf` و فایل تغییر فیچرهای سرویس یعنی `/etc/sysconfig/syslog` پیکربندی کنید.

همچنین می توانید اطلاعات بیشتر را در صفحه راهنمای `etc/syslog.conf` کسب نمایید. هر پیغامی که توسط نرم افزاری تولید می شود اطلاعاتی در مورد محتوای پیغام و مبدأ و تولید کننده آن می دهد. فایل `etc/syslog.conf` به شما امکان می دهد که هر گونه پردازشی را بر روی پیام ها تعیین کنید.

به طور موقت می توانید این اطلاعات را در فایل `message` انبار کنید. همچنین می توانید آنها را در یک فایل سفارشی ذخیره سازید یا می توانید آنها را به یک میزبان (host) راه دور، جایی که میزبان، آنها را مطابق با پیکربندی `syslogd` خودش پردازش خواهد کرد، ارسال نمایید.

## نصب و راه اندازی سرویس Syslog

در بیشتر زمان ها syslog در موقع نصب سیستم نصب می شود و شما نیازی به نصب مجدد ندارید.

ابتدا باید از نصب بودن پکیج syslog اطمینان حاصل کنیم لذا با دستور زیر از سیستم query میگیریم :

```
#rpm -qa | grep syslog
```

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب syslog از yum استفاده می کنیم :

```
#yum -y install syslog
```

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج syslog بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم دوباره query می گیریم :

```
#rpm -qa | grep syslog
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm -ql syslog
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm -qi syslog
```

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
#chkconfig --level 35 syslog on
```

و در انتها سرویس را reset می کنیم :

```
#service syslog restart
```

**نکته مهم :** در لینوکس های redhat base سری 6 به بعد به جای syslog از rsyslog استفاده می شود.

## مبناهای کاری Syslog

syslog بر دو مبنا کار می کند

1. Unix domain socket

2. Internet socket

اگر syslog ، لاگهای سیستم را در سیستم local ذخیره کند بر مبنای Unix domain socket کار می کند و اگر لاگها را از طریق پورت 514 udp به درون سیستم مشخص شده ای در شبکه منتقل شود در مد Internet socket فعالیت می کند.

فایل کانفیگ این سرویس در مسیر `/etc/syslog.conf` قرار دارد. در این فایل یکسری rule یا همان قوانین وجود دارد که از سه قسمت عمده تشکیل شده است :

Facility	severity	where
که در زیر هر کدام از آنها مختصرا توضیح داده شده است.		

(1) **Facility** : مشخص میکند از چه چیزهایی log برداری شود مثل:

**auth** - authentication (login) messages  
**cron** - messages from the memory-resident scheduler  
**daemon** - messages from resident daemons  
**kern** - kernel messages  
**lpr** - printer messages (used by JetDirect cards)  
**mail** - messages from Sendmail  
**user** - messages from user-initiated processes/apps  
**local0-local7** - user-defined (for cisco,servers,...)  
**syslog** - messages from the syslog process itself

اگر بخواهیم log پروسه ای ، به صورت جداگانه در جایی به غیر از facility ثبت شود از **local** استفاده می کنیم. علاوه بر آنکه می توان مشخص کنیم لاگ ها به کجا بروند ، می توانید نوع پیغام هایی که برای سرور لاگ فرستاده می شود را توسط سطوح مختلف severity مشخص کنیم. این سطوح که به آنها level هم گفته می شود استاندارد بوده و براساس شماره و یا حروف اختصاری بکار برده می شوند.

(2) **Severity** : درجه اهمیت ، یا level لاگ را مشخص می کند.

7 - Emergency (emerg)  
 6 - Alerts (alert)      اخطار  
 5 - Critical (crit)      شرایط بحرانی  
 4 - Errors (err)      خطا  
 3 - Warnings (warn)      هشدار  
 2 - Notification (notice)      اطلاعیه  
 1 - Information (info)      اطلاعات بیستر  
 0 - Debug (debug)

در سیستمهای Unix Base درجه اهمیت از صفر الی 7 متغیر است. بالاترین درجه اهمیت کمترین اطلاعات را به ما می دهد و بیشترین اطلاعات را debug اعلام میکند.

اگر تعریف کنیم لاگ برداری از debug شروع شود ، سرور از debug به بالا، همه لاگ ها را ثبت می کند. یعنی از هر کجا تعریف کنیم از آنجا به بالا را لاگ برداری می کند. (از آن سطح و تمام سطوح بالاتر از آن لاگ برداری می کند)

(3) **where** : این قسمت مکان ذخیره سازی فایل های لاگ را مشخص می کند. در اینجا سه متغیر می توانند قرار گیرد :

**tty      file address      /dev/console**

اگر مشخص کنیم که لاگ بر روی /dev/console قرار بگیرد بر روی مانیتور تمام یوزرها قابل رویت می شود.

## سطوح Log ها

رتبه بندی	واژه	شرح
0	emergencies	سیستم عملاً غیرقابل استفاده است
1	alerts	باید سریعاً عکس العمل نشان دهیم
2	critical	شرایط بحرانی می باشد
3	errors	خطائی در سیستم وجود دارد
4	warnings	اخطار...
5	notifications	شرایط عادی ولی مشکلاتی وجود دارد
6	informational	جهت اطلاع....
7	debugging	پیام های مربوط به <b>Debugging</b> سیستم

توضیح : جدول سطوح Log

## تنظیم لاگ بر اساس Unix domain socket

اگر بخواهیم لاگها در سیستم **local** ذخیره شود از شیوه زیر برای نوشتن تنظیمات در فایل کانفیگ **log** بهره می بریم .

**facility.severity**

**where+ log-file-name**

در ادامه برای درک بهتر مفهوم Unix domain socket چند مثال آورده شده است. در اینجا ستاره به معنی همه می باشد.

**\*.info;mail.none;authpriv.none;cron.none**

**/var/log/message**

\*: در اینجا به معنی همه facility ها می باشد.

info: آوردن این کلمه یعنی از info به بالا لاگ بگیرد.

none: اگر بخواهیم از سرویس و یا پروسه ای لاگ بردار نشود از کلمه none استفاده می کنیم. none به معنای منفی شدن است.

**Authpriv.\***

**/var/log/secure**

**Mail.\***

**/var/log/maillog**

**Cron.\***

**/var/log/cron**

ستاره در سه مثال قبل یعنی اینکه لاگ مربوطه تمام severity ها را شامل شود.

**\*.emerg**

**\***

\*. اول این خط یعنی تمام facility ها را شامل می شود و ستاره آخر مشخص می کند خروجی لاگ در **tty** , **file** , **/dev/console** نشان داده شود.

مثال ) به طور پیش فرض لاگ های dhcp در مسیر **/var/log/message** ذخیره می شود. حال می خواهیم کاری کنیم لاگ های dhcp در مسیر **/var/log/dhcp** ذخیره گردد.

ابتدا وارد فایل کانفیگ dhcp شده و عبارت زیر را به آن اضافه می کنیم :

**log-facility local2;**

سپس فایل **/etc/syslog.conf** را باز کرده و مشخص می کنیم لاگهای مربوط به local2 درون چه فایل ذخیره شوند.

**#vi /etc/syslog.conf**

**\*.info; mail.none; authpriv.none; cron.none; local2.none /var/log/message**

**local2.\***

**/var/log/dhcp**



همانطور که از مثال ها مشخص است ، می توان به چندین شکل syslog را جهت نگه داری پیام ها تنظیم کنیم .  
 می توانید با \* تمامی severity ها را مشخص کنید تا در یک فایل ذخیره شوند یا اینکه با مشخص کردن نام آن فقط، آن severity را ذخیره کنید. همچنین می توانید severity های مختلف را در فایل های مختلف ذخیره کنید. توصیه می شود که لاگ ها را بر اساس نیازتان در فایل های مجزا تقسیم بندی کرده تا در آینده آنالیز آنها راحت تر باشد.

## تنظیم لاگ بر اساس Internet socket

برای استفاده از syslog به جهت دریافت لاگ از دستگاه ها و سرورهای دیگر ، می بایست ویژگی Udp logging را در سیستمهای مورد نظر فعال کنیم تا ارسال لاگ بر روی شبکه از طریق پورت 514 پروتکل udp و آدرس تنظیم شده انجام پذیرد.

ابتدا در سرور syslog ، وارد مسیر /etc/sysconfig شده و فایل syslog را edit می کنیم.

```
# vi /etc/sysconfig/syslog
SYSLOG_OPTIONS="m 0 -r "
```

در اینجا ستاره به معنی همه می باشد.

**r:** برای فعال کردن remote UDP logging

**m 0:** برای حذف پیام ها MARK

**X:** برای غیر فعال کردن DNS lookup

برای آنکه مطمئن شویم که UDP logging فعال شده و سرور بر روی پورت 514 به حالت Listen رفته از دستور زیر استفاده می کنیم.

```
# service syslog restart
# netstat -nulp | grep 514
udp 0 0.0.0.0:514 0.0.0.0:*
8621/syslogd
```

حال برای اینکه سیستم های دیگر لاگ خود را به سرور ارسال کنند در سیستمهای ارسال کننده لاگ ، وارد فایل /etc/syslog.conf شده و طبق مثال زیر آدرس سروری که می خواهیم لاگها به آن ارسال شوند را وارد می کنیم.

```
# vi /etc/syslog.conf
*.info;mail.none;authpriv.none;cron.none @10.10.10.1
# service syslog restart
```

توضیح : در قسمت آدرس به جای وارد کردن یک مسیر local آدرس سرور syslog را وارد کنید.

## Log فایل های مهم

در توزیع CentOS و دیگر توزیع ها در زیر دایرکتوری `var/log/` چندین فایل وجود دارد که به مهمترین آنها اشاره می کنیم:

**message**: گزارشهای پیامهای (message logs) هسته سیستم می باشند. این فایل، شامل پیامهای بوت و پیغامهای وضعیت و اجراهای سیستم می باشد. خطاهای IO، شبکه و دیگر خطاهای عمومی سیستم در این فایل گزارش می شوند. سایر اطلاعات از قبیل مواقعی که یک فرد، root میشود نیز در اینجا فهرست می شوند. اگر سرویسهایی مانند سرور DHCP اجرا شوند، فعالیتها را در فایلهای پیغام می توانید مشاهده کنید. فایل `var/log/messages/` معمولاً اولین مکانی است که در مواقع به وجود آمدن دردسر می توانید به آن مراجعه نمایید.

**XFree86.0.log**: این log نتایج آخرین اجرای کارساز Xfree86 Xwindow را نشان می دهد. اگر در بالا آمدن مود گرافیکی دچار مشکل شدید، این فایل معمولاً جوابهایی برای عوامل سوال برانگیز مشکل فراهم می آورد.

**auth.log**: لاگهای مربوط به احراز هویت در این فایل ذخیره می شود.

**kern.log**: این فایل حاوی اطلاعات و رویداد های کرنل سیستم عامل می باشد.

**cron.log**: این فایل حاوی اطلاعات مربوط به این سرویس cron است.

**mail.log**: اطلاعات و رویداد های Mail Server ها و MTA هایی مانند sendmail در این فایل ثبت می شود.

**qmail**: در صورتی که qmail را نصب کرده باشید. رویدادهای این سرویس دهنده میل در این فایل قرار می گیرند.

**httpd**: این فایل مرتبط با وب سرور آپاچی است (در صورتی که httpd را نصب کرده باشید، وجود دارد)

**boot.log**: این فایل مرتبط با اطلاعات و رویداد های فرایند بوت شدن سیستمی می باشد.

**mysqld.log**: این فایل مرتبط با پایگاه داده MySQL می باشد البته در صورتی که MySQL را نصب کرده باشید.

**secure**: حوادث امنیتی سیستم در این فایل ثبت می شوند.

**yum.log**: مختص سیستم های مبتنی بر RedHat و در ارتباط با دستور yum است.

خواندن و مشاهده این فایل ها و حتی استفاده از دستور های خاصی مانند `last` نیاز به دسترسی کاربر ریشه دارد. یعنی یک کاربر عادی نمی تواند این فایل ها را تغییر دهد یا حتی خود مدیر هم شاید نتواند این فایل ها مانند `wtm` را تغییر دهد چون اطلاعات ضروری در آنها ثبت شده است.



## ابزارهای گزارش گیری (logging)

هر گونه ابزار متنی را می توان برای کار با فایل های log به کار برد. در ادامه برخی از این ابزارهای مفید را معرفی نموده ایم:

### • dmesg

برای مرور اجمالی log بوت در آخرین بار بوت شدن سیستم، می توانید از دستور `dmesg` استفاده کنید. خروجی این دستور، عموماً متن طولانی است.

### • tail

برخی اوقات می خواهید فقط یک مرور اجمالی و کوتاه در فایل `log` فعالیتهای در حال وقوع بیندازید. `tail` برای نمایش آخرین خطوط یک فایل متنی طراحی شده است. با افزودن سوییچ `-f`، دستور `tail` به نمایش خروجیهای جدیدی که ناشی از رخ دادن آخرین وقایع است، ادامه می دهد.

### #tail -f /var/log/messages

دستور فوق، آخرین ۱۰ خط فایل `/var/log/messages/` را نشان می دهد، سپس به نظارت در فایل و خروجی هر فعالیت جدید ادامه می دهد. جهت متوقف ساختن دستور فوق، از `Ctrl + C` برای کنسل کردن این فرایند استفاده کنید.

### • more

دستور `more` همان کاری را انجام می دهد که در نگارش DOS انجام می داد. شما می توانید آن را به همراه اسم فایل و نیز برای پایپ کردن اطلاعات در صفحه نمایش استفاده کنید.

### • less

دستور `less` نیز یک مشاهده گر متنی دیگر است که به ما امکان `scroll` در یک فایل و نیز جستجوی اطلاعات در آن را می دهد.

### • logger

ممکن است بخواهید پیغامهای خودتان را در یک فایل `log` قرار دهید. کافی است پیغام `log` را به انتهای فایل متنی، ضمیمه (`append`) کنید. اما مجبور خواهید شد که اطلاعات گزارش را تکرار کنید.

همچنین باید کد خود را در صورت سفارشی بودن سیستم `logging` تغییر دهید. دستور `logger` امکان ارسال پیغامهای شما را به ابزار موجود برای `logging` می دهد. از این دستور در اسکریپتهایی برای تهیه پیغامهایی در مورد نحوه اجرا و خطاها استفاده می شود.

## چرخش لاگها با logrotate

زمانی که سرور تراکنش دیتا بالا و یوزر استفاده کننده زیادی داشته باشد حجم فایل‌های log به مرور می‌تواند خیلی بزرگ شود که این حجم شدن فایل‌های لاگ هم فضای سیستم را اشغال می‌کند و هم واکنشی و خواندن آنها را همراه با تاخیر میکند. لینوکس ابزاری برای چرخش این logها در اختیار دارد که به صورت دوره‌ای لاگهای قدیمی را جابه‌جا و می‌چرخاند. بنابراین اطلاعات log جاری شما با اطلاعات نامربوط قدیمی، ترکیب نمی‌شوند. با این کار حجم لاگها کمتر و مدیریت آنها بهتر می‌شود.

معمولا logrotate به طور خودکار بر اساس یک برنامه زمان بندی اجرا می‌شود. اما به طور دستی نیز قابل تنظیم و اجراست. شما فایل‌هایی در شاخه `var/log/` مشاهده می‌کنید که با یک عدد تمام می‌شوند اینها آرشیوهای دوار (چرخشی) هستند. هنگامی که این سرویس اجرا می‌شود، logrotate، نگارش جاری فایل‌های log را گرفته و یک "۱" به انتهای نام فایل می‌افزاید.

از آن به بعد، ترتیب دیگر فایل‌های چرخش یافته به صورت "۲"، "۳" و غیره خواهد بود. عدد بزرگتر بعد از نام فایل، نشان دهنده گزارشهای جدیدتر میباشد. رفتار خودکار logrotate را با ویرایش فایل `etc/logrotate.conf/` می‌توانید پیکربندی کنید.

## نصب و راه اندازی سرویس logrotate

در بیشتر زمان ها logrotate در موقع نصب سیستم نصب می‌شود و شما نیازی به نصب مجدد ندارید.

ابتدا باید از نصب بودن پکیج logrotate اطمینان حاصل کنیم لذا با دستور زیر از سیستم query میگیریم

```
#rpm -qa | grep logrotate
```

در صورت نصب نبودن، در سیستم های ردهت جهت نصب logrotate از yum استفاده می‌کنیم:

```
#yum -y install logrotate
```

بعد از نصب، باید اطمینان حاصل کنیم که آیا پکیج logrotate بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می‌گیریم:

```
#rpm -qa | grep logrotate
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می‌کنیم:

```
#rpm -ql logrotate
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می‌آوریم:

```
#rpm -qi logrotate
```

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
# chkconfig --level 35 logrotate on
```

**نکته :** logrotate یک سرویس است ولی اسکریپت اجرایی ندارد و خودش فایل ها را چک نمی کند بلکه می رود کار را با cron کامل می کند.

## مهمترین فایل های logrotate

بعد از نصب این سرویس تعدادی مسیر و فایل به سیستم اضافه می شود که سه عدد از مهمترین آنها که کار تنظیم و پیکربندی این سرویس را انجام می دهند به شرح زیر می باشد:

```
/etc/cron.daily/logrotate
/etc/logrotate.conf
/etc/logrotate.d
```

[/etc/cron.daily/logrotate](#)

این فایل ارتباط بین logrotate و سرویس cron را برقرار ساخته و به logrotate می گوید از چه مسیری فایل کانفیگش را بخواند.

[/etc/logrotate.conf](#)

فایل logrotate.conf فایل پیکربندی گلوبال این سرویس است. تنظیمات این فایل به همه اعمال می شود ولی کانفیگ لاگ هر سرویس به تنهایی بر کانفیگ گلوبال ارجحیت دارد. اگر در خود فایل گلوبال و در انتهای آن تنظیماتی برای یک سرویس نوشته شود (داخل کروشه) این بر تنظیمات اصلی ارجحیت اجرایی دارد. در ادامه نمونه ای از یک فایل پیکر بندی آورده شده که بعضی از جزئیات آن را شرح می دهیم:

```
# vi /etc/logrotate.conf
```

```
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# use date as a suffix of the rotated file
dateext
```

```
# uncomment this if you want your log files compressed
compress
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    minsize 1M
    rotate 1
}
/var/log/btmp {
    missingok
:
monthly
    create 0600 root utmp
rotate 1
```

توضیح بعضی از قسمتهای فایل **logrotate.conf**

**# rotate log files weekly**  
**weekly**

این گزینه زمان rotate شدن log فایل ها را مشخص می کند که سه مقدار daily, weekly, monthly می تواند داشته باشد.

**# keep 4 weeks worth of backlogs**  
**rotate 4**

مقدار این خط مشخص می کند تعداد دفعات rotate چند مرتبه باشد.

**# RPM packages drop log rotation information into this directory**  
**include /etc/logrotate.d**

این خط مشخص می کند اطلاعات log های rotate شده به چه مسیری اضافه شود.

**# uncomment this if you want your log files compressed**  
**# compress**

این خط مشخص می کند که آیا لاگهای rotate شده در هنگام ذخیره شدن فشرده شوند. این آپشن به بقیه فایل های کانفیگ لاگ اعمال نمی شود مگر اینکه کانفیگ لاگ سرویسی را درون خود فایل اصلی پیکربندی logrotate بیاوریم.

**size 100k**

اگر بخواهیم به جای هفتگی یا تایی به صورت حجمی عمل rotate انجام شود از این گزینه استفاده می کنیم.

## بررسی /etc/logrotate.d

در زیر این دایرکتوری فایل کانفیگ logrotate سرویس های مختلفی قرار دارد. تمام سرویس هایی که باید از عملکرد آنها لاگ جداگانه تهیه شود در این مسیر یک فایل پیکربندی دارند تا توسط logrotate عمل چرخش لاگ آنها انجام شود. اگر بخواهیم logهای سرویس های مورد نظر در این دایرکتوری فشرده شوند باید فیلد compress را درون هر کدام که می خواهیم اضافه کنیم. یکی از مهمترین فایل های کانفیگی که در این مسیر وجود دارد فایل لاگ سرویس httpd می باشد که جهت آشنایی با گزینه های دیگر کانفیگ logrotate، این فایل را مورد بررسی قرار می دهیم.

```
# vi /etc/logrotate.d/httpd.log
```

```
/var/log/httpd/*log {
size 100k
compress
rotate 5
missingok
notifempty
sharedscripts
postrotate
    Sbin/service httpd reload > /dev/null 2> /dev/null || true
endscript
}
```

نکته مهم: چون rotate این فایل درون خودش نیامده، آن را از فایل global سرویس logrotate خوانده و اجرا می کند. اگر زمان بندی rotate را در این فایل بیاوریم ارجحیت پیدا می کند به زمان بندی که در فایل کانفیگ سرویس درج شده است.

در ادامه به تشریح بعضی از قسمتهای این فایل می پردازیم:

**/var/log/httpd/\*log**: این خط مسیر لاگ فایل ها را مشخص می کند.

**size**: سایز لاگ فایل را مشخص می کند.

**rotate**: تعداد دفعاتی که لاگ فایل قبل از پاک شدن rotate می شود.

**missingok**: یعنی اگر فایل لاگی موجود نبود ایراد نگیرد.

**notifempty**: مشخص می کند اگر فایل لاگ خالی بود آن را rotate نکند.

**postrotate**: این گزینه یک فایل لاگ جدید ساخته و سرویس را reload می کند.

**postscripts**: مشخص می کند بعد از اینکه rotate را انجام داد اسکریپت و یا دستور مورد نظر را انجام دهد.

**prescripts**: مشخص می کند قبل از اینکه rotate را انجام بدهد اسکریپت و یا دستور مورد نظر را اجرا کند .  
**dateext**: این گزینه خیلی مهم و کاربردی است چون در انتهای فایل لاگ تاریخ rotate شدن را درج میکند.  
**Mail**: نتیجه را به آدرس مشخص شده میل می کند.

**مثال**: فایل کانفیگ لاگی بنویسید که اگر حجم فایل مشخصی به 300 بایت رسید آن را rotate کرده و نتیجه را میل و در انتها به جای عدد در نام فایل ها تاریخ را درج کند.  
 ابتدا با دستور dd چند فایل با حجم های متفاوت ایجاد می کنیم.

```
# mkdir /root/logs
# dd if=/dev/zero of=/root/logs/test.log bs=300 count=1
```

سپس یک فایل کانفیگ می نویسیم تا لاگ این فایل را rotate کند.

```
# vi /etc/logrotate.d/test
/root/logs/*.log {
size 100k
compress
rotate 5
mail root@localhost
dateext
}
```

سپس با دستور زیر آن را rotate می کنیم :

```
# logrotate /etc/logrotate.conf
```

### نکته مهم :

در مسیر های /var/log و /var/run دو فایل به نام های wtmp و utmp وجود دارد .

```
/var/log/wtmp
/var/run/utmp
```

این فایل ها باینری هستند و به سادگی خوانده نمی شوند.

wtmp فایل لاگ History Call سیستم است و درون آن اتفاقاتی مثل crash کردن سیستم ، یا اینکه چه یوزری از چه pts ای لاگین کرده است ، ثبت می شود.

utmp هم برای لاگ کردن لاگین های موفق و ناموفق به کار می رود ولی History Call نیست.

دستور last آخرین اطلاعات سیستم را به صورت History Call نمایش می دهد و با دستور lastb می توان Bad login های سیستم را مشاهده کرد.





فصل ششم

اشتراک سیستم فایل با NFS

**NFS**

**NETWORK FILE SYSTEM**

## مقدمه ای بر NFS

NFS که مخفف Network File System است بطور خلاصه امکانیست که اجازه می دهد تا یک سیستم فایل محلی قابل mount شدن توسط دیگر سیستم ها در شبکه مورد استفاده قرار بگیرد. NFS یک روش سنتی برای share کردن دایرکتوری بین سیستم های Unix Base می باشد که به وسیله ی Sun Microsystems در سال ۱۹۸۰ توسعه و به طور پیش فرض بر روی سیستم های Redhat Base نصب است و در توزیع CentOS از ورژن 3 آن استفاده می شود. ورژن 4 از سری 6 لینوکس های Redhat Base به آنها اضافه شد. با کمک NFS ، ما می توانیم اشتراک فایل بین سیستم یونیکس به لینوکس و لینوکس به یونیکس را راه اندازی کنیم. تا پیش از NFSv4 تمامی نسخه های NFS بصورت Stateless بودند، به این معنی که در پروتکل های Stateless ، هر درخواست هیچ اطلاعاتی از درخواست های پیشین ندارد و هر درخواست مستقل از درخواست های قبلی می باشد. انواع متفاوتی از سیستم فایل ها قابل دسترسی هستند که این کار با استفاده از یک سری API انجام می گیرد ، به این api ها Virtual File System گفته می شود.

هدف اصلی از VFS ، اجازه دادن به برنامه های کاربردی کلاینت برای دسترسی به انواع سیستم فایل ها در یک شکل واحد بدون آگاهی از این تفاوت ها میباشد، که در تمامی سیستم های عامل مانند ویندوز، لینوکس، مکینتاش و تمامی شبه یونیکس ها استفاده می شود. یک سیستم فایل NFS که بر روی ماشینی Mount شده، بسیار شبیه به سیستم فایل محلی همان ماشین است .

هدف از VFS دسترسی به سیستم فایل های مختلف بر روی ماشین است. تمام اعمال روی سیستم فایل ماشین محلی که یک سیستم فایل راه دور بر روی آن Mount شده است، از طریق VFS انجام می گیرد.

در ارائه سرویس NFS هیچ محدودیتی در نوع کاربرد سیستم نیست یعنی در یک شرکت با 100 سیستم نباید حتما یکی از آنها بعنوان سرور دهنده NFS باشد و مابقی از آن سرویس بگیرند بلکه هر کدام از این سیستم ها می توانند نقش سرویس دهنده NFS را داشته باشند و دایرکتوری ها و فایل های خود را به اشتراک بگذارند و همزمان به عنوان سرویس گیرنده به فعالیت پردازد.

اما می توان از دیدگاهی دیگر راه اندازی آنرا به دو بخش تنظیم سرویس دهنده و تنظیم سرویس گیرنده تقسیم کرد چون بالاخره یکی دایرکتوری را به اشتراک می گذارد و بقیه استفاده می کنند پس شاید بنا به سیاست یک سیستم هم دایرکتوری را برای دیگران به اشتراک بگذارد و هم از اشتراکات دیگران استفاده کند. تنها مطلبی که باید دقت شود اطمینان از فعال بودن سرویس های لازم در هر دو سمت سرویس دهنده و سرویس گیرنده است که برای کارکرد صحیح سرویس NFS بسیار لازم و ضروری می باشد.

تمامی نسخه های NFS از پروتکل TCP استفاده می کنند. NFSv2 امروزه کمتر استفاده می شود اما NFSv3 و NFSv4 بسیار کاربرد دارند. NFSv4 بواسطه ثابت بودن پورت و تنظیمات فایروال می تواند بر روی اینترنت هم کار کند و همچنین

از ویژگی ACL نیز پشتیبانی می کند. NFSv2 و NFSv3 از پروتکل UDP می توانند استفاده کنند. مشکل اصلی ورژن های 1 و 2 و 3 رندوم بودن پورت های ارتباطی آنها می باشد چون نمی توانیم پورت های مربوطه را بر روی فایروال باز کنیم، که این مشکل در ورژن 4 برطرف گردیده است. فایل [etc/exports](#) بخش اصلی تنظیم NFS است که فهرستی از دایرکتوری های اشتراکی را درون خود نگه می دارد. در این فایل تعیین می شود کدام دایرکتوری ها با چه مجوز هایی باید برای چه کسانی و با چه مجوز هایی به اشتراک گذاشته شوند. به صورت پیش فرض این فایل بدون محتوا بوده و باید توسط کاربر مقدار دهی شود.

در اوایل سرویس NFS بصورت Stateless بودن یعنی توانایی ثبت و نگهداری تاریخچه ای از فعالیت ها و اینکه کدام فایل توسط کدام ماشین ها و کاربران راه دور استفاده شده اند را نداشت. این اطلاعات و تاریخچه ها برای قفل کردن فایل ها لازم می باشند. یک راه حل برای این مشکل بکار بردن ابزار های مجزا از NFS مانند statd و lockd برای قفل کردن فایل هاست که در NFSv2 و NFSv3 استفاده می شوند. اما NFSv4 ویژگی statfull را که نشان دهنده مبتنی بودن بر حالت و نگه داشتن تاریخچه ای از فعالیت هاست را دارا می باشد.

NFSv4 مبتنی بر state است و هم بر روی کلاینت و هم بر روی سرویس دهنده اطلاعاتی را نگه می دارد مانند اینکه کدام فایل ها باز هستند و یا کدام ها قفل شده اند. این اطلاعات در صورتی که سرویس دهنده دچار مشکل شوند برای رفع اشکال سرویس دهنده میان آنها رد و بدل خواهد شد.

## مزایای استفاده از NFS

1. NFS اجازه ی دسترسی محلی به فایل های از راه دور را می دهد.
2. NFS از معماری سرویس گیرنده / سرویس دهنده استاندارد برای به اشتراک گذاری فایل بین همه ی ماشین های مبتنی بر unix استفاده می کند.
3. با NFS، هیچ نیازی نیست که روی هر دو ماشین، سیستم عامل مشابه اجرا شده باشد.
4. با کمک NFS ما می توانیم راه حل های ذخیره سازی مرکزی را پیکربندی کنیم.
5. کاربران می توانند اطلاعات خود را بدون توجه به موقعیت فیزیکی، دریافت کنند.
6. هیچ refresh دستی برای فایل های جدید نیاز نمی شود.
7. نسخه جدید تر NFS همچنین ACL و مانت ریشه ی کاذب را پشتیبانی می کند.
8. می توان با Firewall ها و Kerberos آن را امن تر کرد.

## ویژگی های NFS.v4

1. سازگاری با فایروال ها و ابزار های NAT

2. امنیت بالا

3. بر طرف کردن مشکلات Authentication

4. پشتیبانی از کلاینت های یونیکسی (Linux, BSD, Mac OSX) و ویندوزی

5. پشتیبانی از نوشتن ACL

6. پشتیبانی از نام فایل های یونیکد

7. کارایی بالا حتی بر روی شبکه ها با پهنای باند پایین

8. بر مبنای tcp کار کرده و پورت ثابت آن 2049 می باشد و البته نیاز به نرم افزارهای کمکی مثل Portmapper را ندارد.

9. کلاینت های ویندوزی می توانند از share های nfs استفاده کنند.

10. NFSv4 بر روی شبکه های wan هم کار میکند.

11. مبحث RootDirectory به آن اضافه شده است.

تا ورژن 3 ، سرویس NFS از پورت استاتیک استفاده نمی کرد. نرم افزارهای دیگری مثل Portmapper پیشنهاد دهنده پورت به کلاینت بودند. Portmapper چک می کند NFS با چه پورتهای کار میکند ، به محض اینکه درخواستی از یک کلاینت برسد می آید پورت مربوطه را به کلاینتها پیشنهاد می دهد.

RHEL و دیگر توزیع های لینوکسی ترکیبی از سرویس ها را برای انجام NFS استفاده می کنند. تمامی نسخه های NFS مبتنی بر RPC یا Remote Procedure Call میان کلاینت ها و سرور ها هستند. سرویس های RPC در لینوکس تحت سرویس portmap کار می کنند. در زیر فهرستی از سرویس ها که بصورت ترکیبی با هم در حال ارتباط و کار کردن برای اجرای NFS هستند آمده است:

**nfs** : اصلی ترین سرویس که دیگر سرویس ها با فعال شدن آن نیز فعال می شوند.

**nfslockd** : در سمت کلاینت اجازه می دهد تا فایل ها را بروی سرور قفل یا lock کند.

**portmap** : همانطور که گفته شد سرویس های RPC تحت لینوکس توسط این سرویس کنترل می شوند و مسئول تنظیم کردن اتصال ها برای سرویس های RPC درخواست شده است.

## RPC های مورد استفاده در NFS

وقتی دو کلاینت لینوکسی قصد دارند به منابع اشتراکی هم متصل شوند از پروتکل NFS استفاده می کنند و خود NFS در پس زمینه از RPC ها برای اتصال استفاده میکند. NFS برای ارتباط نیاز به شش نوع RPC دارد. تمام فایل سیستم هایی که تحت شبکه کار می کنند برای عملکرد صحیح خود از سرویس به نام RPC استفاده می کنند. در واقع RPC پروسه ای است که Computing دستورات بین مبدا و مقصد را انجام می دهد تا کلاینت درگیر پروسه ارتباط نشود. فهرست این rpc ها به همراه توضیح در زیر آمده است:

**rpc.mountd** : این سرویس در خواست های mount را از سمت کلاینت در یافت کرده و بررسی می کند که دایرکتوری درخواست شده در حال حاضر به اشتراک گذاشته شده یا خیر. این سرویس عمل mount شدن منبع اشتراکی را انجام می دهد. فهرست دایرکتوری ها صادر شده در فایل `etc/exports` قرار می گیرند. exports شدن به این معنی است که یک دایرکتوری را در شبکه به اشتراک بگذاریم. این سرویس در هنگام فعال کردن سرویس nfs بصورت خودکار فعال می شود.

**rpc.nfsd** : این سرویس با هسته لینوکس برای مواجه شدن با در خواست های پویا از سمت کلاینت کار می کند.

**rpc.nfslock** : به کلاینت اجازه می دهد فایل های خود را بر روی سرور قفل کند اگر فایل مورد استفاده lock نشود ممکن است توسط شخص دیگری مورد تغییر قرار بگیرد. این سرویس در NFSv4 استفاده نمی شود.

**rpc.statd** : کلاینت ها را در صورتی که سرور restart شود باخبر می کند. کار این RPC مانیتور کردن مبداء و مقصد می باشد یعنی اگر هر کدام از طرفین ریوت شوند به طرف مقابل خبر داده می شود تا lock فایل ها برداشته شود. این سرویس توسط nfslock فعال می شود و در NFSv4 استفاده نمی شود.

**rpc.rquotad** : این سرویس اطلاعات سهمیه بندی را برای کاربران راه دور (remote users) فراهم می کند. سهمیه بندی یعنی اینکه هر کاربر راه دور چه مقدار فضا می تواند برای ایجاد فایل و ... داشته باشد. این سرویس به همراه فعال کردن سرویس nfs خودکار فعال می شود.

**rpc.idmapd** : بعنوان نگاشت کننده نام ها و ID ها در NFS است. این سرویس UID ها و GID ها را به نام ها ترجمه می کند. فایل مرتبط با آن `etc/idmapd.conf` می باشد.

**rpc.gssd** : یک پروتکل امنیتی است که در ایجاد ارتباط، Security Conetxt های لازم برای کرنل را load می کند. نکته ضروری دیگر اینکه برای استفاده از NFS بر روی سیستم تان مطمئن شوید که بسته های portmap, nfs-utils و nfs-utils-lib نصب شده باشد که در بخش نصب سرویس آنها را نصب خواهیم کرد. بسته سرور NFS شامل سه سرویس گنجانده شده در بسته های portmap و nfs-utils می باشد.

## نصب و راه اندازی سرویس NFS

ابتدا باید از نصب بودن پکیج NFS اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

```
#rpm -qa | grep nfs
```

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب nfs از yum استفاده می کنیم :

```
#yum -y install nfs-utils
```

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج NFS بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

```
#rpm -qa | grep nfs
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm -ql nfs
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm -qi nfs
```

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
# chkconfig --level 35 nfs on
```

و در انتها سرویس را reset می کنیم :

```
#service nfs restart
```

```
#exportfs -rva
```

### توضیح دستور exportfs :

با این دستور می توانیم لیست export های سیستم را مشاهده کنیم.

r : این سوئیچ export های حذف شده را پاک و یک لیست به روز را نشان میدهد.

v : جزئیات را کاملاً نشان میدهد.

a : export های جدید را نشان می دهد.

برای اینکه بفهمیم چه پورتهایی به این سرویس اختصاص داده شده از این دستور بهر می بریم:

#rpcinfo -p

#showmount -e 192.168.10.1

خروجی این دستور منابع اشتراکی شبکه را نشان می دهد

**نکته:** سرویسی که باعث می شود در هر بار ریست سرویس پورت جدیدی به آن تعلق بگیرد portmapper می باشد که در ورژن 4 هیچ کاربردی ندارد.

## Export دایرکتوری در NFS

### توضیحات سمت سرور:

Export کردن به این معنی است که تعیین کنیم کدام یک از دایرکتوری ها بر روی ماشین محلی برای کدام یک از ماشین ها راه دور و با چه مجوز هایی قابل mount شدن باشد. در NFSv2 و NFSv3 هر دایرکتوری صادر شده بعنوان یک ورودی مستقل بود اما در NFSv4 این طور نمی باشد.

روند کار برای پیاده سازی سرویس NFS بدین گونه است که ابتدا باید سرویس ها و برنامه های لازم را هم بر روی سرویس دهنده (NFS Server) و هم بر روی (NFS Client) نصب کنیم. سپس تعیین کنیم چه دایرکتوری هایی که باید اشتراکی شوند و در نهایت دایرکتوری های اشتراکی شده را در سیستم های کلاینت mount می کنیم مطلب دیگر اینکه لازم است Firewall نیز بصورت مناسب برای اجازه تردد به ترافیک NFS تنظیم شود.

همانطور که گفته شد NFS دارای فایلی به نام exports در زیر دایرکتوری /etc است که باید آنرا ویرایش کنیم.

شکل کلی هر خط این فایل بصورت زیر است:

**shared\_directory IP or machin\_name(OPTIONS)**

که در آن **shared\_directory** نام دایرکتوری از NFS Server است که برای دیگر ماشین های شبکه Share شده است و **IP or machin\_name** آدرس یا نام ماشین (های) کلاینتی است که مجاز به mount کردن دایرکتوری در سیستم محلیشان می باشند و **OPTIONS** هم گزینه هایی هستند که بر شیوه استفاده کلاینت از دایرکتوری اشتراکی تاثیر می گذارند. در زیر تعدادی از مهمترین این Option توضیح داده شده است:

**ro**: مخفف Read Only است که کلاینت های تنظیم شده با این گزینه، تنها دسترسی فقط خواندنی روی دایرکتوری Mount شده دارند.

**rw**: مخفف Read&Write است که کلاینت های تنظیم شده با این گزینه، دسترسی خواندن و نوشتن بر روی دایرکتوری اشتراکی دارند.



**sync** : این گزینه باعث می شود که سرور تنها پس از اینکه نوشتن داده ها (اعمال تغییرات) به طور کامل انجام شد، به کلاینت پاسخ دهد. این گزینه، بعنوان پیشفرض است و اگر هم ننویسید، یاز هم همین گزینه در نظر گرفته می شود.

**async** : نقطه مقابل sync است و یعنی اینکه سرور قبل از تکمیل تغییرات بر روی Storage، به در خواست های دیگر نیز پاسخ خواهد داد و یوزرهای مقابل می تواند از هر مقدار فایل که کپی شده به صورت Real Time استفاده کند.

**نکته** : sync قابلیت اطمینان و امنیت بالاتری دارد اما async دارای کارایی و سرعت بالاتری می باشد.

**no\_root\_squash** : گزینه ای بسیار مهم در تنظیم یک دایرکتوری برای اشتراک گذاشتن است، چون که باعث کاهش امنیتی در سرور NFS خواهد شد. اگر از این گزینه استفاده می کنید باید بدانید که کاربر root روی ماشین کلاینت، روی دایرکتوری اشتراک شده، دسترسی root مطابق با root ماشین سرور را خواهد داشت و توصیه می شود که از این گزینه استفاده نکنید.

**root\_squash** : نقطه مقابل no\_root\_squash است. با استفاده از این گزینه، درخواست های آمده از uid=0 و gid=0 از کاربر anonymous که به nobody user یا nfsnobody user شناخته می شود، نگاشت خواهد شد. یعنی دسترسی کاربر root روی ماشین کلاینت بر روی دایرکتوری Share شده، معادل با دسترسی کاربر root روی ماشین سرور **نخواهد بود** که این کار باعث افزایش امنیت خواهد شد. از این به بعد سطح دسترسی فایل مربوطه nfsnobody می باشد. به طور پیش فرض این آپشن فعال است.

**no\_subtree\_check** : این گزینه باعث عدم پیمایش در دایرکتوری بالا دستی دایرکتوری اشتراکی می شود. یعنی وقتی یک دایرکتوری اشتراکی می شود، با تنظیم این گزینه نمی گذاریم که کلاینت ها به دایرکتوری های بالایی دایرکتوری اشتراکی شده دسترسی پیدا کنند و تنها به زیر دایرکتوری های، دایرکتوری اشتراکی دسترسی دارند.

**نکته** : هر یوزری که با nfs به منبعی وصل می شود، یوزر پسورد پرسیده نمی شود به جای آن وقتی وارد یک سیستم می شود با یوزر nfsnobody به آن وصل می شود به این تکنیک squash گفته می شود.

در زیر چندین مثال از فرمت های گوناگون exports شدن یک منبع آورده شده است:

در مثال زیر دایرکتوری /nfs/nfs-share/ برای یک کلاینت با آدرس 192.168.10.1 به اشتراک گذاشته شده و در خط دوم برای دو ماشین به آدرس های 192.168.10.2 و 192.168.10.1 به اشتراک گذاشته شده است.

```
/nfs-share/ 192.168.10.1(rw,sync)
/nfs-share/ 192.168.10.1(rw,sync) 192.168.10.2 (rw,sync)
```

در مثال زیر یک محدوده IP ها (IP Range) در نظر گرفته میشود.

```
/nfs-share/ 192.168.10.0/24(rw, sync)
```

و در این مثال یک ماشین عضو دامنه skywan13.local را شامل میشود.

```
/nfs-share/ pc2.skywan13.local(rw, sync)
```

در مثال زیر تمامی ماشین های عضو دامنه skywan13.local را شامل میشود.

```
/nfs-share/ *.skywan13.local(rw, sync)
```

برای دیدن مثال های بیشتر به exports 5 man رجوع کنید.

### Example :

فرض کنید می خواهیم دایرکتوری /nfs-share/ را export کنیم تا کاربران شبکه 192.168.10.0/24 با گزینه های ro و root\_squash از آن استفاده کنند.

```
#mkdir /nfs-share
#groupadd nfs-users
#chgrp nfs-users /nfs-share
#chmod g+s /nfs-share
#chmod -R 777 /nfs-share
#vi /etc/exports
    /mnt/nfs-share 192.168.10.10(rw, sync)
#service nfs restart
```

## Export دایرکتوری به سبک NFS v4

در ورژن های قبل از NFSv4 باید مسیر کامل دایرکتوری را هم در فایل exports و هم در خط mount وارد می کردیم یعنی حتما باید از ریشه مسیره می شود. به اینکار rootDirectory گفته می شود. این یک نقص امنیتی است که در ورژن 4 اصلاح گردیده است. جهت بررسی از ورژن 3 و 4 مثالهای ذکر می شود:

### NFS.v3

```
#mkdir -p /mnt/nfs-test
#vi /etc/exports
    /mnt/nfs-test *(ro)
#service nfs restart
#exportfs -rva
#mount -t nfs -o vers=4 192.168.10.1:/mnt/nfs-test
```

### NFS.v4

```
#mkdir -p /mnt/nfs-test
#vi /etc/exports
    /mnt *(ro,fsid=0)
    /mnt/nfs-test *(rw,nohide)
#service nfs restart
#mount -t nfs -o vers=4 192.168.10.1:/nfs-test
```

یا اینکه به این صورت می نویسیم :

```
#mount -t nfs4 192.168.10.1:/nfs-test
```

توضیح :

Fsid=0 : این گزینه نشان می دهد /mnt با NFS ورژن 4 به اشتراک گذاشته شده است.

Nohide : یعنی محتوا را نشان دهد .

## Export دایرکتوری Home

هیچ یوزری دایرکتوری /home واقعی خودش را نباید در اختیار یوزر دیگری قرار بدهد. این کار بهترین روش امنیتی برای share کردن بین دو لینوکس می باشد ، مزیت این کار جلوگیری از rootdirectory است .

```
#mkdir /mnt/home
#mount --bind /home /mnt/home
#vi /etc/exports
    /mnt *(ro,fsid=0)
    /mnt/home *(rw,nohide)
#service nfs restart
```

**--bind** : این آپشن باعث می شود محتوای home اصلی با /mnt/home یکسان شود .

## اختصاص پورت های ثابت به NFS

اگر از NFS زیر ورژن 4 استفاده می کنید می توانید پورتهای آن را ثابت کنید تا دیگر درگیر مشکلات فایروال نشوید. برای این کار به مسیر زیر رفته و فایل nfs را باز میکنیم. فایل nfs فایل اصلی پیکربندی سرویس NFS میباشد. هر زمان که NFS اجرا می شود محتوای این فایل را چک می کند تا اگر تغییری دید آنها را اعمال کند.

**#vi /etc/sysconfig/nfs**

پس از اجرای دستور بالا باید خطوط زیر را Uncomment کنید یعنی علامت # ابتدای خطوط زیر را بردارید

**LOCKD\_TCPPORT=32803**

**LOCKD\_UDPPORT=32769**

**MOUNTD\_PORT=892**

**RQUOTAD\_PORT=875**

**STATD\_PORT=662**

**STATD\_OUTGOING\_PORT=2020**

بعد از ذخیره تغییرات ، سرویس را ریست کرده و دوباره از پورت های NFS یک لیست می گیریم :

**#service nfs restart**

**#rpcinfo -p**

این کار برای کسانی مناسب است که می خواهند فایروال سیستم روشن باشد و این پورتها را درون آن باز کنند. در انتها به منظور پیکربندی iptables خطوط زیر را به فایل /etc/sysconfig/iptables اضافه می کنیم .

**#/etc/sysconfig/iptables**

Firewall configuration written by system-config-firewall #

.Manual customization of this file is not recommended #

filter\*

A INPUT -m state --state NEW -m udp -p udp --dport 2049 -j ACCEPT-

A INPUT -m state --state NEW -m tcp -p tcp --dport 2049 -j ACCEPT-

A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT-

A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT-

A INPUT -m state --state NEW -m udp -p udp --dport 32769 -j ACCEPT-

A INPUT -m state --state NEW -m tcp -p tcp --dport 32803 -j ACCEPT-

A INPUT -m state --state NEW -m udp -p udp --dport 892 -j ACCEPT-

A INPUT -m state --state NEW -m tcp -p tcp --dport 892 -j ACCEPT-

A INPUT -m state --state NEW -m udp -p udp --dport 875 -j ACCEPT-

A INPUT -m state --state NEW -m tcp -p tcp --dport 875 -j ACCEPT-

```

A INPUT -m state --state NEW -m udp -p udp --dport 662 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 662 -j ACCEPT-
[INPUT ACCEPT [0:0:
[FORWARD ACCEPT [0:0:
[OUTPUT ACCEPT [0:0:
A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT-
A INPUT -p icmp -j ACCEPT-
A INPUT -i lo -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT-
A INPUT -j REJECT --reject-with icmp-host-prohibited-
A FORWARD -j REJECT --reject-with icmp-host-prohibited-
COMMIT

```

در آخر هم سرویس iptables را ریست می کنیم .

```
#service iptables restart]
```

## دستورات مهم برای NFS

برخی از دستورات مهم برای NFS

**showmount -e :**

نمایش share های در دسترس روی ماشین محلی

**showmount -e server-ip or hostname :**

لیست share های در دسترس در سرور از راه دور ( remote )

**showmount -d :**

لیست تمام زیر دایرکتوری ها ( sub directorie )

**exportfs -v :**

نمایش یک لیست از فایل های share شده و آپشن های روی یک سرور

**exportfs -a :**

Export همه ی share های لیست شده در /etc/exports با توجه به نام

**exportfs -u :**

Unexport همه‌ی share های لیست شده در `/etc/exports` با توجه به نام

**exportfs -r :**

تازه کردن (Refresh) لیست سرور پس از تغییر `/etc/exports`

## استفاده از دایرکتوری Export شده در کلاینت ها

**سمت کلاینت :**

پس از تمامی کارهای بالا، نوبت به پیکربندی کلاینت ها می رسد. پیش از هر کاری باید در هر کلاینت در مسیر مناسب یک دایرکتوری بسازیم تا دایرکتوری اشتراکی را به آن متصل کنیم. بهترین جا برای اتصال سیستم فایل های اشتراکی دایرکتوری `/mnt` است.

**#cd /mnt**

**#mkdir nfs-share**

سپس به ازای هر خط فایل `exports` باید یک دستور `mount` بصورت زیر اجرا کنیم.

**#mount -t nfs -o vers=4 192.16.10.1:/mnt/nfs-share**

**#mount -t nfs4 192.168.10.1:/mnt/nfs-share** یا به این صورت:

مشکل دستور `mount` موقتی بودن آن است یعنی پس از خاموش شدن سیستم، نقطه اتصال دایرکتوری اشتراکی `mount` شده از بین می رود اگر می خواهید که دایرکتوری مورد نظر بصورت دائمی باشد با یک خط به فایل `etc/fstab/` اضافه کنید .

**#vi /etc/fstab**

192.168.10.1:/mnt/nfs-share /mnt nfs defaults 0 0

با اضافه کردن این خط به فایل `fstab` نقطه اتصال ما به دایرکتوری اشتراکی مورد نظر دائمی می باشد. در صورتی که با خطای زیر در هنگام `mount` کردن یک دایرکتوری در کلاینت مواجه شدید دلیل آن بسته بودن پورت های 111 و 2049 است. بطور کلی برای تست می توانید در سرور و کلاینت فایروال را خاموش یا غیر فعال کنید.

mount.nfs: mount to NFS server '192.168.10.1' failed: System Error: No route to host

سرویس `nfs` بروی پورت 2049 و `portmap` بروی پورت 111 کار می کنند.

## مبحث Autofs

وقتی یک کلاینت دایرکتوری share شده توسط سرور را mount می کند یعنی یک session و کانال ارتباطی شکل گرفته است، حال اگر ارتباط سرویس دهنده قطع شود کلاینت متوجه این موضوع نشده و سیستم درگیر چک کردن مداوم آن می شود. این حالت در کار سیستم کلاینت اختلال ایجاد کرده و کلاً آن را از کار می اندازد و به هیچ وجه اجازه umount شدن را نمی دهد مگر اینکه سیستم را ریست کنیم .

این که ما همیشه یک دایرکتوری را به صورت ثابت mount شده داشته باشیم یک ضعف به حساب می آید چون اگر سرور از کار بیافتد کلاینتها در کارکرد با مشکل مواجه خواهند شد. برای رفع این ضعف سرویس autofs معرفی شد. لذا هر وقت به دایرکتوری خاصی احتیاج داشتیم autofs آن را برایمان mount کرده و بعد از طی شدن زمان خاصی آن دایرکتوری را umount می کند.

لازم است این سرویس بر روی سیستم کلاینت نصب شود چون کلاینت استفاده کننده از دایرکتوری اشتراکی میباشد. با نصب این پکیج در زیر دایرکتوری etc/ تعدادی فایل ایجاد می شود که با auto شروع می شوند :

auto.ftp  
auto.master  
auto.misc

auto.master کانفیگ اصلی و گلوبال سرویس autof در این فایل قرار دارد. در این فایل آدرس دایرکتوری سیستم local را که قرار است چیزی در آن mount شود را مشخص می کنیم.

**#vi /etc/auto.master**

```
/media          /etc/auto.media      ==timeout=20
```

توضیح خطوط تعریف شده :

/media : آدرس دایرکتوری از سیستم local که قرار است چیزی درون آن mount شود.

/etc/auto.media : فایل کانفیگ دایرکتوری که قرار است چیزی درون آن mount شود.

==timeout=20 : این خط مشخص میکند بعد چه مدتی دایرکتوری mount شده umount گردد. (بر حسب ثانیه)

در مرحله دوم باید فایل auto.media را ایجاد کنیم. برای این کار از فایل auto.misc یک کپی با نام auto.media می سازیم.

**#cp /etc/auto.misc /etc/auto.media**

سپس فایل auto.media را باز کرده تغییرات لازم را اعمال می کنیم.

**#vi /etc/auto.media**

nfs-share                      -rw,sync                      192.168.10.1:/mnt/nfs-share

توضیح خطوط تعریف شده :

nfs-share : نام فولدري که قرار است در سیستم کلاینت mount شود. این نام به دلخواه انتخاب می گردد.

-rw,sync : دایرکتوری مورد نظر به چه صورتی mount شود.

192.168.10.1:/mnt/nfs-share: در اینجا آدرس سروری که قرار است از دایرکتوری export شده آن استفاده کنیم را وارد می کنیم.

#service autofs restart بعد از انجام تغییرات ، autofs را ریستارت می کنیم.

ما مشخص کردیم دایرکتوری /mnt/nfs-share از سرور به دایرکتوری /media کلاینت و با نام nfs-share مونت شود. وقتی در کلاینت وارد /media می شویم و ls میگیریم محتویات پیش فرض در آن قرار دارد. ولی به محض اینکه دستور cd nfs-share را تایپ کنیم دایرکتوری /mnt/nfs-share برای ما mount می شود. دقت کنید حتما باید در مسیر مشخص شده باشیم و اگر طبق زمان تعریف شده استفاده ای از دایرکتوری mount شده نداشته باشیم توسط این سرویس umount می گردد.



Linux Cookie in Persian v.2

## فصل هفتم

### زمان بندی فرایندها توسط Cron



## زمان بندی اجرای برنامه ها توسط Cron

### مقدمه :

Cron بر گرفته از chromos یک کلمه یونانی به معنای زمان است که به خدای زمان یونان باستان گفته می شد و در سیستم عامل های خانواده یونیکس برای خودکارسازی انجام دستور ها و پروسس ها از آن استفاده می شود. برنامه ریزی و زمان بندی برای انجام فعالیت های مختلف در لینوکس امر بسیار مهمی هست که شاید خیلی از ما روزانه با آن سر و کار داشته باشیم. برخی اوقات ما دستور یا دستورات مورد نظر خود را به طور مستقیم از شل می خواهیم و انجام میشود اما برخی اوقات نیاز هست تا در زمان ( یا زمان هایی ) مقرر سیستم عامل به طور خودکار برای ما کاری را انجام دهد. برای مثال نیاز داریم تا سیستم برای ما هرروز در ساعتی که مشخص میکنیم بکاپ بگیرد. از مهمترین راه های زمان بندی برنامه ها در سیستم عامل های شبه یونیکس استفاده از نرم افزار cron و دستور at می باشد. زمانی که به صورت یک دوره متناوب بخواهیم فعالیتی انجام شود میتوانید از Crond و Cron Table ها کمک بگیرید.

مدیریت زمان بندی اجرای دستورات توسط این برنامه در قالب یک فایل به نام crontab که بر گرفته از crontable است و معمولا در مسیر etc/crontab قرار دارد انجام می شود. وقتی ما خطی را به فایل crontab اضافه می کنیم ، برای اعمال شدن آن حتما باید سرویس cron ریست شود تا cron مجبور به خواندن فایل پیکربندی خود شده و اسکریپت جدید را در صف اجرا قرار دهد.

همچنین هرکاربر دارای یک فایل شخصی cron است که در مسیر var/spool/cron قرار دارد. پروسه cron هر یک دقیقه یکبار به فایل crontab رجوع کرده و از آن stat می گیرد ، اگر تغییری در این فایل اعمال شده باشد یکبار از اول این فایل را می خواند.

## نصب و راه اندازی سرویس Cron

ابتدا باید از نصب بودن پکیج cron اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم. در سری 6 به بعد CentOS نام این سرویس به cronie تغییر نام یافته است:

```
#rpm -qa | grep cronie
```

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب cron از yum استفاده می کنیم :

```
#yum -y install cronie
```

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج cron بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

```
#rpm -qa | grep cronie
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm -ql cronie
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm -qi cronie
```

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
#chkconfig --level 35 crond on
```

و در انتها سرویس را reset می کنیم :

```
#service crond restart
```

## مسیرها و فایل های اضافه شده به سیستم

با نصب این سرویس چندین مسیر مهم در سیستم اضافه می شود که در زیر مختصرا توضیح داده می شود :

/etc/cron.d

/etc/pam.d/crond

/etc/rc.d/init.d/crond

/etc/sysconfig/crond

/etc/crontab

/usr/bin/crontab

**/etc/cron.d**: در این دایرکتوری محتوایی وجود ندارد. فایل های cron ای که می سازیم در این دایرکتوری قرار میگیرد.

**/etc/pam.d/crond**: مکانیزم احراز هویت cron توسط pam اجرا می شود.

**/etc/rc.d/init.d/crond**: سرویس cron زیر مجموعه init اداره میشود. در این مسیر اسکریپت startup سرویس cron قرار دارد.

**/etc/sysconfig/crond**: اگر بخواهیم سرویس cron با یکسری فیچر خاص استارت شود باید فیچرهای مورد نظر را در این فایل قرار دهیم. مثل debugging mod و یا اجرای سرویس روی یک کارت شبکه خاص .

**/etc/crontab**: در این مسیر فایل اصلی پیکربندی cron قرار دارد.

**/usr/bin/crontab**: و در این جا هم فایل دستور cron قرار گرفته است.

چندین فایل وجود دارد که برای تمامی سیستم بوده و مالک این فایل ها کاربر root است. همه کاربران دارای یک فایل crontab مخصوص خود هستند ولی فایلهای زیر متعلق به کل سیستم می باشند:

- `etc/cron.d/` : دایرکتوری شامل چندین فایل
- `etc/cron.daily/` : زمانبندی برای انجام روزانه
- `etc/cron.hourly/` : زمانبندی بصورت ساعتی
- `etc/cron.monthly/` : زمانبندی ماهانه
- `etc/cron.weekly/` : زمانبندی هفتگی

به طور مثال فایل هایی که در شاخه `/etc/cron.daily/` قرار دارند، بطور روزانه اجرا خواهند شد. در زیر نمونه ای از محتویات این دایرکتوری را مشاهده میکنید:

**# ls -l /etc/cron.daily/**

total 56

```
-rwxr-xr-x 1 root root ۳۱۱ Jun 20 ۲۰۱۰ .anacron
-rwxr-xr-x 1 root root 15399 Apr 20 ۲۰۱۲ apt
-rwxr-xr-x 1 root root ۳۱۴ Mar 30 ۲۰۱۲ aptitude
-rwxr-xr-x 1 root root ۵۰۲ Mar 31 ۲۰۱۲ bsdmainutils
-rwxr-xr-x 1 root root ۲۵۶ Apr 13 ۲۰۱۲ dpkg
-rwxr-xr-x 1 root root ۳۷۲ Oct ۵ ۲۰۱۱ logrotate
-rwxr-xr-x 1 root root ۱۳۶۵ Mar 31 ۲۰۱۲ man-db
-rwxr-xr-x 1 root root ۶۰۶ Aug 17 ۲۰۱۱ mlocate
-rwxr-xr-x 1 root root ۲۴۹ Apr ۹ ۲۰۱۲ passwd
-rwxr-xr-x 1 root root ۳۸۳ Apr 25 ۲۰۱۲ samba
-rwxr-xr-x 1 root root ۲۹۴۷ Apr ۲ ۲۰۱۲ standard
```



## توضیح فایل پیکربندی سرویس Cron

سرویس cron در دو scope کار می کند : (1 Global 2 User Base و به طور کلی دو نوع فایل برای cron مورد استفاده قرار میگیرد User cron table & system cron table .

(1) **Global (system cron table)**: این نوع از کانفیگ مربوط به مدیریت پروسه های سرور است و ربطی به یوزرها نداشته و فقط در اختیار یوزر root می باشد. فایل پیکربندی گلوبال cron در مسیر `/etc/crontab` و فایل های سیستم نیز برای زمانبندی فعالیت ها در `/etc/cron.d` قرار دارد. نوشتن این نوع از تنظیمات فقط در اختیار یوزر root می باشد و یوزرهای عادی فقط اجازه دیدن فایل ها را دارند.

(2) **User Base (user cron table)**: ولی این نوع از کانفیگ را یوزرهای عادی سیستم برای انجام کارهای شخصی خودشان انجام می دهند. فایل های یوزر در آدرس `/var/spool/cron` میباشد و همانطور که از نام آن پیداست این فایل توسط کاربران ایجاد میشود .

توضیحات نوشتن cron در بخش بعدی آمده است. فایل `crontab` ، فایل اصلی پیکربندی این سرویس می باشد تنظیمات global از طریق این فایل به سیستم اعمال می گردد. در ادامه به تشریح خطوط مهم این فایل می پردازیم :

**#vi /etc/crontab**

```
SHELL = /bin/bash
PATH = sbin/bin: /usr/sbin: /usr/bin
MAILTO = root
HOME = /
* * * * * root run-parts /etc/cron.daly
```

**SHELL = /bin/bash**: این مشخص می کند اسکریپتهای که می خواهیم با cron اجرا کنیم با چه شلی اجرا شوند.

**PATH = sbin/bin: /usr/sbin: /usr/bin**: یک اسکریپت مجموعه ای از دستورات است که با هم ترکیب شده اند. این خط مشخص می کند دستورات داخل اسکریپت از چه مسیرهایی اجرا شوند. اگر مسیر دستورات داخل اسکریپت در اینجا درج نشود آن بخش از اسکریپت و یا همه آن کار نمی کند.

**MAILTO = root**: به طور پیش فرض سرویس cron خروجی کار خود را بر روی صفحه نمایش نشان نمی دهد بلکه در حالت پیش فرض cron خروجی را برای email کاربر مالک job میفرستد. بهتر است به جای ایمیل شدن خروجی cron ، آن را به یک فایل `redirect` کنید تا دسترسی به آن راحت تر باشد. این دستور در صورتی اجرا خواهد شد که از `/dev/null/` برای `redirect` استفاده نکرده باشید. اگر به آخر فایل مربوطه چیزی اضافه نشد یعنی اینکه خط نوشته شده ایراد دارد.

**HOME = /**: مسیر خانگی cron را نشان می دهد.



**\* \* \* \* \* root run-parts /etc/cron.daly/**

این خط مهمترین قسمت این فایل است که دارای 8 فیلد می باشد:

1. Field 1 (Minuets) 0 – 59

فیلد اول نشان دهنده دقیقه بوده و از 0 تا 59 متغیر است. محدودیت cron در دقیقه است. یعنی حداقل زمان بین دو کار یک دقیقه می باشد و نمی تواند به ثانیه کار کند.

2. Field 2 (Hour) 0 – 23

فیلد دوم نشان دهنده ساعت بوده و از صفر تا 23 متغیر است.

3. Field 3 (Day of Month) 1 – 31

این فیلد نشان دهنده روز از ماه می باشد.

4. Field 4 (Month) 1 – 12

فیلد چهارم به ماه اشاره دارد.

5. Field 5 (Day of Week) 0 – 7

فیلد پنجم مربوط به هفته می باشد. در cron یکشنبه برابر با صفر است و صفر با 7 فرقی ندارد لذا به جای 7 از 6 استفاده می شود.

**نکته:** ستاره (\*) در cron معنی هر می دهد مثل هر ساعت یا هر دقیقه.

6. Field 6 (root)

این فیلد مشخص می کند اسکریپتی که مسیر آن را در اینجا مشخص کرده ایم توسط چه یوزری اجرا می شود.

7. Field 7 (run-parts)

اگر ما تعداد زیادی اسکریپت را درون یک دایرکتوری قرار دهیم و بخواهیم بوسیله cron اجرا شود تنها راه آن استفاده از دستور run-parts می باشد. برای اجرای یک مجموعه اسکریپت در زمان مشخص از run-parts استفاده میکنیم. این دستور اسکریپتهای یک دایرکتوری را به ترتیب حروف الفباء، به صورت تک تک پشت سر هم اجرا می کند. این دستور با پکیج crontabs بر روی سیستم نصب می شود.

دستور `#rpm -qf `which run-parts`` مشخص میکند run-parts مربوط به چه پکجی است.

8. Field 8 (/etc/cron.daily)

فیلد آخر هم اسکریپت اجرایی و مسیر آن را مشخص می کند.

نکته :

(1) در خطوط cron نباید از دبل کوتیشن استفاده کنید.

(2) با دستور زیر می توانیم از زمان آخرین تغییرات یک فایل مطلع شویم.

```
# stat /etc/crontab
```

(3) و با این دستور می توانیم log تغییرات crontab را مشاهده کنیم.

```
# tail -f /var/log/cron
```

برای ویرایش crontab یک کاربر، از طریق کاربر root بدین شکل عمل میکنیم:

```
#crontab -e -u username
```

## استفاده از سرویس Cron

همانطور که گفته شد اسکوپ Global مخصوص یوزر root است و فقط root حق edit آن را دارد. یوزرهای عادی فقط اجازه دیدن فایل Global را دارند حال اگر یوزرهای عادی سیستم بخواهند یک cron تعریف کنند باید از اسکوپ UserBase بهره ببرند. مکان ذخیره سازی فایل های cron یوزرها در آدرس /var/spool/cron/ میباشد و همانطور که از نام آن پیداست این فایل توسط کاربران ایجاد میشود. این دایرکتوری به طور پیش فرض خالی است و cron تعریف شده توسط یوزرها در اینجا ذخیره می شود. یوزر عادی برای تعریف cron باید از دستور crontab -e استفاده کند که editor پیش فرض آن ، vi است.

نکته مهمی که وجود دارد این است که در cron جدید دیگر نیازی نیست که تمام محتویات یک فایل cron را وارد آن کنیم بلکه فقط خط مربوط به اجرای اسکریپت و زمانبندی را وارد می کنیم.

```
# crontab -e
2 17 * * * ls /
```

هر فایل cron ای که در /var/spool/cron/ ذخیره شود با نام یوزر سازنده آن ذخیره می شود و فقط همان یوزر و یوزر root حق خواندن و edit آن را دارند. یوزر root با دستور زیر می تواند cron متعلق به یک یوزر را edit کند.

```
# crontab -u skywan13 -e
```



اگر یوزری بخواهد چند cron داشته باشد باید تمام آنها را در یک فایل نوشته و ذخیره کند. اگر هم فیلدی را به اشتباه در فایل cron وارد کند در هنگام ذخیره به کاربر هشدار داده می شود. چک کردن فایل cron توسط crontab انجام می شود.

**نکته مهم:** یک سری علائم در نوشتن ورودی cron استفاده میشود که سعی شده با مثال توضیح داده شود:

**\*** (ستاره) برای نادیده گرفتن یک فیلد در نظر گرفته شده، یعنی اگر مثلاً در فیلد ساعت بود بدون توجه به این فیلد سر هر ساعت دستور اجرا میشود و یا اگر در فیلد دقیقه بود سر هر دقیقه و....

**,** (کاما) در هر فیلد که احتیاج داریم چندین بار دستور در ساعات مختلف اجرا بشود کاربرد دارد مثلاً در فیلد دقیقه 15,30 به معنای اجرا در دقیقه های 15 و 30 یا در فیلد ساعت 2,9 به معنای اجرا در ساعتهای 9 و 2 میباشد و....

**-** (خط تیره) برای تعیین یک بازه زمانی است مثلاً در فیلد روزهای ماه اگر بخواهیم بین روزهای 8 تا 15 دستور اجرا بشود بدین صورت مینویسیم 8-15 و....

همانطور که پیش تر در بالای همین مطلب ذکر شد، شما می توانید با دستور "crontab -e" یک فایل crontab بسازید. به هر حال ممکن است شما از قبل یک فایل crontab داشته باشید. برای مشخص کردن فایل خود، دستور زیر را وارد می کنیم:

```
crontab -u <username> <crontab file>
```

```
crontab -u skywan13 sky.log
```

سپس دستور زیر را وارد کنید تا فایل crontab کاربر skywan13 با نام crontab در پوشه آن خانگی آن ذخیره شود.

```
crontab -u skywan13 ~/crontab
```

و برای حذف فایل crontab دستور زیر را در cli وارد می کنیم:

```
# crontab -r
```

برای لیست کردن jobهایی که با crontab -e اضافه کردیم میتوانیم با استفاده از دستور زیر cron jobها را لیست شده ببینیم:

```
# crontab -l
```

تا اینجا دیدید که برخلاف ظاهر پیچیده، crontab به آسانی تنظیم میشود و ابزاری کاربردی و مهمی در فرآیند مدیریت سیستم میباشد.

**نکته مهم :** برای استفاده از cron باید از crontab جهت load کردن jobها استفاده کرد که برای این منظور 2 راه پیش رو داریم :

۱- استفاده از crontab -e ، یعنی مستقیماً دستور مربوط رو در crontab مینویسیم.

۲- از طریق فایل یعنی ابتدا یک فایل متنی میسازیم و طبق قوانین توضیح داده شده در بالا ورودی مناسب با شرایط خودتان رو در فایل مورد نظر قرار داده و سپس فایل را Load میکنید.

قبل از Load با استفاده از crontab -l لیست jobهای جاری را تهیه کرده و هر کدام را که لازم داریم در فایل جدید مینویسیم چون با load این فایل تمامی jobهای گذشته پاک خواهند شد. جهت درک بهتر موضوع به مثال زیر توجه کنید:

```
#touch /skywan13/mycrontab
#echo "30 4 * * * ls -s / skywan13/web >> / skywan13/webdirlist.log 2>&1" > /
skywan13/mycrontab
#crontab -u skywan13/ skywan13/mycrontab
#crontab -l
```

با اجرای crontab -l از load شدن فایل اطمینان حاصل میکنیم. توجه داشته باشید که برای افزودن job جدید یا ویرایش jobهای موجود کافی است با ویرایش فایل mycrontab و افزودن دستور مورد نظر در خط جدید و پاک کردن crontab با استفاده از crontab -r فایل را دوباره Load کنیم cron jobهای تحت کاربری که فایل را با آن load کردیم اجرا خواهد شد.

## محدود کردن دسترسی یوزرها برای استفاده از Cron

اگر بخواهیم برای استفاده از cron دسترسی یوزر و یا گروهی از یوزرها را محدود کنیم و یا فقط به بعضی از یوزرها دسترسی بدهیم باید در فایل های cron.allow و cron.deny تغییراتی اعمال کنیم. این دو فایل در زیر شاخه /etc قرار دارند. اگر هم وجود نداشتند مهم نیست چون می توانیم آنها را بسازیم.

اگر فایل cron.allow موجود باشد **حتماً** باید اسامی یوزرهای مجاز در آن وارد شود، در غیر این صورت به هیچ یوزری استفاده از cron داده نمی شود.

اگر هم فایل cron.allow وجود نداشته باشد چک می شود آیا cron.deny وجود دارد یا خیر. اگر موجود باشد و نام یوزری در آن آمده باشد از دسترسی آن به cron جلوگیری می شود. اگر هیچ کدام از این دو فایل وجود نداشته باشد اجازه کار با cron به همه یوزرها داده می شود.

**نکته مهم:** چون ارجحیت اجرا با `cron.allow` است ابتدا این فایل مورد بررسی قرار می گیرد در صورت وجود نداشتن و یا خالی بودن آن، فایل `cron.deny` مورد بررسی قرار می گیرد.

## مثالهای برای Cron

برای درک بهتر، مثال هایی به همراه توضیحات آورده شده است. ابتدا مثالهایی ساده بیان می شود و سپس مثال هایی پیشرفته مورد بررسی قرار می گیرد.

نوشتن فایل `crontab` ممکن است برای اولین بار کمی گیج کننده به نظر برسد. بنابراین در زیر تعدادی مثال ساده به همراه توضیح آورده شده است تا ابتدا با شیوه نوشتن `cron` آشنا شوید:

هر دقیقه اجرا می شوند `* * * * * <command>`

هر ساعت رأس دقیقه ۳۰ ام اجرا می شوند `30 * * * * <command>`

هر روز ساعت ۶:۴۵ صبح اجرا می شوند `45 6 * * * <command>`

هر روز صبح ساعت ۶:۴۵ بعد از ظهر اجرا می شوند `45 18 * * * <command>`

هر یکشنبه ساعت ۱ صبح (بامداد؟) اجرا می شوند `00 1 * * 0 <command>`

هر یکشنبه ساعت ۱ بامداد اجرا می شوند `00 1 * * 7 <command>`

هر یکشنبه ساعت ۱ بامداد اجرا می شوند `00 1 * * Sun <command>`

اولین روز هر ماه ساعت ۸:۳۰ `30 8 1 * * <command>`

ماه پنجم هرروز در ساعت ۵ هر دقیقه یکبار `* 5 * 5 <command>`

روز اول ماه اول سال `0 0 1 1 * <command>`

هر پنج دقیقه راس هر ساعت `<command> * * * * */5`

از دقیقه 5 هر 15 دقیقه به 15 دقیقه `<command> * * * * */5`

از روز سوم، 3 روز به 3 روز `<command> * * 3/3 * *`

هر روز، هر ساعت دقیقه 20 و 50 `<command> * * * 20,50`

دقیقه 5، هر 3 ساعت یکبار **روزهای اداری 1-5** `* * 5 */3`

هر روز بین 5 الی 10 صبح `<command> * * * 5,6,7,8,9,10`

برای اجرای برنامه ها در startup از طریق crontab، در زمان بوت سیستم کافیت برنامه مورد نظر را بدین طریق در crontab قرار دهیم:

`@reboot /path/to/my/program @reboot updatedb`

این برنامه در هر بار بوت مجدد سیستم اجرا خواهد شد. در ادامه مثالهایی برای درک بهتر موضوع آورده شده است:

`<command> @reboot` هنگام بوت سیستم اجرا می شود

`<command> @yearly` هر سال اجرا می شود `[* * * * 0]`

`<command> @annually` هر سال اجرا می شود `[* * * * 0]`

`<command> @monthly` هر ماه اجرا می شود `[* * * * 0]`

`<command> @weekly` هر هفته اجرا می شود `[* * * 0 0]`

`<command> @daily` هر روز اجرا می شود `[* * * * 0]`

`<command> @midnight` هر روز اجرا می شود `[* * * * 0]`

<command> @hourly هر ساعت اجرا می شود [0 \* \* \* \*]

برای اجرای چندین دستور پی در پی، آنها را با استفاده از "&&" به صورت پی در پی بنویسید. مثال زیر ابتدا دستور command\_01 و سپس دستور command\_02 را در هر روز اجرا می کند :

@daily <command\_01> && <command\_02>

### مثال های پیشرفته :

1) در مثال زیر در ساعت 3:12 دقیقه صبح هر روز از هر ماه، cron با اجرای این خط شروع به تهیه پشتیبان از /etc/ میکند. `dev/null 2>&1` به معنی ارسال هرگونه خروجی استاندارد به `dev/null` که سطل آشغال لینوکس است و هدایت خطاهای استاندارد 2 (standard error) به همان جایی که خروجی استاندارد رفته است بدون هرگونه خروجی در ترمینال یا هر نقطه دیگری از سیستم. اگر بجای `>>` از `>` استفاده کنیم در هر دفعه باز اجرا شدن دستور و فرستادن خروجی به فایل مربوطه، محتویات فایل پاک و خروجی جدید جایگزین میشود ولی با استفاده از `>>` خروجی به انتهای فایل افزوده خواهد شد.

`12 3 * * * root tar cfz /tohid/backup.tar.gz /etc >> /dev/null 2>&1`

2) در این مثال در تاریخ یکشنبه 7 oct ساعت 15:30 از ماه دهم روز 7 پیغام تبریکی برای کاربر ارسال میشود:

`30 15 7 10 0 * root echo "happy birthday,tohid!!"`

3) مثال بالا را به شیوه زیر هم میتوان نوشت، دقت کنید حروف اول روز و یا ماه باید بزرگ نوشته شود:

`30 15 7 Oct Sun * root echo "happy birthday,tohid!!"`

4) اگر می خواهید که کاربر tohid یک دستور را دقیقه 15، بعد از هر ساعت بدون توجه به تاریخ اجرا کند بدین طریق عمل میکنیم:

`15 * * * * tohid echo "I'm skywan13 Remember"`

5) یا اگر تمایل دارید هر 15 دقیقه اجرا بشود از این خط استفاده میکنیم :

`*/15 * * * * tohid echo "I'm skywan13 Remember"`

6) برای اجرا شدن یک دستور هر 2 ساعت یعنی در 2 صبح، 4 صبح، 6 صبح و... 12 ظهر، 2 بعد از ظهر، 4 بعد از ظهر و... از این خط بهره می بریم :

```
0 */2 * * * tohid echo "I'm skywan13 Remember"
```

(7) با افزودن '، ' در یک فیلد امکان چندین مرتبه اجرا شدن دستور مورد نظر بدست می آید. مثلاً برای اجرای دستور در ۱۵ و ۳۰ دقیقه گذشته از هر ساعت بدین طریق عمل میکنیم:

```
15,30 * * * * tohid echo "I'm skywan13 Remember"
```

(8) برای اجرای دستور مورد نظر در یک زمان مشخص، برای اولین هفته ماهی که شما تمایل دارید در فیلد روز از ۱-۷ استفاده میکنیم (خط تیره به معنی تا و یا الی می باشد). در این خط مشخص کرده ایم در ۱۵ و ۳۰ دقیقه گذشته هر دو ساعت از روز ۱ تا ۷ این خط اجرا شود:

```
15,30 */2 1-7 * * tohid echo "I'm skywan13 Remember"
```

(9) خط زیر هر ۲ دقیقه به ۲ دقیقه به صفحه ایندکس یک وب سرور وصل شده و آن را دانلود کرده و سپس در دایرکتوری کاربر ذخیره می کند:

```
*/2 * * * wget http://192.168.1.10/index.php >> /home/skywan13/cron
```

(10) در ساعت ۱۲:۳۰ هر روز فایل های خالی دایرکتوری tmp/ را پاک می کند. دستور find برای اجرا شدن توسط crond از مجوز های کاربر root استفاده می کند. باید ابتدا دستور -e crontab را اجرا کنید تا فایل crontab شما برای ویرایش باز شود:

```
30 0 * * * root find/tmp -type f -empty -delete
```

(11) با دستور زیر در ۱۰ ژوئن ساعت ۸:۳۰ صبح یک backup توسط اسکریپتی گرفته می شود. توجه کنید برای ساعت ۸:۳۰ شب باید از ۲۰:۳۰ استفاده کنید.

```
30 8 10 06 * root /home/skywan13/full-backup
```

برای گرفتن backup در ساعت ۱۱ ظهر و ۴ بعد از ظهر (۱۶) هر روز از دستور زیر استفاده کنید:

```
00 11,16 * * * /home/skywan13/bin/incremental-backup
```

(12) برای انجام در روز های خاص از هفته مثل روز دوم هفته (یکشنبه روز اول هفته میلادی، عددش ۰ است و دوشنبه روز دوم، عددش ۱ است) تا روز ششم یعنی جمعه هر هفته بین ساعت های ۹ صبح تا ۶ عصر انجام می شود:

```
00 09-18 * * 1-5 /home/skywan13/bin/check-db-status
```

(13) برای اجرای وظایف در یک محدوده زمانی خاص مثل بین ساعت 9 صبح تا 6 عصر (18) از دستور زیر استفاده میکنیم:

**00 09-18 \* \* \* /home/skywan13/bin/check-db-status**

(14) می خواهیم اسکریپتی به نام clean.cache که cache سیستم را پاک می کند، هر 10 روز یکبار اجرا شود. لذا فایل اسکریپت مربوطه را در مسیر /etc/cron.daily/ قرار می دهیم. محتوای اسکریپت به شرح زیر می باشد:

```
#!/bin/bash
# A sample shell script to clean cached file from lighttpd web server
CROOT="/tmp/cachelighttpd/"
DAYS=10
USER="lighttpd"
LGROUP="lighttpd"
# start cleaning
/usr/bin/find ${CROOT} -type f -mtime +${DAYS} | xargs -r /bin/rm
# if directory deleted by some other script just get it back
if [ ! -d $CROOT ]
then
/bin/mkdir -p $CROOT
/bin/chown ${USER}.${LGROUP} ${CROOT}
fi
```

سپس برای اجرایی کردن اسکریپت از خطوط زیر استفاده می کنیم:

```
# crontab -l > /backup/cron/cronjobs.bakup
# crontab -u username -l > /backup/cron/cronjobs_username.bakup
```

## اجرا برنامه ها با واسط گرافیکی کاربر

برنامه هایی که کاربر می خواهد اجرا کند به دو صورت می باشند :

برنامه هایی که دارای محیط گرافیکی کاربر هستند (GUI) و نیازمند تعامل با سرویس دهنده پنجره X هستند مانند مرورگر Firefox و برنامه هایی که بدون GUI هستند که خروجی و ورودی این برنامه ها در پوسته خط فرمان است و برای اجرا شدن نیازی به تعامل با سرویس دهنده پنجره X ندارند.

برنامه هایی که در دسته دوم (CLI) قرار می گیرند بدون هیچ مشکلی به وسیله Cron اجرا می گردند. اما برنامه های دسته اول که دارای GUI هستند فقط با نوشتن دستور مورد نظر اجرا نخواهند شد. قبل از دستور باید به سرویس دهنده X بگویید که برنامه در کدام صفحه نمایش برای شما اجرا شود.

برای این منظور قبل از دستور مورد نظر از `env DISPLAY=:0` استفاده می کنیم. به عنوان مثال برای اجرای برنامه gedit در ساعت 10:25 هر روز صبح در فایل crontab خط زیر را وارد می کنیم :

```
25 10 * * * env DISPLAY=:0 /usr/bin/gedit
```

`DISPLAY=:0` به Cron می گوید که برنامه در صفحه جاری (desktop) اجرا شود و اگر دارای چندین صفحه نمایش هستید از دستور زیر استفاده کنید.

`DISPLAY=:0.0` به Cron می گوید که برنامه در صفحه نمایش اول و در صفحه جاری اجرا شود. این دستور حتما باید قبل از اجرا شدن یک برنامه GUI به وسیله Cron اجرا شود. شما می توانید این دستور را در قسمت Startup Applications قرار دهید تا هنگام بوت شدن سیستم اجرا شود.

```
25 10 * * * env DISPLAY=:0.0 /usr/bin/gedit
```

## زمانبندی اجرای فرامین توسط Anacron

**anacron** را میتوان برای اجرای دستورات بصورت دوره ای استفاده کرد که این فواصل به روز تعیین می شوند. برخلاف cron، تصور نمی کند که دستگاه 24 ساعت بطور مستمر روشن است. از این رو آن را میتوان در دستگاههایی که 24 ساعت روشن نیستند برای کنترل jobهای روزانه، هفتگی و ماهانه که بطور معمول با cron صورت میگیرد استفاده کرد. بنابراین در anacron موضوع اصلی اجرا شدن jobهاست نه اجرا شدن آنها سر ساعت و دقیقه تعیین شده همانند cron.

cron-daily در لینوکس CentOS در ساعت چهار و دو دقیقه اجرا می شود حال فرض کنید سیستم راس ساعت 4 خاموش شود و در ساعت 5 مجددا UP شود و این مدت 23 ساعتی که باید به چهار و دو دقیقه بعدی برسد ممکن است در سیستم کلی اتفاق رخ دهد که ما در این صورت یک روز کامل را از دست داده ایم. یا فرض کنید سیستمی داریم که باید 5



روز به 5 روز از آن بکاپ بگیریم. اگر 5 روز اول را از دست بدهیم و بکاپ دوم را با cron بگیریم در اصل ده روز را از دست داده ایم. این خلاء یک نقص برای سیستم و سرویس cron محسوب می شود. این ضعف با سرویسی به نام anacron پوشش داده می شود.

کار anacron اجرای cron ای است که از دست رفته و زمان اجرای آن گذشته و اجرا نشده است. anacron برای سیستم های است که UP و DOWN زیادی دارند مثل سیستم های خانگی و یا کلاینت های تحت شبکه. محدودیت cron به دقیقه بود اما anacron محدود به روز است. یعنی اگر cron ساعتی یک بار کاری را انجام ندهد anacron یک روز بعد شروع به انجام آن می کند. این سرویس مناسب سرور نیست چون محدودیت زمانی آن به روز می باشد. برای استفاده از anacron می بایست بسته ی مربوط نصب و سپس سرویس anacron را اجرا کنیم.

```
# yum -y install cronie-anacron
# rpm -qa | grep cronie-anacron
# rpm -qi cronie-anacron
```

## تنظیم کردن وظایف Anacron

(tasks) وظایف anacron در فایل `/etc/anacrontab` لیست شده است. علاوه بر این ، در دایرکتوری `/var/spool/anacron` هم برای خودش مانند cron یک دایرکتوری مستقل دارد. محتویات فایل `/etc/anacron` تقریباً شبیه فایل کانفیگ cron است اما تفاوت هایی هم دارد. خطوط اصلی و متفاوت این فایل در انتهای آن قرار دارد که حاوی 4 فیلد اصلی می باشد. هر خط در این فایل تنظیمات مربوط به یک وظیفه است و بدین ترتیب نوشته شده اند :

Period	Delay	Job-identifier	Command
1	65	cron.daly	ran-parts /etc/.....

**: period**

عدد روزهایی که باید بین اجرای دستورات طی بشود مثلاً 9 ، یعنی دستور هر 9 روز یکبار اجرا میشود یا 7 برای اجرای هفتگی است. این فیلد بر مبنای روز می باشد.

**: delay**

برای هر job , anacron بررسی میکند که آیا این دستور در  $n(n=period)$  روز گذشته اجرا شده است یا نه. اگر نشده باشد anacron آن را اجرا میکند. در اصل این فیلد زمانی است که سیستم بعد از UP شدن شروع به انجام کار عقب افتاده می کند. مبنای زمانی این فیلد دقیقه می باشد.

**: Job-identifier**

آخرین زمانی که anacron یک کاری را انجام داده است در این فایل ثبت می شود.

### command :

دستوراتی که خواهان اجرای آن هستیم.

خطوط زیر در فایل اصلی کانفیگ anacron آمده که به ترتیب توضیح داده می شود.

```
1 65 cron.daily
7 70 cron.weekly
30 75 cron.monthly
```

**1 65 cron.daily**: این خط می گوید یک روز به یک روز که سیستم up می شود 65 دقیقه صبر کند سپس این فایل را بررسی کند، اگر روز قبل این کار انجام شده باشد که هیچ، در غیر این صورت باید job مورد نظر انجام شود.

**7 70 cron.weekly**: این خط بیان می کند هر 7 روز به 7 روز که سیستم up شد 70 دقیقه صبر کند و job مورد نظر را در صورت انجام نشدن انجام دهد.

**30 75 cron.monthly**: این خط هم می گوید هر 30 روز یکبار که سیستم بالا آمد 75 دقیقه صبر کند بعد job مورد نظر را انجام می دهد حال فرض کنید 15 روز از اول ماه گذشته و سیستم را up می شود anacron توسط این خط بررسی می کند چون 15 روز به سر ماه بعدی مانده، job مربوطه را اجرا نمی کند.

متغیرهای محیطی همچون SHELL و PATH را در بالای فایل /etc/anacrontab/ می توانید تنظیم کنید چنانکه در cron هم تنظیم می کردیم.

برای period می توانید هم از اعداد برای نشان دادن دوره ی اجرای دستور استفاده کنید هم از نشانه هایی مانند زیر :

```
@daily
@weekly
@monthly
```

در این مثال هر روز با تاخیر یک دقیقه ای جمله ی hi,how are u today به انتهای فایل home/skywan13/hi افزوده میشود.

**@daily 1 skywan13 echo "hi,how are u today?" >> home/skywan13/hi**

**نکته**: این تایمی که بر اساس ساعت آمده به این علت است که سیستم بعد از up شدن به یک پایداری برسد.

**نکته مهم**: این سرویس به صورت پیش فرض stop است و اگر فعال شود هر روز علاوه بر کارهای خودش وظایف cron را هم انجام می دهد، آنوقت است که بین cron و anacron تضاد کاری پیش می آید. اگر anacron زودتر از cron اجرا شود، cron متوجه نمی شود و هر دو job مورد نظر را انجام می دهند.

وقتی anacron یک job را برای دفعه اول اجرا میکند فایلی به نام job-identifer را در /var/spool/anacron/ میسازد که محتوای فایل تاریخ اجرای job است (نه ساعت). اصطلاحا به این فایل timestamp گفته میشود. بعد از اجرای مجدد این job دوباره با تاریخ جدید بازنویسی می شود.

کاربر عادی در حالت پیش فرض قادر به استفاده از anacron نیست به یک دلیل ساده ، چونکه اجازه ساخت فایل timestamp را در دایرکتوری /var/spool/anacron/ ندارد. برای حل این مشکل بدون اینکه مشکل جدیدتری بوجود آید بدین ترتیب عمل میکنیم :

1- یک گروه میسازیم و کاربرها را به آن اضافه میکنیم:

برای انجام اینکار از groupadd یا addgroup میتوانید استفاده کنید :

```
# groupadd anacron
or
# addgroup anacron
```

حالا شما یک گروه بدون کاربر داریم که باید کاربران مورد نظرتون را به این گروه اضافه نمایید:

```
#adduser skywan13 anacron
```

2- مجوز مالکیت /var/spool/anacron را تغییر میدهیم برای تغییر مالکیت حتما باید با کاربر root وارد شده باشید:

```
# chown root.anacron /var/spool/anacron
# chmod g+w /var/spool/anacron
```

خوب حالا شما عضو گروه anacron هستید و مجوز نوشتن را در دایرکتوری مربوطه دارید.

3- برای ادامه کار باید فایل anacron مربوط به خودتان را ایجاد کنید:

فایل anacrontab را به یک جایی در دایرکتوری خانگی خودتان مثلا /home/skywan13/anacron/ کپی کنید و طبق آموزش های بالا فایل را تنظیم نمائید.

4- anacron یک daemon نیست و فقط هنگام بالا آمدن سیستم برای کاربر root اجرا میشود پس باید برای خودتان هم آن را اجرا کنید. بدین ترتیب عمل میکنیم:

```
# echo anacron -t $HOME/etc/anacrontab >> .bashrc
```

```
# echo anacron -t $HOME/etc/anacrontab >> .bash_profile
```

## زمان‌بندی دستورات با at

خیلی وقت‌ها شده که بخواهیم یک دستور را برای اجرا در زمانی خاص زمان‌بندی کنیم. مثلاً ممکن است در ساعاتی از شب دریافت فایل از اینترنت رایگان باشد ولی در آن ساعات خواب باشیم و بیدار ماندن سخت! برای حل این مشکل در ویندوز IDM داشتیم، در لینوکس چه کنیم؟

در این جا نرم‌افزاری رو به شما معرفی می‌کنم به نام **at** که برای برنامه‌ریزی کردن دستورات است و کار با آن نیز بسیار ساده است. **at** تقریباً در همه توزیع‌های لینوکس نصب است. **at** فایل یا اسکریپت را اجرا نمی‌کند بلکه فقط توانایی اجرای یک دستور، در یک زمان خاص را دارد و البته دوره زمانی هم ندارد. خروجی دستور **at** به یوزر استفاده‌کننده میل می‌شود و با ریست سیستم هم خط‌نوشته شده **at** از بین می‌رود.

### نحوه نوشتن خط at :

```
at time date
# at now +1min
at> ls /etc
```

### یک مثال ساده

دستورات زیر را در ترمینال وارد کنید:

```
# at 2:00
at> echo \"Hello World!\" >> /home/$USER/log
```

و سپس **Ctrl + D** بزنید.

مثال بالا از **at** می‌خواهد که دستور خط دوم را که خود عبارت **Hello World!** را در لاگ کاربر کپی کند و در ساعت ۲ صبح اجرا کند. زدن **Ctrl + D** بعد از وارد کردن خط دوم، به **at** پایان وارد کردن لیست کارها را اعلام می‌کند. برای این که **at** بتواند دستورات را اجرا کند باید **Daemon** آن در حال اجرا باشد:

### # service atd start

همان‌جور که مشخص است با **at** می‌توان تمامی کارها را زمان‌بندی کرد. فرض کنید لیستی از فایل‌ها برای دانلود دارید و می‌خواهید دانلود ساعت ۲ صبح شروع شده و در ساعت ۸ صبح پایان یابد. ابتدا لینک‌های مورد نظر را در فایلی متنی به طوری که هر لینک در یک خط باشد کپی کنید.

در این مثال ما فایل **dl-list.txt** نامیدیم و از نرم‌افزار دانلود **Aria2** برای دانلود کمک گرفتیم:

```
# at 2:00 + 1000 days
at> aria2c -i ~/dl-list.txt -j 1 -x 5
```

و سپس **Ctrl + D** مثال بالا فرمان دانلود را با کمک **at**، به مدت ۱۰۰۰ روز پیاپی در ساعت ۲ صبح اجرا می‌کند.

حالا برای بسته شدن دانلود در ۸ صبح:

# at 8:00 + 1000 days

at> pkill aria2c

بعضی از وبسایت‌ها ممکن است فایل را تنها در اختیار کاربرانی که در آن وبسایت حساب دارند بگذارند مثل Rapidshare که در آن صورت کافیسیت نام کاربری و رمزعبور خود را در قالب اطلاعات درخواست دانلود با aria2 بفرستید:

at> aria2c -i ~/dl-list.txt -j 1 -x 5 --http-user=ali --http-passwd=123456

دستور بالا فایل‌های لیست شده در dl-list.txt را با نام کاربری ali و رمزعبور 123456 دانلود می‌کند.

## جزئیات at

atq لیست دستورات برنامه‌ریزی شده را به همراه شماره آن‌ها نشان می‌دهد.

atrm دستورات برنامه‌ریزی شده را پاک می‌کند:

# atrm que\_id

برای یافتن que\_id دستور مورد نظر، از atq کمک بگیرید.

قالب زمانی به صورت HH:MM وارد می‌شود، استفاده از am و pm هم معتبر است. مثلاً ۸ صبح در مثال بالا را 8 am هم می‌توان نوشت.

تاریخ به صورت [CC]YY-MM-DD باید وارد شود، از مخفف ماه و روزها نیز می‌توان استفاده کرد. عبارت‌هایی مثل فردا، امروز، عصر و نیمه شب هم معتبر هستند.

sun mon tue wed thu fri sat

jan feb mar apr may jun jul aug sep oct nov dec

tomorrow today noon midnight

برای تکرار یک کار در چند روز:

+ N days

چند زمان‌بندی پیچیده‌تر با at :

at 3:00pm tomorrow

at 2:00am jul 5 + 4 days

at 2:00 2012-7-5

at 2:00 wed

نکته:

واحد های زمانی کوچک تر در اول قرار دارند. یعنی مثلاً ساعت و دقیقه قبل از ماه.

aria2 فقط یک نمونه برنامه برای دانلود است؛ Axel, wget و lftp از دیگر مثال ها هستند.

atd همان طور که در بالا گفته شد یکی از فرمان ها at است. برای استفاده از آن میشود از سیلابس هایی جهت نشان دادن دقیق زمان استفاده کرد که به آن ها میپردازیم. همانطور که در قبل اشاره شد از دستور at برای کارهایی که یکبار انجام میشوند استفاده میشود:

**at now**

**at 04:11 am**

**at now +5 min**

**at now +5 hours**

**at now +4 days**

**at now +4 weeks**

**at 13:13 pm October 18**

برای مثال با این فرمان به این صورت میتوان در ۵ دقیقه آینده سیستم را خاموش کرد:

**# at now +5 min**

**at> date > /file1**

پس از اعلان، سیستم از شما اطلاعات مربوط را میگیرد و در زمان تعیین شده کار را انجام میدهد. برای مشاهده لیست کار هایی که توسط این فرمان انجام میشود از فرمان at -l ویا atq استفاده میکنیم.

**# at -l**

2 Sat Aug 24 10:35:00 2013 a Ali

**# atq**

2 Sat Aug 24 10:35:00 2013 a Ali

Linux Cookie in Persian v.2

## فصل هشتم

## بررسی LVM در لینوکس

**LVM**



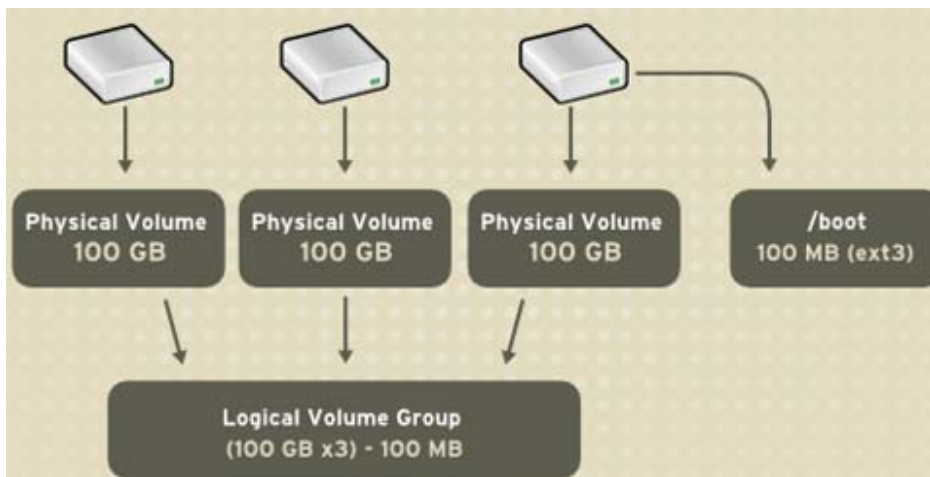
## LVM چیست؟

LVM بر گرفته از Logical Volume Management ابزاری است که امکان مدیریت Disk Drive ها و پارتیشن ها را می دهد و به وسیله آن به راحتی می توان پارتیشن را resize کنید. امروزه در تمامی توزیع های لینوکسی LVM وجود داشته و می توانید در زمان نصب توزیع لینوکسی یا پس از نصب توزیع، از LVM برای مدیریت دیسک ها و پارتیشن ها استفاده کنید. LVM امکان resize کردن پارتیشن، ایجاد Snapshot Backup یا ترکیب چندین دیسک برای اجرا شدن تحت یک پارتیشن واحد و بسیاری دیگر از امکانات را فراهم می کند. به طور کلی LVM انعطاف پذیری بهتری در مدیریت دیسک ها و پارتیشن ها ارائه میکند. بدون وجود LVM، تغییر اندازه یا resize کردن یک پارتیشن کاری دشوار است و ممکن است اطلاعات پارتیشن از بین برود و یا باعث Downtime و از دسترس خارج شدن سیستم شود ولی با استفاده از LVM به راحتی می توان این کار را انجام داد.

در حالت قدیمی مدیریت دیسک ها، سیستم عامل در زیر دایرکتوری `/dev` به دنبال دیسک ها تحت نام هایی مانند `/dev/sda/` و `/dev/sdb/` می گردد.

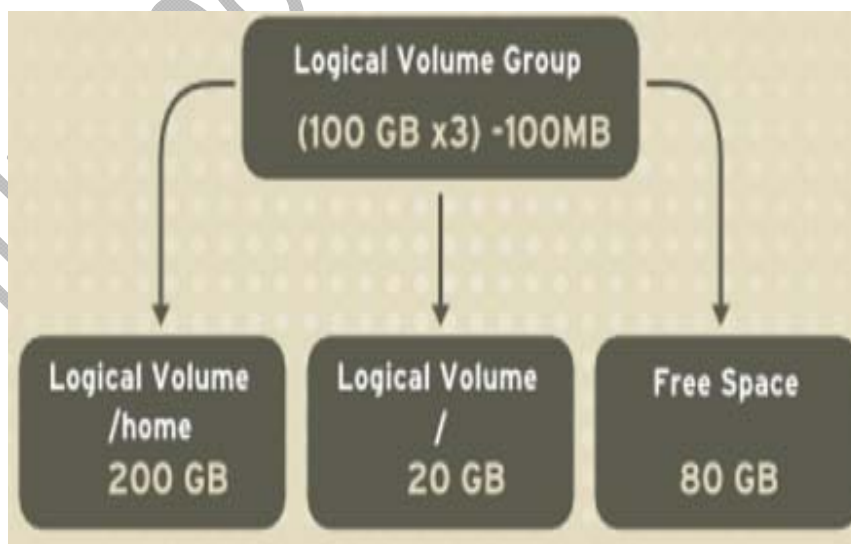
با LVM دیسک ها و پارتیشن ها می توانند شامل دیسک ها و پارتیشن های متعددی در غالب یک دستگاه واحد باشند. LVM به سیستم عامل، در غالب Volume Group (دیسک ها) و Logical Volume (پارتیشن ها) نشان داده می شود. چونکه Volume Group ها و Logical Volume ها وابسته به هارد درایوها نیستند و می توان به راحتی دیسک ها و پارتیشن ها را تغییر اندازه داد یا دیسک و پارتیشن جدید ایجاد کرد. فرض کنید یک Logical Volume به نام `/dev/mapper/VolGroup00-LogVol00` بر روی دایرکتوری ریشه mount شده است و حجم آن به اندازه 50 گیگابایت است و می خواهیم حجم آنرا توسط دیسک دیگری که اندازه 200 گیگابایت دارد افزایش دهیم تا حجم Logical Volume به 250 گیگابایت افزایش یابد. این کار بدون downtime انجام می گیرد. ویژگی دیگر LVM که در سیستم فایلی مانند EXT3 وجود ندارد این است LVM قادر به ایجاد Snapshot Backup از Logical Volume بدون Unmount کردن سیستم فایل است.

هر Volume Group مجموعه ای از Physical or Logical Volume ها است که در عموم سیستم ها تنها به یک Volume Group نیاز است که شامل تمامی Physical or Logical Volume های موجود است. Physical Volumes برابر با دیسک هاست که Block Device هایی هستند مانند دیسک `sda` که فضای ذخیره سازی Logical Volume ها را فراهم می کنند. Logical Volume ها برابر با پارتیشن ها هستند که سیستم فایل بر روی آنها سوار می شود. (کل یک دیسک می تواند به صورت یک پارتیشن باشد یا اینکه به چندین پارتیشن تقسیم شود)



از شکل بالا مشخص است که دایرکتوری boot نمی تواند LVM باشد زیرا Bootloader نمی تواند آنرا بخواند پس اگر دایرکتوری / یا دایرکتوری ریشه، به صورت LVM بود، می بایست boot را در پارتیشنی جدا از دایرکتوری / و غیر LVM قرار داد.

همانطور که گفته شد Volume Group می تواند به چندین Logical Volume تقسیم شود که به نقاط اتصالی مانند دایرکتوری / و یا دایرکتوری /home/ و غیره اختصاص داده شده باشند. وقتی که فضای یک پارتیشن پر می شود، فضای اضافی را می توان از یک Volume Group به پارتیشن اختصاص داد تا فضای پارتیشن افزایش یابد. زمانی که یک دیسک جدید (یک هارد جدید) به سیستم اضافه می شود، می تواند به Volume Group اضافه شود و پارتیشن هایی که Logical Volume هستند می توانند افزایش یابند.



## مزایای LVM

برای سیستم های کوچک : زمانی که شما در سیستم خانگی خودتون با مشکل کم بودن فضا مواجه می شوید و برای مثال شاخه home شما پر می شود می توانید به راحتی یک هارد دیسک جدید تهیه کنید و فضای جدید را به راحتی به پارتیشن home خود اضافه کنید. بدون آنکه نیاز به نصب مجدد سیستم عامل داشته باشید.

برای سیستم های بزرگ: برای سیستم های بزرگ مدیریت دیسکها می تواند کار بسیار زمان بری باشد. با کمک LVM مدیر سیستم می تواند تنها زمانی که به فضای بیشتری نیاز داشت یک دیسک جدید به سیستم اضافه کرده و آن را به فضای قبلی اضافه کند.

## ساختمان LVM

hda1 hdc1 (PV:s on partitions or whole disks)

diskvg (VG)

usr1v root1v var1v (LV:s)

ext2 reiserfs xfs (filesystems)

برای کار با LVM باید با بخشهای مختلف ساختمان LVM آشنا شوید که در اینجا آنها را معرفی خواهیم کرد.

### :(PV) Physical Volume

PV ها معمولا یک هاردیسک یا چیزی شبیه با آن (مثلا یک Raid Device) می باشد.

### :(VG) Volume Group

VG بالاترین سطح ظاهری است که به وسیله LVM استفاده می شود. VG مجموعه ای از LV ها و PV ها را در یک واحد مدیریتی جمع می کند.

### : (LV) Logical Volume

مساوی پارتیشن ها در سیستم های غیر LVM است.

### : (PE) Physical Extent

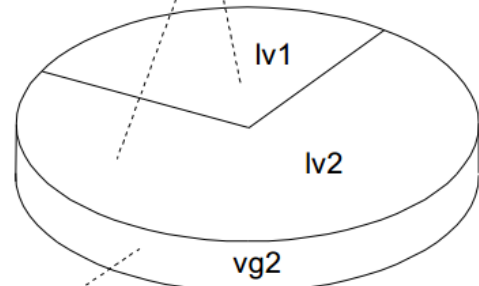
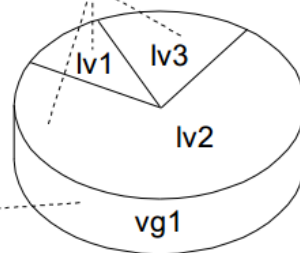
هر PV به تکه های بزرگی از داده تقسیم می شود به نام PE ، این تکه ها (extent) دارای اندازه ای برابر LE در LV ها هستند.

## : (LE) Logical extent

هر LV به تکه های بزرگی از داده تقسیم می شود به نام LE ، این اندازه برای تمام LV ها در VG یکسان است.

### Logical Volumes (LV)

Equivalent of "Partitions"

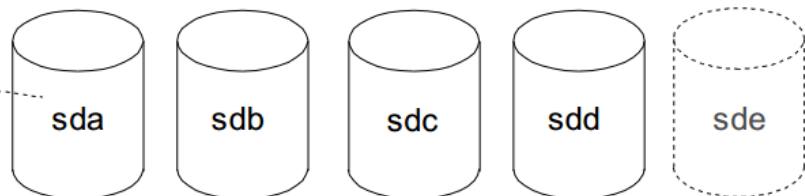


### Volume Groups (VG)

Equivalent of "Hard Discs"

### Physical Volumes (PV)

Has no equivalent: Completely hidden to the system behind the LVM



## شروع کار با LVM

اولین کاری که باید انجام دهیم Initialize کردن پارتیشن هاست. قبل از این کارها باید توسط دستوراتی مانند fdisk هارد یا پارتیشن مورد نظر را پیکربندی کرده و آنها را جزء پارتیشن های LVM ای قرار داده باشیم. ساخت PV توسط دستور pvcreate انجام می گیرد. این دستور یک توصیفگر VG در اول دیسک ایجاد می کند.

### ساخت PV

```
# pvcreate /dev/sda5
```

### ساخت VG

خوب حالا می توانیم یک VG بسازیم.

```
# vgcreate my_volume_group /dev/sda5
```

my\_volume\_group : یک نام دلخواه است که باید به VG داده شود.

## اضافه کردن یک PV به VG

در صورتی که بخواهید یک PV دیگر را به VG اضافه کنید می توانید به شکل زیر عمل نمایید:

```
# vgextend my_volume_group /dev/sdb6
```

## ساخت LV

برای ساختن یک LV به ظرفیت ۱۰ گیگ به صورت زیر عمل کنید:

```
# lvcreate -L 10G my_volume_group -name my_logical_volume
```

اگر بخواهید یک LV بسازید که تمام VG را در بر داشته باشد از vgdisplay استفاده کنید تا مجموع PE های موجود را ببینید سپس دستور lvcreate را اجرا کنید:

```
# vgdisplay | grep "Total PE"
```

```
Total PE 3576
```

در اینجا ۳۵۷۶ عدد PE در این VG وجود دارد. برای ساخت LV که تمام این فضا را شامل شود از lvcreate به شکل زیر می توانیم استفاده کنیم:

```
# lvcreate -l 3576 my_volume_group -name my_logical_volume
```

دقت کنید در اینجا از حرف کوچک l برای مقدار دهی استفاده کردیم.

## ساخت سیستم فایل

اکنون LV آماده است و شما می توانید با آن به صورت یک پارتیشن معمولی رفتار کرده و آن را فرمت کنید:

```
# mkfs.ext3 /dev/my_volume_group/my_logical_volume
```

سپس آن را mount و از آن استفاده نمایید. در صورتی که می خواهید در هنگام راه اندازی سیستم به صورت خودکار mount شود آن را به fstab اضافه کنید.

## توسعه یک LV

در صورتی که یک PV به VG اضافه کردید یا در VG فعلی فضای خالی در اختیار دارید می توانید LV را توسعه دهید. برای توسعه LV به دو صورت می توانید عمل کنید:

```
# lvextend -L12G /dev/my_volume_group/my_logical_volume
```

دستور بالا حجم LV را به ۱۲G افزایش می دهد.

```
# lvextend -L+1G /dev/my_volume_group/my_logical_volume
```

دستور بالا یک گیگابایت به my\_logical\_volume اضافه می کند.

بعد از آنکه LV را توسعه دادید و شما باید سیستم فایل را به اندازه ایی که با آن مطابقت داشته باشد افزایش دهید. با کمک دستور `resize2fs` می توانید این کار را انجام دهید. نیاز نیست نگران مشخص کردن اندازه باشید. قبل از اجرای `resize2fs` سیستم فایل را چک کنید:

```
#e2fsck -f /dev/my_volume_group/my_logical_volume
#resize2fs /dev/my_volume_group/my_logical_volume
```

## پاک کردن LVM

برای پاک کردن LVM بر عکس مسیر ساخت عمل می کنیم. ابتدا باید LV را پاک شود و قبل از همه باید آن را `umount` کنید. سپس با کمک دستور زیر آن را `remove` کنید:

```
#lvremove /dev/my_volume_group/my_logical_volume
```

بعد از اون نوبت به پاک کردن VG می رسد:

```
#vgremove my_volume_group
```

## و در آخر پاک کردن PV:

```
#pvremove /dev/sdb1
```

جهت درک بهتر مطالب گفته شده ، در ادامه به دو روش ، ساخت LVM آموزش داده می شود. ابتدا به صورت کامندی و بر اساس چند سناریو کوچک و در ادامه به صورت گرافیکی در زمان نصب توزیع CentOS بیان می شود.

## راه اندازی LVM به صورت کامندی

در این سناریو ما سیستمی داریم که علاوه بر اینکه یک هارد دومی دارد مقداری از هارد اول آن استفاده شده و مقداری از آن باقی مانده است. ما می خواهیم ابتدا از باقی مانده آن یک پارتیشن لینوکسی بسازیم لذا با دستور `fdisk` ابتدا یک آمار از موجودی ظرفیت هارد دیسک می گیریم:

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           38       305203+   83  Linux
/dev/sda2             39          430       3148740   83  Linux
/dev/sda3            431          688       2072385   83  Linux
/dev/sda4            689         1305       4956052+    5  Extended
/dev/sda5            689          819       1052226   82  Linux swap / Solaris

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
[root@localhost ~]# _
```

سپس با دستور fdisk وارد فضای پارتیشن بندی شده و یک پرینت از پارتیشن های هارد می گیریم.

```
[root@localhost ~]# fdisk /dev/sda

The number of cylinders for this disk is set to 1305.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           38       305203+   83  Linux
/dev/sda2             39          430       3148740   83  Linux
/dev/sda3            431          688       2072385   83  Linux
/dev/sda4            689         1305       4956052+    5  Extended
/dev/sda5            689          819       1052226   82  Linux swap / Solaris

Command (m for help): _
```

در اینجا به ما اعلام میکند از سیلندر 820 تا 1305 هارد خالی است لذا ما اولین سیلندر را 821 قرار داده و در خط بعد ظرفیتی که می خواهیم پارتیشن مذکور داشته باشد را وارد می کنیم.

```
Command (m for help): n
First cylinder (820-1305, default 820): 821
Last cylinder or +size or +sizeM or +sizeK (821-1305, default 1305): +2G
Command (m for help): _
```

همانطور که در شکل زیر می بینید پارتیشن جدید با نام sda6 ساخته شد.

```
Command (m for help): p

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           38       305203+   83  Linux
/dev/sda2              39          430      3148740   83  Linux
/dev/sda3             431          688      2072385   83  Linux
/dev/sda4             689         1305      4956052+    5  Extended
/dev/sda5             689          819      1052226   82  Linux swap / Solaris
/dev/sda6            821         1064      1959930   83  Linux

Command (m for help): _
```

پارتیشن جدید لینوکسی می باشد که باید آن را به یک پارتیشن LVM ای تبدیل کنیم. برای این کار از دستور زیر استفاده می کنیم

```
Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): _
```

و سپس با تایپ دستور **l** لیست پارتیشنهایی که می تواند برای ما بسازد را مشاهده کرده و نوع دلخواه را انتخاب می کنیم.



```

0 Empty 1e Hidden W95 FAT1 80 Old Minix bf Solaris
1 FAT12 24 NEC DOS 81 Minix / old Lin c1 DRDOS/sec (FAT-
2 XENIX root 39 Plan 9 82 Linux swap / So c4 DRDOS/sec (FAT-
3 XENIX usr 3c PartitionMagic 83 Linux c6 DRDOS/sec (FAT-
4 FAT16 <32M 40 Venix 80286 84 OS/2 hidden C: c7 Syrix
5 Extended 41 PPC PReP Boot 85 Linux extended da Non-FS data
6 FAT16 42 SFS 86 NTFS volume set db CP/M / CTOS / .
7 HPFS/NTFS 4d QNX4.x 87 NTFS volume set de Dell Utility
8 AIX 4e QNX4.x 2nd part 88 Linux plaintext df BootIt
9 AIX bootable 4f QNX4.x 3rd part 8e Linux LVM e1 DOS access
a OS/2 Boot Manag 50 OnTrack DM 93 Amoebs e3 DOS R/O
b W95 FAT32 51 OnTrack DM6 Aux 94 Amoebs BBT e4 SpeedStor
c W95 FAT32 (LBA) 52 CP/M 9f BSD/OS eb BeOS fs
e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a0 IBM Thinkpad hi ee EFI GPT
f W95 Ext'd (LBA) 54 OnTrackDM6 a5 FreeBSD ef EFI (FAT-12/16/
10 OPUS 55 EZ-Drive a6 OpenBSD f0 Linux/PA-RISC b
11 Hidden FAT12 56 Golden Bow a7 NeXTSTEP f1 SpeedStor
12 Compaq diagnost 5c Priam Edisk a8 Darwin UFS f4 SpeedStor
14 Hidden FAT16 <3 61 SpeedStor a9 NetBSD f2 DOS secondary
16 Hidden FAT16 63 GNU HURD or Sys ab Darwin boot fb VMware VMFS
17 Hidden HPFS/NTF 64 Novell Netware b7 BSDI fs fc VMware VMKCORE
18 AST SmartSleep 65 Novell Netware b8 BSDI swap fd Linux raid auto
1b Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid fe LANstep
1c Hidden W95 FAT3 75 PC/IX be Solaris boot ff BBT
Hex code (type L to list codes): _

```

با وارد کردن گزینه 8e نوع پارتیشن مذکور به LVM تغییر پیدا می کند.

```

Hex code (type L to list codes): 8e
Changed system type of partition 6 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           38       305203+   83  Linux
/dev/sda2              39          430      3148740   83  Linux
/dev/sda3             431          688      2072385   83  Linux
/dev/sda4             689         1305     4956052+    5  Extended
/dev/sda5             689          819      1052226   82  Linux swap / Solaris
/dev/sda6            821         1064     1959930   8e  Linux LVM

Command (m for help): _

```

سپس جهت اعمال تغییرات از گزینه W استفاده می کنیم اما همانطور که میبینید پیام سیستم این مفهوم را می رساند که باید سیستم ریست شود تا جدول پارتیشن سیستم بروز گردد. این کار در سرور امکان پذیر نیست و ممکن است باعث صدماتی شود.

```

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@localhost ~]# _

```

لذا برای حل این گونه مشکلات شرکت ردهت دستور `partprobe` را ارائه کرد. اگر بخواهیم کل جدول پارتیشن سیستم به روز شود از دستور اول استفاده و اگر بخواهیم فقط تغییرات صورت گرفته در جدول پارتیشن سیستم درج شود از دستور دوم استفاده می کنیم.

```

[root@localhost ~]# man partprobe
[root@localhost ~]# partprobe /dev/sda
[root@localhost ~]# partprobe /dev/sda6
[root@localhost ~]#

```

همانطور که می بینید یک `sdb` هم وجود دارد که جهت اجرای سناریو آن را هم پارتیشن بندی می کنیم.

```

[root@localhost ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           38       305203+   83  Linux
/dev/sda2              39          430      3148740   83  Linux
/dev/sda3             431          688      2072385   83  Linux
/dev/sda4             689         1305      4956052+    5  Extended
/dev/sda5             689          819      1052226   82  Linux swap / Solaris
/dev/sda6             821         1064      1959930   8e  Linux LVM

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System

```

```
[root@localhost ~]# fdisk /dev/sdb

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
    (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1958, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1958, default 1958): +2G

Command (m for help): p

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks    Id  System
/dev/sdb1             1         244     195989+    83  Linux

Command (m for help): █
```

همانطور که میبینید پارتیشن ساخته شده لینوکسی است و باید به LVM تغییر type پیدا کند.

```
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): █
```

در اینجا کد تغییر type را وارد کرده تغییرات را write می کنیم.

```

Hex code (type L to list codes): 8e
Changed system type of partition 1 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1             1           244     195989+   8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@localhost ~]# █

```

در انتها جدول پارتیشن را بروز کرده و دوباره از پارتیشن‌های موجود جهت اطمینان از تغییر لیست می‌گیریم.

```

[root@localhost ~]# partprobe /dev/sdb1
[root@localhost ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *             1           38     305203+   83  Linux
/dev/sda2             39          430     3148740   83  Linux
/dev/sda3            431          688     2072385   83  Linux
/dev/sda4            689         1305     4956052+    5  Extended
/dev/sda5            689          819     1052226   82  Linux swap / Solaris
/dev/sda6            821         1064     1959930   8e  Linux LVM

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1             1           244     195989+   8e  Linux LVM

```

تا الان ما فقط دو پارتیشن با نوع lvm ساخته ایم. برای ادامه کار باید با آنها یک **PV** تشکیل دهیم. لازم به ذکر است این پارتیشن‌ها بر روی هارد های جداگانه قرار دارد.

```

[root@localhost ~]# pvdisplay
/dev/hdc: open failed: No medium found
[root@localhost ~]# pvcreate /dev/sda6 /dev/sdb1
Physical volume "/dev/sda6" successfully created
Physical volume "/dev/sdb1" successfully created
[root@localhost ~]# █

```

بعد از ساخت PV، جهت اطمینان از صحت ساخت با دستور `pvdisk` یک آمار از سیستم می گیریم:

```
[root@localhost ~]# pvdisk
/dev/hdc: open failed: No medium found
"/dev/sda6" is a new physical volume of "1.87 GB"
--- NEW Physical volume ---
PV Name                /dev/sda6
VG Name
PV Size                1.87 GB
Allocatable            NO
PE Size (KByte)        0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                kBfKgt-seno-jfVU-5hF6-yXXb-a3IN-2aTT3c

"/dev/sdb1" is a new physical volume of "1.87 GB"
--- NEW Physical volume ---
PV Name                /dev/sdb1
VG Name
PV Size                1.87 GB
Allocatable            NO
PE Size (KByte)        0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                1ES0eL-h3LS-YWZo-xlYU-PHct-W2DU-c7ogPC
```

حال باید PV های درست شده را در یک گروه قرار دهیم که به این گروه VG گفته می شود. یک VG مجموع PV ها می باشد.

```
[root@localhost ~]# vgdisplay
/dev/hdc: open failed: No medium found
[root@localhost ~]# vgcreate testlvm /dev/sda6 /dev/sdb1
/dev/hdc: open failed: No medium found
Volume group "testlvm" successfully created
[root@localhost ~]# echo $?
0
[root@localhost ~]# █
```

سپس از محتویات VG یک لیست میگیریم:

```
[root@localhost ~]# vgdisplay
/dev/hdc: open failed: No medium found
--- Volume group ---
VG Name                testlvm
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   1
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0
Open LV                 0
Max PV                 0
Cur PV                 2
Act PV                 2
VG Size                 3.73 GB
PE Size                 4.00 MB
Total PE                956
Alloc PE / Size         0 / 0
Free PE / Size           956 / 3.73 GB
VG UUID                 ysD7D2-lgaK-T51R-bPH6-geC3-D6ax-nCzLoY
```

بعد از این مرحله باید از VG یک LV بسازیم که یا به یک دارکتوری ما اضافه شود و یا خودش یک دایرکتوری مجزا باشد.  
 -L : با این آپشن حجم LV را مشخص می کنیم.  
 testlvm : نام VG را وارد می کنیم. (این نام فرضی بوده و هر چیزی می تواند باشد)  
 -n datalvm : با این آپشن نام LV را مشخص می کنیم.

```
[root@localhost ~]# lvcreate -L 3G testlvm -n datalvm
/dev/hdc: open failed: No medium found
Logical volume "datalvm" created
```



```
[root@localhost ~]# lvs
/dev/hdc: open failed: No medium found
--- Logical volume ---
LV Name                /dev/testlvm/datalvm
VG Name                testlvm
LV UUID                4FpK8m-43TG-x48k-wi2d-d2M5-yWlm-hXRRLZ
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                3.00 GB
Current LE             768
Segments               2
Allocation             inherit
Read ahead sectors     auto
  - currently set to   256
Block device           253:0
```

و با این دستور هم لیست LV ها را مشاهده می کنیم.

```
[root@localhost ~]# lvs
/dev/hdc: open failed: No medium found
--- Logical volume ---
LV Name                /dev/testlvm/datalvm
VG Name                testlvm
LV UUID                4FpK8m-43TG-x48k-wi2d-d2M5-yWlm-hXRRLZ
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                3.00 GB
Current LE             768
Segments               2
Allocation             inherit
Read ahead sectors     auto
  - currently set to   256
Block device           253:0
```

LV ای که ساخته ایم هنوز خام است و باید برای آن فایل سیستم انتخاب کرده و آن را فرمت کنیم. برای این کار از دستور زیر بهره می بریم.

```
[root@localhost ~]# mkfs.ext3 /dev/testlvm/data1vm
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
393216 inodes, 786432 blocks
39321 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=805306368
24 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

در انتها هم باید این فضای ساخته شده را mount کنیم.

```
[root@localhost ~]# mount /dev/testlvm/data1vm /DATA/
[root@localhost ~]# df -ah
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	2.0G	874M	986M	47%	/
proc	0	0	0	-	/proc
sysfs	0	0	0	-	/sys
devpts	0	0	0	-	/dev/pts
/dev/sda2	3.0G	69M	2.7G	3%	/home
/dev/sda1	289M	16M	258M	6%	/boot
tmpfs	252M	0	252M	0%	/dev/shm
none	0	0	0	-	/proc/sys/fs/binfmt_misc
sunrpc	0	0	0	-	/var/lib/nfs/rpc_pipefs
/dev/mapper/testlvm-data1vm	3.0G	69M	2.8G	3%	/DATA

حال فرض کنید که هارد سیستم پر شده و می خواهیم یک PV را به VG موجود اضافه کنیم تا مشکل کمبود ظرفیت بر طرف شود. لذا از هاردی که قبلا به سیستم اضافه شده یک مقداری را طبق روش زیر جدا می کنیم:



```
[root@localhost DATA]# fdisk /dev/sdb

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (245-1958, default 245):
Using default value 245
Last cylinder or +size or +sizeM or +sizeK (245-1958, default 1958): +5
G
Command (m for help): t
Partition number (1-4): 8e
Value out of range.
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sdb: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks    Id  System
/dev/sdb1             1          244     1959898+    8e  Linux LVM
/dev/sdb2          245          853     4891792+    8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

سپس برای اعمال تغییرات سیستم عامل را مجبور می‌کنیم تا جدول پارتیشن‌های سیستم را بخواند:

```
[root@localhost DATA]# partprobe /dev/sdb
[root@localhost DATA]# partprobe /dev/sdb2
```

حالا باید با دستور pvcreate از پارتیشن ساخته شده یک PV بسازیم.

```
[root@localhost DATA]# pvcreate /dev/sdb2
Physical volume "/dev/sdb2" successfully created
```

و در این قسمت باید PV ساخته شده را به VG موجود اضافه کنیم .

```
[root@localhost DATA]# vgextend testlvm /dev/sdb2
/dev/hdc: open failed: No medium found
Volume group "testlvm" successfully extended
[root@localhost DATA]# 
[root@localhost DATA]# vgsdisplay
/dev/hdc: open failed: No medium found
--- Volume group ---
VG Name                testlvm
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No  3
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV                1
Max PV                 0
Cur PV                3
Act PV                 3
VG Size                8.40 GB
PE Size                4.00 MB
Total PE               2150
Alloc PE / Size        768 / 3.00 GB
Free PE / Size         1382 / 5.40 GB
```

و برای جدا کردن یک PV از VG طبق روش زیر عمل می کنیم.

```
[root@localhost DATA]# vgreduce testlvm /dev/sdb2
/dev/hdc: open failed: No medium found
Removed "/dev/sdb2" from volume group "testlvm"
[root@localhost DATA]# █

[root@localhost DATA]# vgdisplay
/dev/hdc: open failed: No medium found
--- Volume group ---
VG Name                testlvm
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   4
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 1
Open LV                 1
Max PV                 0
Cur PV                 2
Act PV                 2
VG Size                 3.73 GB
PE Size                4.00 MB
Total PE                956
Alloc PE / Size        768 / 3.00 GB
Free PE / Size          188 / 752.00 MB
VG UUID                ySD7D2-lgaK-T51R-l
```

حال فرض کنید حجم یک پارتیشن کم است و ما می خواهیم یک PV به VG اضافه کرده تا بتوانیم حجم LV را افزایش دهیم.

```
[root@localhost DATA]# vgextend testlvm /dev/sdb2
/dev/hdc: open failed: No medium found
Volume group "testlvm" successfully extended
[root@localhost DATA]# vgdisplay
/dev/hdc: open failed: No medium found
--- Volume group ---
VG Name                testlvm
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   5
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV                1
Max PV                 0
Cur PV                3
Act PV                 3
VG Size                8.40 GB
PE Size                4.00 MB
Total PE               2150
Alloc PE / Size        768 / 3.00 GB
Free PE / Size         1382 / 5.40 GB
```

ما می خواهیم 5 گیگ به حجم LV اضافه شود

```
[root@localhost DATA]# lvdisplay
/dev/hdc: open failed: No medium found
--- Logical volume ---
LV Name                /dev/testlvm/datalvm
VG Name                testlvm
LV UUID                4FpK8m-43TG-x48k-wi2d-d2M5-yWlm-hXRRlZ
LV Write Access        read/write
LV Status              available
# open                 1
LV Size                3.00 GB
Current LE             768
Segments               2
Allocation             inherit
Read ahead sectors     auto
  - currently set to   256
Block device           253:0
```

دستور **lvresize** هم می تواند حجم را کم کند و هم می تواند آن را افزایش دهد. در اینجا عدد 8 بدین معنا است که ظرفیت LV به 8 گیگ برسد. اگر +5G هم قرار می دادیم همین کار را انجام می داد. آدرسی که در اینجا وارد می کنیم آدرس LV است که قبلا ایجاد کرده ایم. در اینجا بدون مشکل خاصی عمل **resize** با موفقیت انجام شد ولی ممکن است در بعضی مواقع پیغام **Busy** با ما نشان داده شود. علت آن هم این است که دایرکتوری مورد نظر در حال ارائه سرویس به کاربران و یا سیستم می باشد که در چنین مواقعی باید ابتدا دایرکتوری مورد نظر را **umount** کرده و بعد آن را **resize** کنیم.

```
[root@localhost DATA]# lvresize -L 8G /dev/mapper/testlvm-datalvm
/dev/hdc: open failed: No medium found
Extending logical volume datalvm to 8.00 GB
Logical volume datalvm successfully resized
[root@localhost DATA]# █
```

ما عمل **resize** را انجام دادیم ولی همانطور که می بینید افزایش حجمی به ما نشان نمی دهد.

```
[root@localhost DATA]# df -ah
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        2.0G  874M  986M  47% /
proc            0      0      0  -  /proc
sysfs            0      0      0  -  /sys
devpts          0      0      0  -  /dev/pts
/dev/sda2        3.0G   69M  2.7G   3% /home
/dev/sda1        289M   16M  258M   6% /boot
tmpfs           252M     0  252M   0% /dev/shm
none            0      0      0  -  /proc/sys/fs/binfmt_misc
sunrpc          0      0      0  -  /var/lib/nfs/rpc_pipefs
/dev/mapper/testlvm-datalvm
                 3.0G   69M  2.8G   3% /DATA
[root@localhost DATA]#
```

چون ما قبلاً روی LV مورد نظر یک فایل سیستم ایجاد کرده ایم که خود آن بلاک و آینود درست می کند. حجم با موفقیت تغییر می کند ولی مقدار صحیح را به ما نشان نمی دهد، ما باید به طریقی به آن بفهمانیم که سائز تغییر کرده است. لذا از دستور زیر برای قابل استفاده شدن فضای مورد نظر استفاده می کنیم:

-p: این گزینه درصد **resize** را به ما نشان می دهد.

-f: این گزینه هم به صورت **force** کار درخواستی را انجام می دهد.

```
[root@localhost DATA]# resize2fs -p -t /dev/mapper/testlvm-datalvm
resize2fs 1.39 (29-May-2006)
Filesystem at /dev/mapper/testlvm-datalvm is mounted on /DATA; on-line
resizing required
Performing an on-line resize of /dev/mapper/testlvm-datalvm to 2097152
(4k) blocks.
The filesystem on /dev/mapper/testlvm-datalvm is now 2097152 blocks long.
```

با وارد کردن دستور **df -ah** می توانید تغییر حجم را مشاهده کنید.

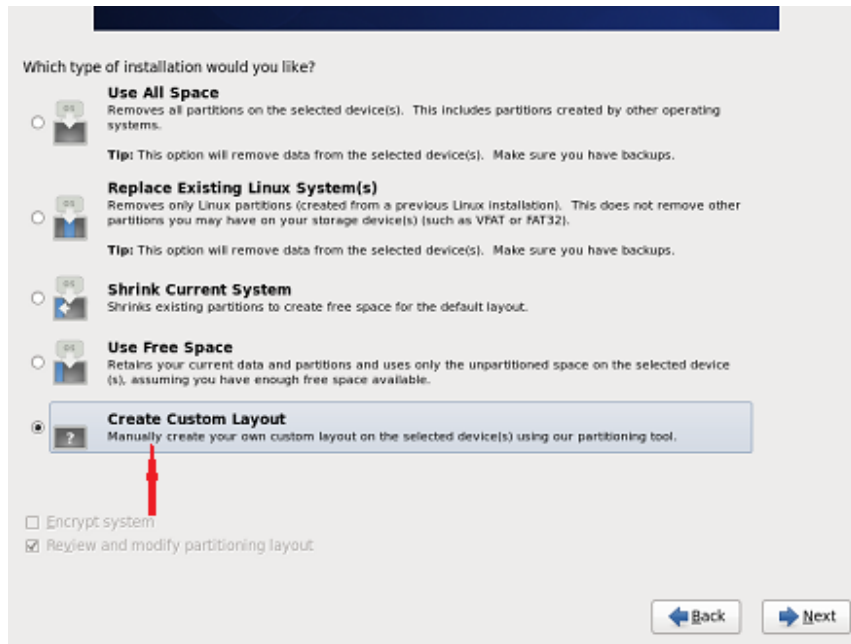
```
[root@localhost DATA]# df -ah
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        2.0G  874M  986M  47% /
proc             0      0      0  -  /proc
sysfs            0      0      0  -  /sys
devpts           0      0      0  -  /dev/pts
/dev/sda2        3.0G   69M  2.7G   3% /home
/dev/sda1        289M   16M  258M   6% /boot
tmpfs            252M      0  252M   0% /dev/shm
none             0      0      0  -  /proc/sys/fs/binfmt_misc
sunrpc           0      0      0  -  /var/lib/nfs/rpc_pipefs
/dev/mapper/testlvm-datalvm
                 7.9G   71M   7.5G   1% /DATA
[root@localhost DATA]#
```

امید است با اجرای این سناریو توانسته باشیم مفاهیم LVM را به طور ساده بیان کنیم.



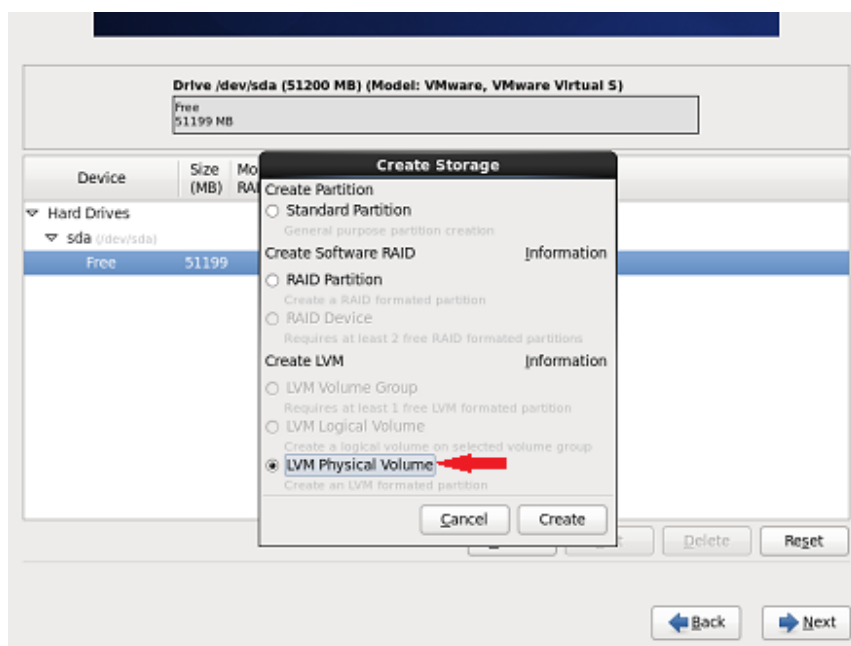
## ایجاد LVM Partition در زمان نصب لینوکس

در بخش قبلی راه اندازی و مدیریت LVM را به صورت کامندی توضیح دادم ، در این بخش راه اندازی LVM در زمان نصب سیستم عامل را بررسی می کنیم .  
پس از بوت شدن CentOS 6 در پنجره شکل زیر گزینه Create Custom Layout را انتخاب کنید.

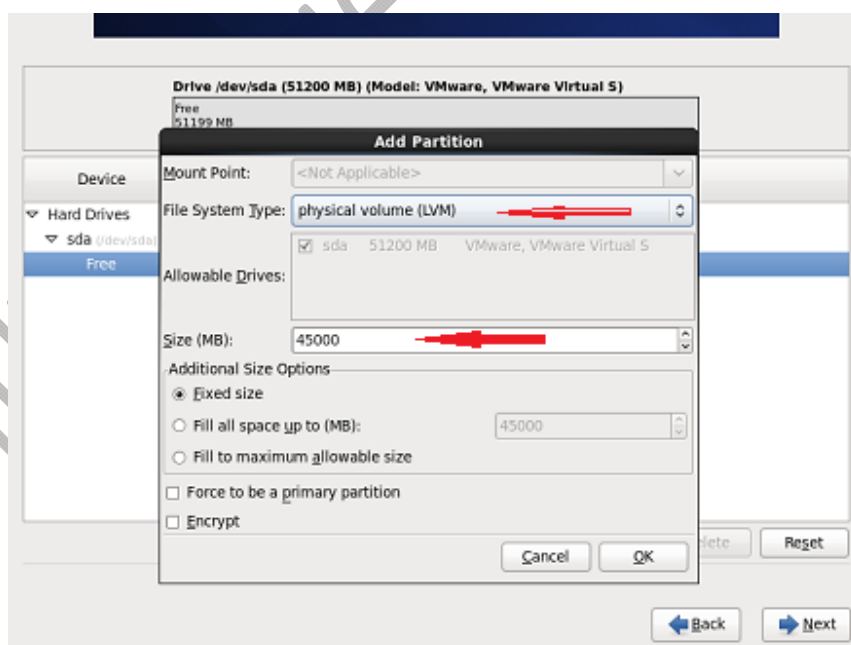


سپس از پنجره شکل زیر Create را انتخاب و از پنجره باز شده LVM Physical Volume را انتخاب و Create را کلیک کنید. نخستین گام ایجاد یک LVM Physical Volume است. سپس Group Volume ها را ایجاد و در نهایت Logical Volume ها را ایجاد می کنیم که نقش پارتیشن را داشته و به دایرکتوری ها متصل یا mount می شوند.

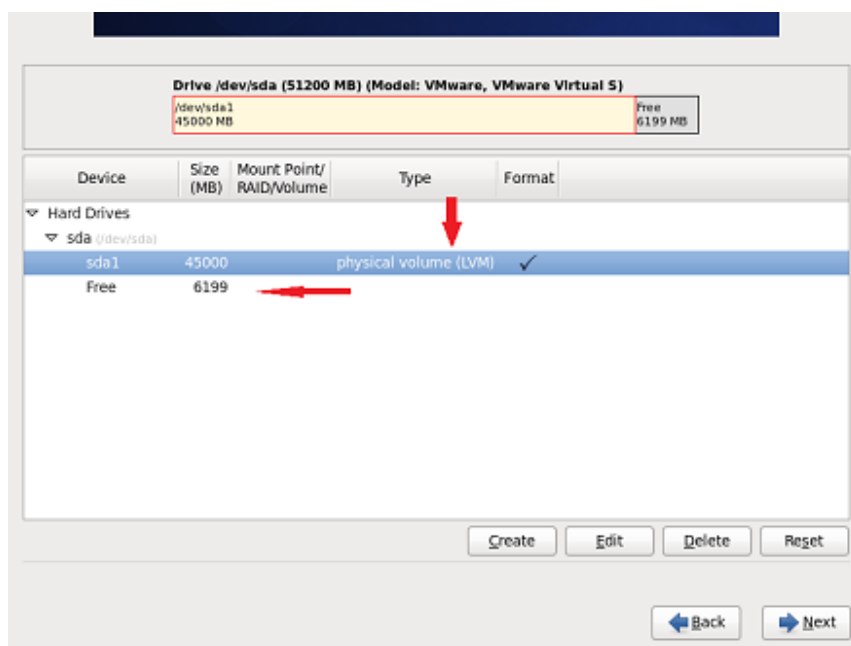




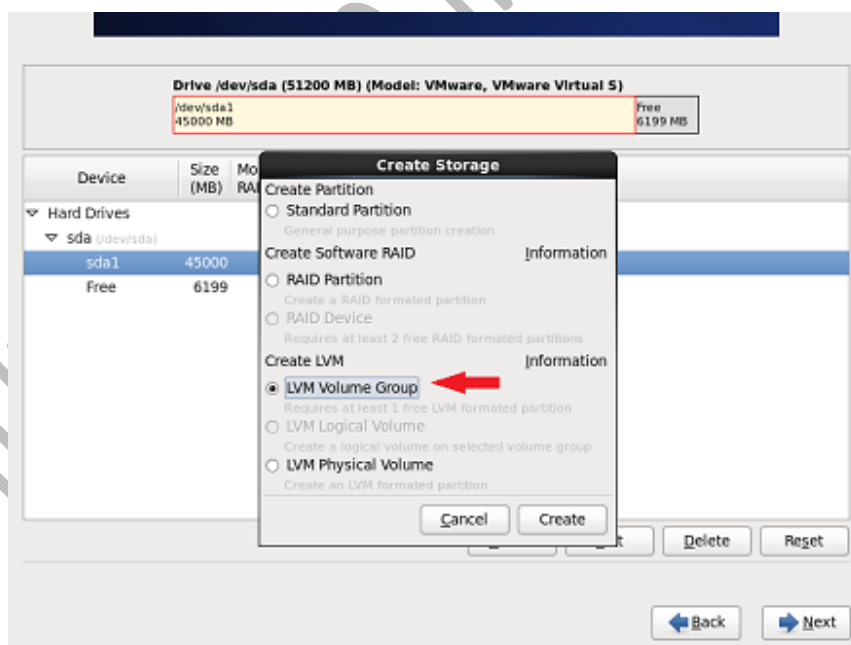
سپس مطابق با شکل زیر می توانید اندازه آنرا تعیین کنید. من از 50 گیابایت دیسکی که توسط VMware در اختیار داشتم 45000 مگابایت یا 45 گیگابایت آنرا برای LVM Physical Volume انتخاب کرده ام. توجه کنید که پارتیشن boot نباید LVM باشد پس فضایی البته فضایی کم برای آن در این سناریو لازم است.



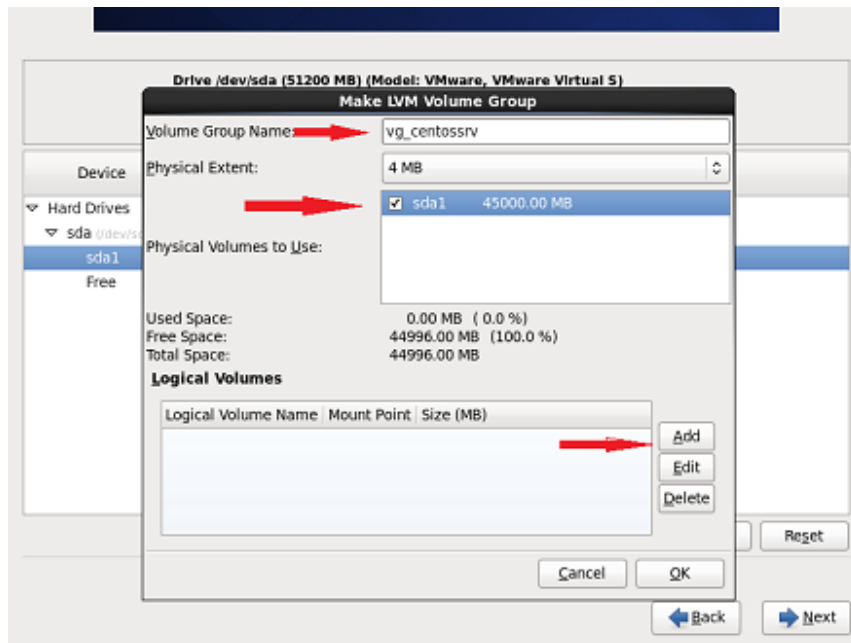
مطابق با شکل زیر خواهید دید که LVM Physical Volume به حجم 45 گیابایت ایجاد شد و 6 گیگابایت نیز بلااستفاده ماند. در گام بعدی باید یک Group Volume بر روی LVM Physical Volume ایجاد کنیم.



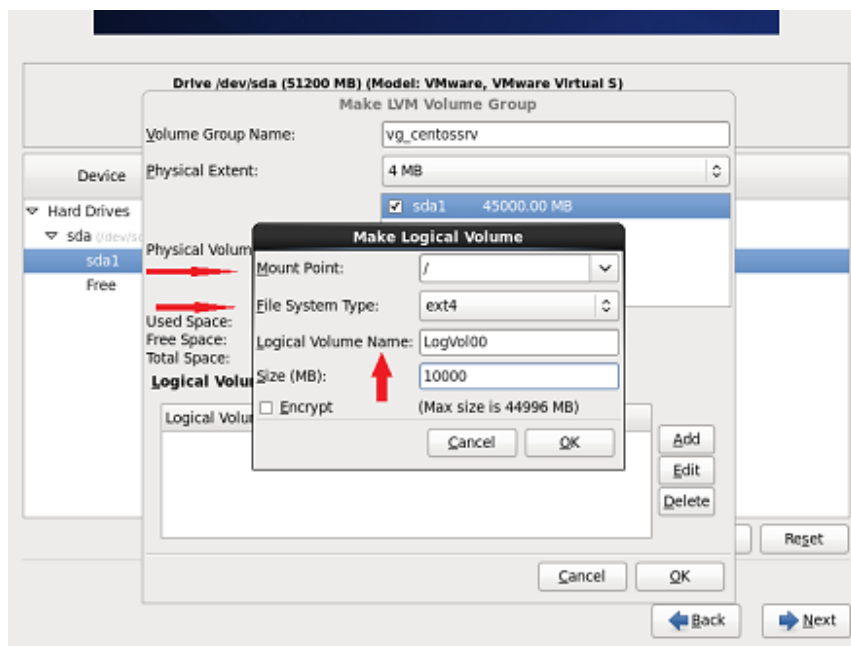
برای ایجاد یک LVM Group Volume بر روی Create پنجره شکل بالا کلیک کنید تا پنجره شکل زیر باز شود و سپس LVM Group Volume را انتخاب کنید.



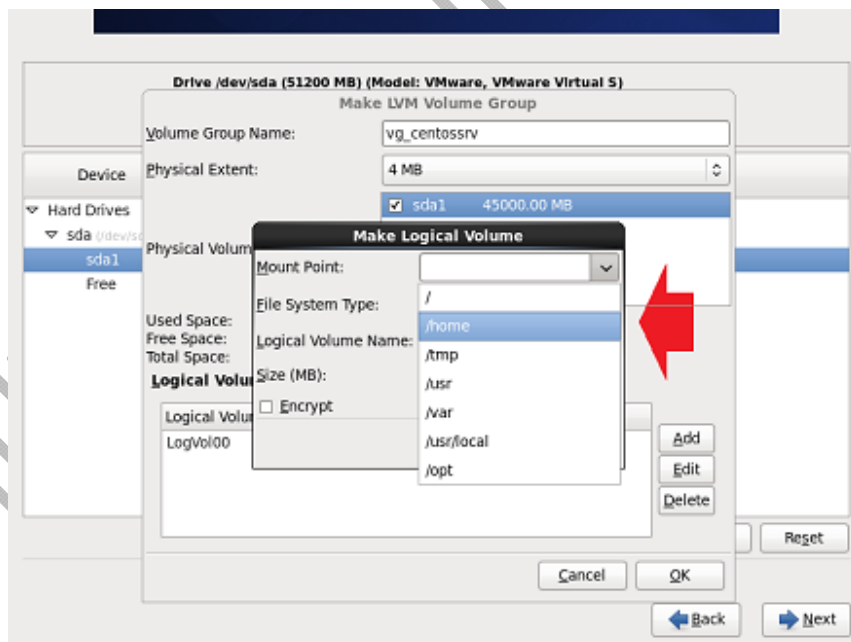
سپس در پنجره باز شده شکل زیر نام LVM Group Volume را انتخاب کنید. اگر چندین LVM Physical Volume داشته باشید می‌توانید آنها را برای عضویت در Group Volume انتخاب کنید. همانطور که گفته شد هر Group Volume می‌تواند چندین Physical Volume را داشته باشد.



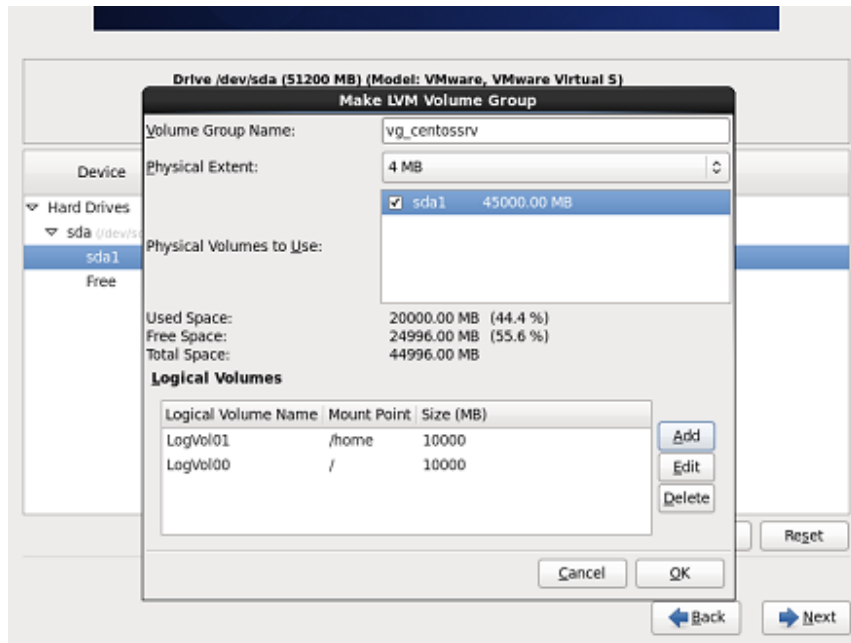
پس از تعیین نام Group Volume و انتخاب Physical Volume های آن و پیش از کلیک روی OK از شکل بالا، می توانید روی Add از شکل بالا کلیک کنید و همین جا Logical Volume ها که نقش پارتیشن را دارند ایجاد کنید و آنها را به دایرکتوری های پیشفرض موجود در نصاب CentOS متصل کنید. به هر حال نخست می بایست Physical Volume ها و سپس Group Volume ها و در نهایت Logical Volume ها را ایجاد کنید. (یعنی می توانید از شکل بالا OK کند و سپس یک Logical Volume بسازید. شکل زیر پس کلیک روی Add چگونگی ایجاد یک Logical Volume به اندازه 10000 مگابایت یا 10 گیگابایت را که روی دایرکتوری / یا دایرکتوری root با سیستم فایل ext4 نشان می دهد.



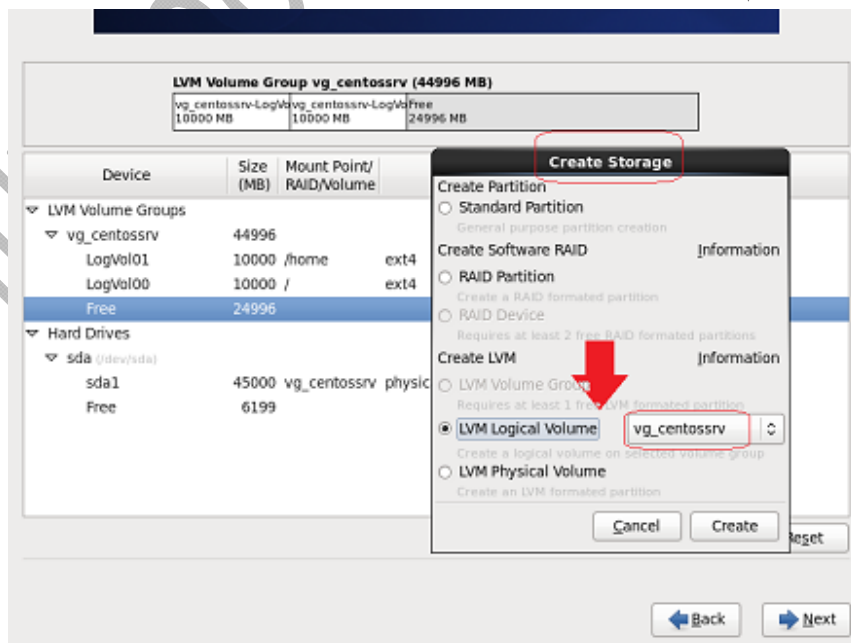
یا در شکل زیر یک از همان پنجره و پیش از OK کردن و کلیک بر روی Add یک Logical Volume به اندازه 10 گیابایت ایجاد می کنیم. همانطور که در شکل پایین مشاهده می کنید، دایرکتوری /boot در لیست نقاط اتصال نیست.



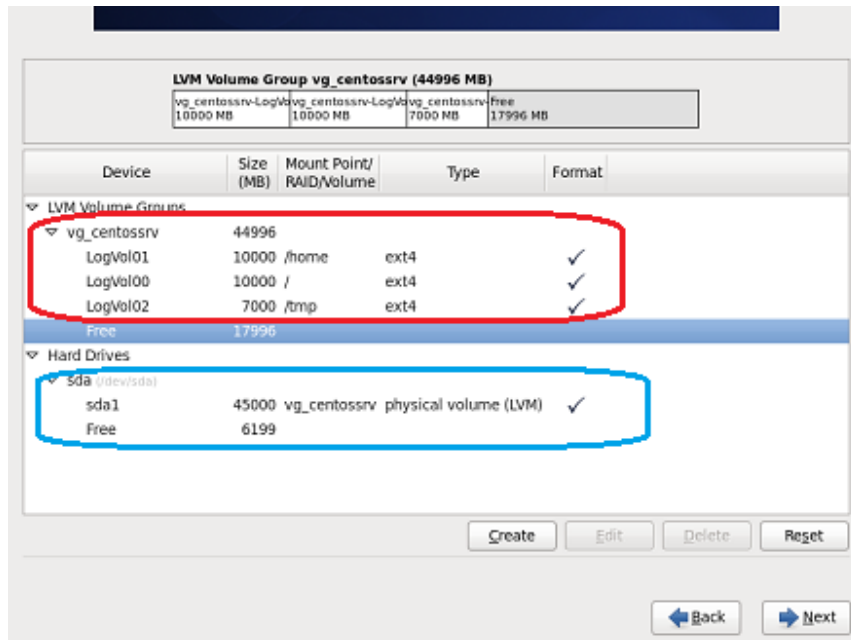
شکل زیر دو Logical Volume ایجاد شده را نشان می دهد. روی OK کلیک کنید.



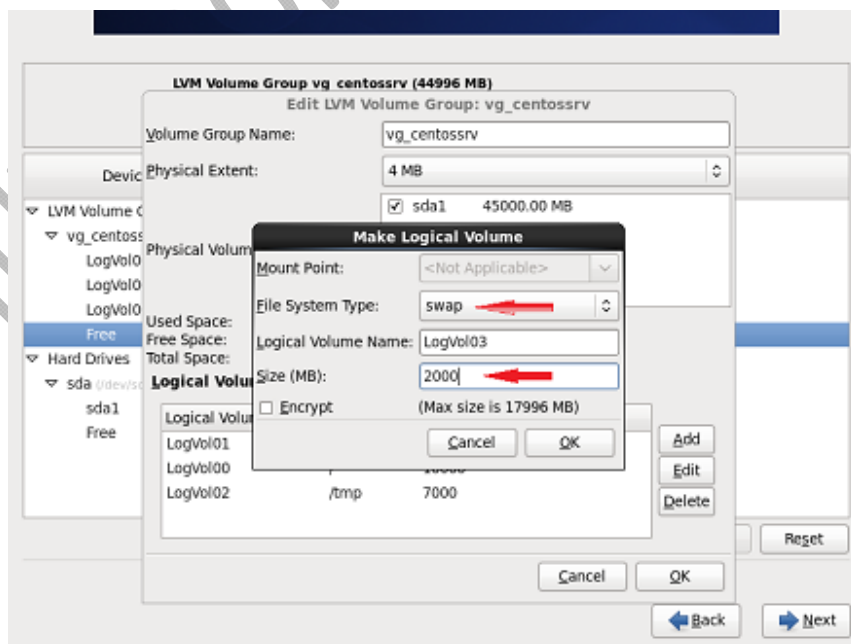
در پنجره شکل زیر نخست روی Create کلیک کنید تا پنجره Create Storage باز شود و از آن گزینه LVM Logical Volume را انتخاب کنید. در جلوی آن فهرستی از Group Volume ها خواهد آمد که باید گروه مورد نظر را انتخاب و در نهایت روی Create کلیک کنید تا یک Logical Volume دیگر برای اتصال بر روی دایرکتوری دیگری مانند tmp/ ایجاد کنیم. (پنجره ای که پس از کلیک روی Create باز خواهد شد، مانند پنجره بالا است)



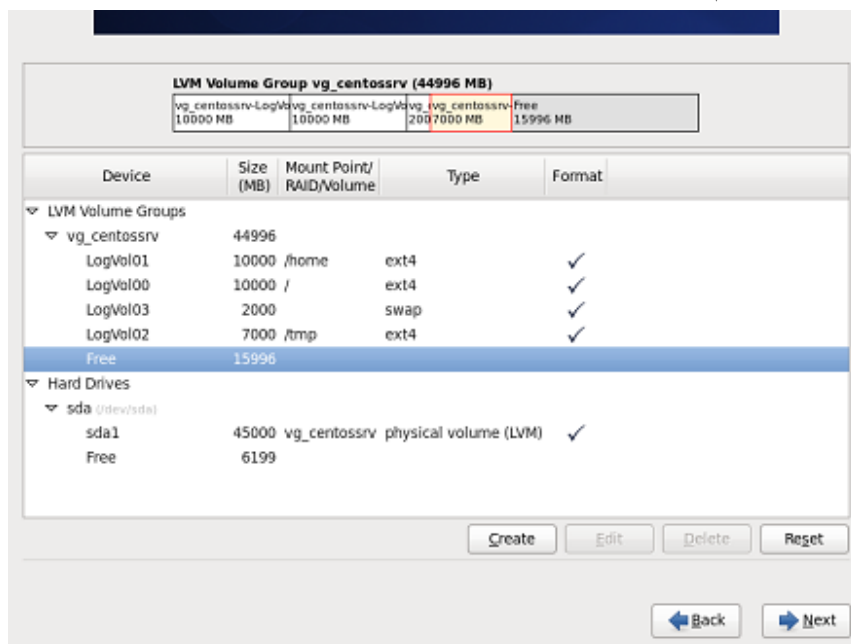
در شکل زیر خواهید دید که سه Logical Volume تحت گروه vg\_centosrv ایجاد کردیم. همانطور که می بینید هارد دیسک به دو قسمت 45 گیابایتی و 6 گیابایت خالی تقسیم شده است. از این 6 گیگ فضای خالی بعد استفاده می کنیم تا دایرکتوری /boot را بر روی آن mount کنیم.



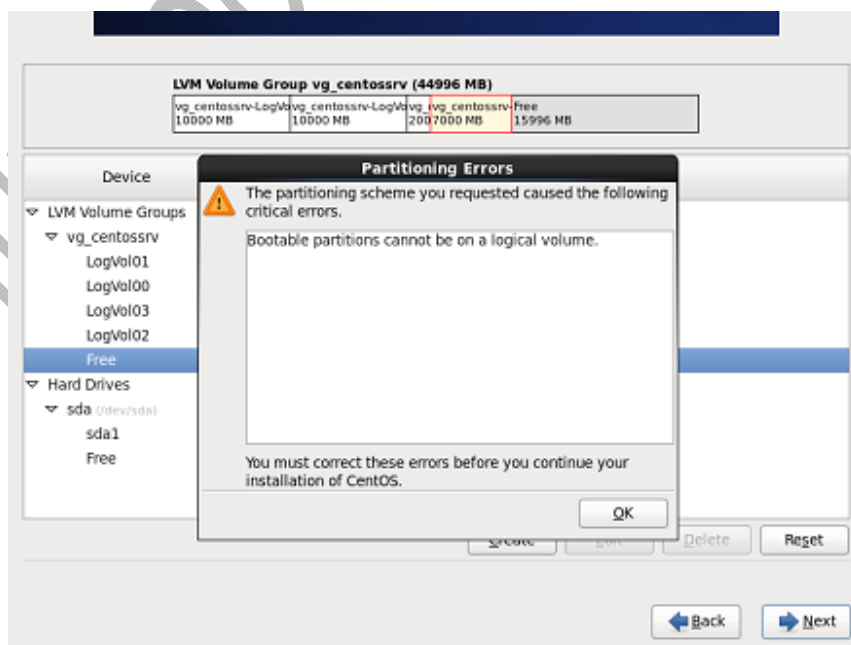
شکل زیر چگونگی ایجاد یک Logical Volume برای mount کردن پارتیشن swap را نشان می دهد و swap. دو دایرکتوری / دو دایرکتوری هستند که باید حتما ایجاد شوند. در دو مطلب 1 و 2 چگونگی ایجاد یک swap file و یک swap partition اضافی را گفته ام اما یکی دیگر از راه های ایجاد swap چه در زمان نصب و چه پس از نصب ایجاد آن تحت LVM است و این باعث می شود که بعد با انعطاف بیشتری فضای آنرا تغییر دهید.



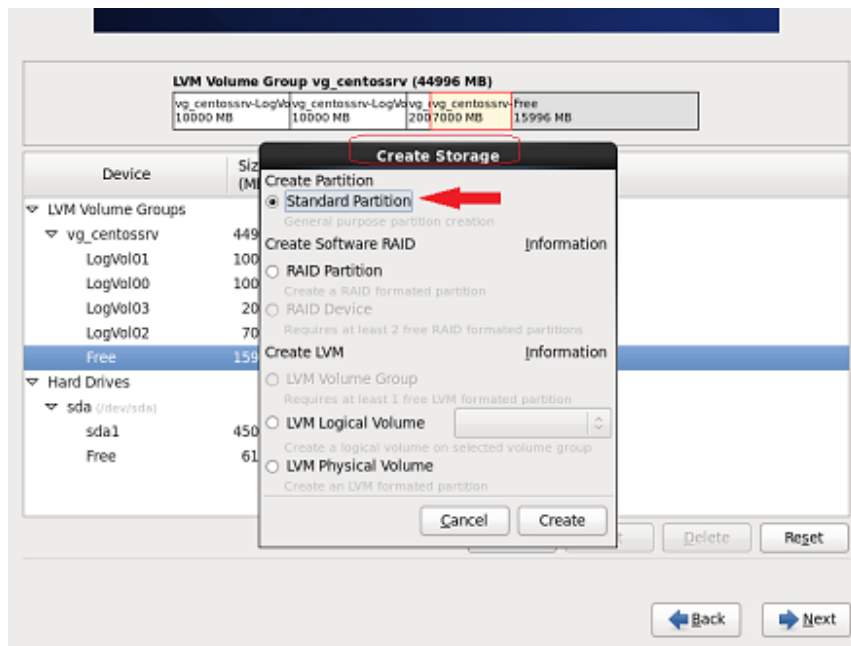
مطابق شکل زیر دو دایرکتوری لازم و اجباری به علاوه دایرکتوری های دیگر ایجاد شده اند. روی **Next** کلیک کنید تا وارد مرحله بعدی نصب **CentOS** شویم.



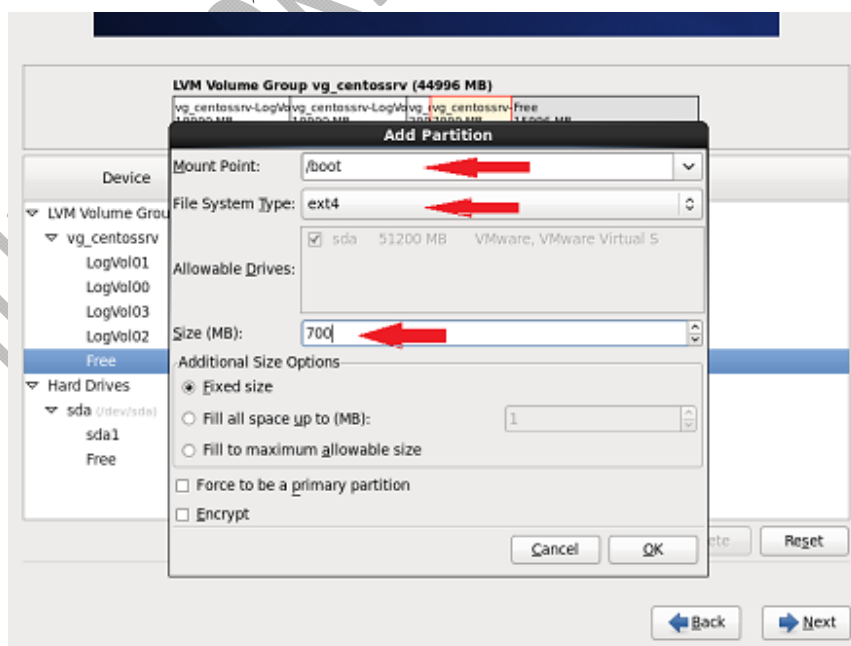
پس از کلیک روی **Next** شکل بالا باید خطای زیر نشان داده شود و به این خاطر است که دایرکتوری **/boot** تحت دایرکتوری **/** که به صورت **LVM** است و همانطور که در بالا گفته شد دایرکتوری **/boot** نمی تواند به صورت **LVM** باشد.



برای ایجاد دایرکتوری boot/ از فضاهای Free f به اندازه 6 گیگابایت استفاده کنید (آنها انتخاب کنید) و سپس روی Create کلیک تا پنجره Create Storage مانند شکل زیر باز شود و از شکل زیر Standard Partition را انتخاب کنید.

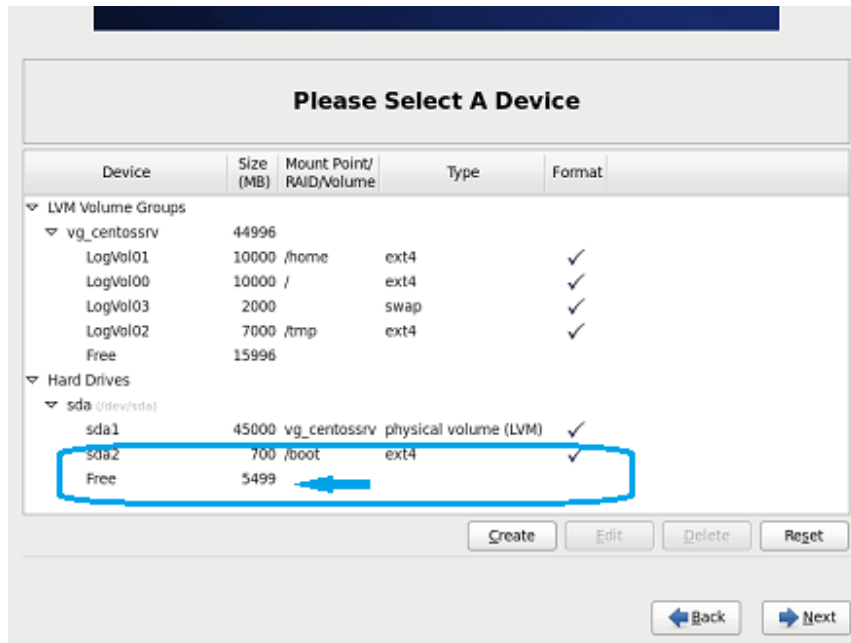


مطابق با شکل زیر من 700 مگابایت را به دایرکتوری boot/ اختصاص داده ام.





مطابق با شکل زیر تمامی موارد لازم انجام شده و می توانید برای رفتن به گام بعدی نصب CentOS روی Next کلیک کنید. همانطور که می بینید از فضای خالی 6 گیابایتی کم برای ایجاد /boot/ کم شده و نه از فضای آزاد در بخش LVM Physical Volume.



ایجاد LVM در تمامی توزیع های لینوکس شامل همین سه مرحله Create LVM Physical Volume و سپس Create Group Volume و در نهایت Create Logical Volume و اتصال آنها به دایرکتوری های مورد نظر است. (توجه کنید LVM Logical Volume نقش پارتیشن را بازی می کنند). اگر بخواهید پس از نصب توزیعی، این کار را انجام دهید دستور هایی وجود دارد که آنها در تمامی توزیع های لینوکسی یکسان هستند اما در زمان نصب تنها در ظاهر واسط گرافیکی تفاوت هایی وجود دارد.

Linux Cookie in Persian v.2

## فصل نهم

# سهامیه بندی دیسک در لینوکس

QUOTA

## سهمیه بندی دیسک توسط Disk Quota

مقدمه:

در علوم رایانه، سهمیه بندی دیسک قابلیت است در سیستم عاملهای مدرن که از آن می توان به منظور ایجاد محدودیت بر روی فایل سیستم ها و اختصاص دادن فضای دیسک سخت به روشی معقول استفاده کرد. با Disk Quota مدیران می توانند برای هر کاربر، یا هر گروهی از کاربران، استفاده از فضای دیسک را سهمیه بندی کنند. Disk Quota به صورت مجزا روی پارتیشن، پیاده سازی می شود و نمی توان آن را روی Folder راه اندازی کرد. Disk quota راهکاری است که توسط آن کاربران با توجه به مسئولیتشان، میزان فضای مورد نیاز و مجاز را دریافت می کنند.

## علت استفاده از Disk Quota

اگر کل فضای مربوط به / که همان root است پر شود سیستم به مشکل اساسی برخورد کرده و دیگر به طور صحیح کار نمی کند. این مشکل می تواند به خاطر حمله به سرور اتفاق بیفتد و یا همانطور که گفته شد به خاطر انباشت دیتای زیادی توسط یک یوزر یا یک گروه رخ دهد.

این مشکل از آنجا ناشی می شود که تمام دایرکتوری ها بر روی یک پارتیشن قرار دارند. مثلاً پارتیشن home ، یک پارتیشن مستقلی ندارد و به طور مستقیم از زیر مجموعه root محسوب میشود لذا با توجه به اینکه کاربران تمام فایل های خود را در این مکان ذخیره می کنند اگر پارتیشن home جدا از مجموعه پارتیشن های سیستمی نباشد به محظ پر شدن از دیتا، سیستم عامل در کارکرد خود به مشکل بر می خورد.

جهت رفع این مشکل باید بعضی از پارتیشن ها مثل home و var را بر روی پارتیشن های جداگانه ای قرار داد ، و با disk quota سهمیه هر کاربر را مشخص کرد.

یا در جاهایی که وب سرور و یا ایمیل سرور استفاده می کنند دایرکتوری var را بر روی یک پارتیشن جداگانه قرار می دهند تا اگر این دایرکتوری به هر نحوی پر شد در کار سیستم عامل مشکلی ایجاد نشود.

یا در سرورهایی که یوزر استفاده کننده بالایی دارند مثل فایل سرور و یا FTP ، پارتیشن home را جداگانه ایجاد می کنند، آنوقت یوزرها هر چه قدر هم دیتا ذخیره کنند محدود به همین پارتیشن است و به محدوده root کاری ندارد لذا سیستم عامل به راحتی به کار خود می پردازد چون محدوده کاری مناسب خود را در اختیار دارد. این نوع مرز بندی پارتیشن می تواند از مصرف کل دیسک توسط یک یا چند کاربر ، جلوگیری کند.

نکته: پارتیشن `/boot` حتماً باید دارای یک پارتیشن Basic باشد چون grub می‌بایست بر روی یک Basic Partition قرار بگیرد. و البته استثنائاتی هم وجود دارد مثل `/bin` , `/sbin` , `/etc` که تحت هیچ شرایطی نباید بر روی یک پارتیشن جداگانه ای قرار بگیرند اما برای دو پارتیشن `swap` و `boot` باید دو فضای اختصاصی در نظر گرفت.

## انواع سهمیه‌بندی در Disk Quota

بسته quota شامل ابزارهایی است که به شما اجازه می‌دهند تا مقدار فضای دیسک و فایل‌هایی که کاربر می‌تواند مصرف نماید را محدود نمایید. در quota دو نوع سهمیه‌بندی دیسک وجود دارد:

**نوع اول** که به سهمیه‌بندی استفاده یا سهمیه‌بندی Block معروف است، مقدار فضای خالی که می‌تواند مورد استفاده قرار گیرد را محدود می‌کند. برای مثال می‌توان محدودیتی اعمال کرد که یک کاربر فقط بتواند از ۱۰ گیگابایت فضای دیسک استفاده کند.

**نوع دوم** که به سهمیه‌بندی فایل یا سهمیه‌بندی iNod معروف است، در این متد تعداد فایل‌ها یا دایرکتوری‌هایی که می‌تواند ایجاد شوند را محدود می‌کنند. برای مثال یک کاربر حق دارد تنها ۲۰۰۰ فایل در دیسک داشته باشد. در اینجا اگر کاربری قصد استفاده بیشتر از فضای هارد را داشته باشد، Disk Quota به او این اجازه را نمیدهد. به این ترتیب میتوان کاربر را مجبور کرد تا اطلاعات بی مورد خود را از روی هارد پاک کند تا بتواند فضای خالی بیشتری داشته باشد. توجه داشته باشید در تمامی سیستم عامل‌های مبتنی بر یونیکس، همه چیز فایل است، حتی دایرکتوری‌ها و یا Device ها.

## سطوح هشدار در Disk Quota

در quota برای هر کاربر میتوان مرز هشدار تعیین نمود تا پیش از پر شدن فضای هارد به او اخطار دهد. دو سطح هشدار وجود دارد که با آن مقدار فضای قابل استفاده برای کاربر و یا گروه را مشخص می‌کنند:

- **Hard limit**: هارد حداکثر مقدار قابل استفاده به صورت مطلق قطعی را تعیین می‌کند. یعنی یوزر، حق ندارد بیش از این مقدار، فضای دیگری از هارد را در اختیار بگیرد. در صورت اقدام به ذخیره کردن دیتا، از آن جلوگیری میشود.

- **Soft limit**: مدیران سیستم معمولاً یک سطح هشدار یا سهمیه‌بندی نرم هم در نظر می‌گیرند. اگر کاربری به سطح هشدار رسید، به او خبر داده می‌شود که در حال نزدیک شدن به نهایت اندازه مجاز می‌باشد. یعنی آن حدی که به

یوزر اعلام شود فضای مجاز قابل استفاده در حال پر شدن است. **soft limit** حتما باید مشخص شود ولی تعیین **hard limit** اختیاری است. و در صورتی که **hard limit** نیز مشخص شود (که مقداری بزرگتر از **soft limit** است)، تا یک مدت زمانی کاربر می تواند تا رسیدن به مقدار **hard limit** از فضای دیسک استفاده نماید و به محض رسیدن به مقدار **hard limit**، به هیچ عنوان مجاز به استفاده از فضای دیسک نمی باشد.

- **Grace Period**: مشخص کننده فضای قابل استفاده مابین **soft** تا **hard** می باشد. در اصل زمانی است از اعلام **Soft limit** تا زمانی که تایم آن سر برسد به محض سر رسیدن این زمان یوزر مربوطه **lock** می شود.

فرض کنید یک یوزری 8 مگابایت فضای **soft** دارد و **hard** آن 10 مگابایت می باشد. **Grace Period** آن را هم 7 روز قرار می دهیم. اگر **Grace Period** یوزر به روز هفتم برسد و یوزر مربوطه هنوز از 10 مگابایت فضای خود استفاده نکرده باشد، در هر مقداری که باشد دیگر اجازه فعالیت به کاربر داده نمی شود. نکته: برای کار با **quota** یا باید **selinux** را خاموش و یا اینکه آن را پیکربندی کنید.

## نصب و راه اندازی سرویس Disk Quota

در ابتدا می بایست بررسی کنیم که آیا کرنل مورد استفاده از **Disk Quota** پشتیبانی میکند یا خیر؟ لذا برای بررسی، از دستور زیر استفاده می کنیم. اگر خروجی دستور بدین صورت باشد، بدین معناست که کرنل، از سهمیه بندی دیسک پشتیبانی میکند در غیر این صورت باید کرنل را مجدداً کامپایل کرد.

```
#dmesg | grep quota
VFS: Disk quotas dquot_6.5.2
```

سپس برای اینکه از نصب بودن پکیج **quota** اطمینان حاصل کنیم با دستور زیر از سیستم **query** می گیریم

```
#rpm -qa | grep quota
```

در صورت نصب نبودن، در سیستم های ردهت جهت نصب **quota** از **yum** استفاده می کنیم:

```
#yum -y install quota
```

بعد از نصب، باید اطمینان حاصل کنیم که آیا پکیج **quota** بر روی سیستم نصب شده است یا خیر لذا دوباره از سیستم **query** می گیریم:

```
#rpm -qa | grep quota
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم

```
#rpm -ql quota
```

و با این دستور هم اطلاعات لازم در مورد پکیج این سرویس را به دست می آوریم:

```
#rpm -qi quota
```

بسته **quota** شامل مجموعه ابزارهایی است که به شما اجازه می دهند تا مقدار فضای دیسک را محدود و مدیریت کنید که شامل ابزارهای زیر می شوند:

**quota** : مقدار فضای استفاده شده توسط یک کاربر را نمایش می دهد.

**edquota** : برای ویرایش سهمیه بندی ها استفاده می شود.

**repquota** : سهمیه بندی های یک فایل سیستم را به صورت خلاصه اعلام می کند.

**quotaon** : سهمیه بندی ها را فعال می کند.

**quotaoff** : سهمیه بندی را غیر فعال می کند.

**quotacheck** : برای بررسی ثبات و استحکام فایل سیستم ها استفاده می شود.



## اجرا و اعمال Disk Quota

رویه ای که در چند بخش بعدی شرح داده می شود بر این فرض است که شما برای ساخت سهمیه ها، یک پارتیشن مجزا برای `/home` در نظر گرفته اید، البته می توانید علاوه بر پارتیشن `home` از هر پارتیشن دیگری استفاده نمایید. اگر شما فقط یک پارتیشن نصب شده در سیستم فایل (`/`) دارید، می توانید سهمیه های مربوط به کل سیستم فایل را با جایگزینی `/home` با `/` را بوسیله ادیت فایل `fsts` انجام دهید.

### ویرایش فایل `/etc/fstab`

ما باید ابتدا پشتیبانی از سهمیه بندی دیسک را به سیستم فایل اضافه نماییم. برای انجام این کار فایل `/etc/fstab` را ویرایش نموده و گزینه های `usrquota` و `grpquota` را به فیلد چهار پارتیشنی که می خواهید سهمیه ها را برای آن تنظیم نمایید اضافه کنید. اگر بخواهیم `quota` به یوزرها اعمال شود فقط `usrquota` و اگر بخواهیم به گروه ها اعمال شود `grpquota` را در این خط وارد کرده و اگر بخواهیم به هر دو اعمال شود هر دو کلمه را می آوریم.

```
#vi /etc/fstab
```

```
/dev/sda2    /home    ext3    defaults,usrquota,grpquota    1    2
```

با این کار سهمیه بندی دیسک برای تمامی دایرکتوری های خانگی کاربران تحت دایرکتوری `/home` بکار برده می شود. پیش از اینکه تغییرات اعمال شده بتواند اثر کند، سیستم فایل باید مجددا در دسترس قرار بگیرد. لذا با `remount` و دوباره `mount` پارتیشن مورد نظر تغییرات را به فایل `fstab` اعمال می کنیم.

```
#mount -o remount /dev/sdb2
```

اگر `quota` را برای سیستم فایل ریشه تنظیم کرده باشیم مجبوریم برای اعمال تغییرات سیستم را `reboot` کنیم.

**نکته مهم:** اصولاً وقتی در هارد تغییراتی اعمال کنیم اگر کرنل باری کاری زیادی داشته باشد به این سادگی تغییرات را اعمال نمی کند. برای اینکه `partition table` به روز شود باید کرنل را مجبور کنیم این جدول را بخواند. در سیستم های ردهت می توانیم از دستور انحصاری این شرکت جهت خواندن دوباره جدول پارتیشن استفاده کنیم:

```
#partprobe /dev/sdb2
```

### ساخت فایل های `quota`

برای تنظیم سهمیه کاربران و یا گروه ها شما نیاز دارید تنظیمات را در فایل هایی به نام `aquota.user` و `aquota.group` وارد کنید. این فایلها باید در ریشه پارتیشنی که `Quota` را روی آن فعال کرده ایم، قرار بگیرند. این



فایل ها به صورت پیش فرض وجود ندارند و باید ساخته شوند. یک روش ساخت این فایل ها بکارگیری دستور quotacheck می باشد. دستور quotacheck موجود در اینجا به دنبال پارتیشن سیستم فایل نصب شده بر روی /home می گردد تا یک جدول از فضای مصرف شده از دیسک بسازد.

### #quotacheck -cugv /home

برای ایجاد این فایل ها از سوئیچ -c و برای بررسی user quota از سوئیچ -u و برای بررسی group quota از سوئیچ -g استفاده می کنیم. اگر هیچکدام از سوئیچ های -u و -g آورده نشود، تنها فایل مربوط به user quota ایجاد می شود و اگر تنها سوئیچ -g مشخص شده باشد، فقط فایل مربوط به group quota ایجاد می گردد.

-m چک می کند تا ببیند فایل های اصلی ما در پارتیشن مشخص شده ساخته شده است یا خیر. اگر همه چیز صحیح باشد باید در داخل دایرکتوری quota دو فایل به نام های aquota.group و aquota.user ساخته شده باشد. -v هم خروجی کاملی از اجرای دستور به ما نشان می دهد. مجوزهای مربوط به این دو فایل بگونه ای تنظیم شده که فقط کاربر ریشه می تواند به آنها دسترسی پیدا کند.

سپس برای اجرایی شدن پروسه quota بر روی پارتیشن مورد نظر از دستور زیر بهره می بریم. با اجرا و تایپ این دستور ، با هر بار UP شدن سیستم ، سرویس quota هم اجرا می شود:

### #quotaon /home

این سوئیچ به صورت ساده quota را روی پارتیشن مورد نظر اجرا میکند. اگر بعد از این مرحله پیام errorای مشاهده نشد یعنی کار به درستی انجام شده است. و اگر بخواهیم quota را بر روی پارتیشن مشخصی غیر فعال کنیم از فرمان زیر استفاده می کنیم:

### #quotaoff /home

## ساخت قوانین quota

شما می توانید از دستور edquota یا setquota برای ساخت قوانین quota مربوط به یک گروه یا تعدادی از کاربران استفاده کنید. دستور edquota از ویرایشگر متنی vi برای ویرایش فایل های quota یتان استفاده می نماید. برای بکارگیری یک ویرایشگر متفاوت، مقدار متغیر محیطی EDITOR یا VISUAL را پیش از اجرای edquota تغییر دهید. برای مثال ، برای بکارگیری ویرایشگر nano پیش از اجرای edquota دستور زیر را تایپ نمایید:

### #export EDITOR=nano

اگر می خواهید به ازای هر کاربر دیسک را سهمیه بندی نمایید از دستور زیر استفاده کنید:

### #edquota -u UserName

Or

### #edquota -up UserName 1 UserName 2 ... UserName n

اولین دستور بالا تنها برای یک کاربر، فایل را ویرایش می کند و دومین دستور با استفاده از سوئیچ `-p` برای چندین کاربر، همزمان تنظیمات را انجام می دهد. و یا اگر می خواهید به جای اختصاص سهمیه به کاربران آنها را به یک، یا چند گروه اعمال کنید به جای استفاده از گزینه `-u` از گزینه `-g` مانند مثال های زیر استفاده کنید:

```
#edquota -g Group
```

Or

```
#edquota -gp Group 1 Group 2 ... Group n
```

در اینجا با ایجاد یک سهمیه برای یک کاربر به توضیح محتویات فایل ایجاد شده می پردازیم.

```
#edquota -u admin
```

```
Disk quota for user admin (uid 501)
```

filesystem	block	soft	hard	inodes	soft	hard
/dev/sdb2	596	512000	716800	10	800	1000

~

~

~

~

```
"/tmp//Edp.aSt1qgQ" 3L, 219C
```

با اجرای دستور `edquota` برای یک کاربر یا گروهی از کاربران، فایلی باز می شود. اولین ستون، پارتنشینی که `Disk Quota` روی آن فعال شده است را نشان می دهد. در این مثال سهمیه کاربران را می توان برای کاربر `admin` بر روی پارتیشن `/dev/sda2` تنظیم نمود. این فایل شامل تنظیمات تعداد بلاک ها (میزان فضای مصرفی از حجم دیسک که به کاربر یا گروه اختصاص داده می شود.) و تعداد `Inode` های اختصاص داده شده به کاربر یا گروهی از کاربران است.

**نکته:** کوچکترین واحد نرم افزاری ذخیره سازی دیتا در لینوکس `block` است که برابر با 1024 بایت یا 1 کیلوبایت می باشد.

اعداد زیر ستون های `blocks` و `inodes` توسط سیستم ایجاد شده اند و نباید آنها را تغییر داد. اعداد زیر `blocks` نشان دهنده تعداد بلاک هایی است که کاربر تا به حال استفاده کرده و عدد زیر `inodes` تعداد فایل ها و دایرکتوری هایی است که کاربر ایجاد نموده، یعنی فایل ها و دایرکتوری هایی که کاربر مالک آنها می باشد. هم برای `Blocks Limit` و هم برای `Nodes Limit` محدودیت های `Soft` و `Hard` وجود دارد:

**soft limit block:** حداکثر میزان استفاده از دیسک که کاربر یا گروهی از کاربران مجاز هستند را مشخص می کند.

**hard limit block:** میزان مطلق و قطعی استفاده از دیسک را برای کاربر یا گروهی از کاربران مشخص می کند.

**soft limit inode**: حتما باید مشخص شود ولی تعیین **hard limit inode** اختیاری است. مقدار **soft limit** نشان می دهد کاربر به چه میزان مجاز به استفاده از فضای دیسک برای ایجاد فایل می باشد و در صورتی که **hard limit** نیز مشخص شود که مقداری بزرگتر از **soft limit** است و اگر کاربر از محدودیت **soft** عبور کند، وی هفت روز وقت دارد که به مقداری پائین تر از محدودیت تعیین شده باز گردد، در غیر این صورت از استفاده بیشتر یوزر از فضای هارد جلوگیری خواهد شد. **hard** استفاده به صورت قطعی را مشخص می کند و سافت حداکثر مقدار استفاده را نشان می دهد که برخلاف هارد بصورت مطلق نیست.

**نکته**: عدد صفر زیر ستون های **soft** و **limit** یعنی اینکه هیچ محدودیتی وجود ندارد.

سعی در نوشتن بر روی یک پارتیشن پس از اینکه از محدودیت **hard** تجاوز گردید، باعث می شود که عمل نوشتن بر روی هارد دیسک با شکست مواجه شود. وقتی چنین چیزی رخ می دهد، کاربری که سعی دارد تا فایلی را ایجاد نماید که از محدودیت وی تجاوز نموده است، پیامی مشابه با پیام زیر خواهد دید:

```
ide0(3,2): write failed, user block limit reached.
cp: writing 'test.doc': Disk quota exceeded
```

می توانیم بنابر صلاحدید برای تغییر دوره های زمانی تنظیم شده که همان **grace period** است از دستور زیر استفاده می کنیم. این زمانها می تواند بر اساس روز، ساعت، دقیقه و ثانیه تنظیم شود:

**#edquota -t**

## بروزرسانی تنظیمات quota

پس از اینکه تنظیمات **quota** مربوط کاربر یا گروه را تغییر دادید باید دستور **quotacheck** را مجددا اجرا نمایید. همچنین باید برای بروز نگهداشتن رکوردهای **quota** بصورت دوره ای دستور **quotacheck** را اجرا کنید. یکی از روشهای انجام این کار کار اجرای هفتگی این دستور با استفاده از یک ورودی **cron** است.

## بررسی quota ها

برای گرفتن گزارش از مقدار فضا و تعداد **inod**های مصرف شده بر روی سرور توسط کاربران و گروه ها از دستور **repquota** استفاده می شود. در این مثال برای خواننده داده های **quota** مرتبط با تمامی پارتیشن هایی که در حال استفاده از **quota** ها هستند به همراه جریات از سوئیچ های **av** - استفاده میکنیم:

**# repquota -av**

Linux Cookie in Persian v.2

## فصل دهم

## راه اندازی سرویس DHCP در لینوکس



## سرویس DHCP در لینوکس

مقدمه

در شبکه‌هائی با ساختار کوچک، پیکربندی شبکه کامپیوترهای رومیزی به صورت دستی امکان پذیر است اما پیکربندی شبکه سیستم‌ها در ساختارهای بزرگتر امری است مشکل و بعضاً غیر ممکن. حتی در شبکه‌های کوچک، پیکربندی سیستم‌ها با هر تغییری، فعالیت کاربران را مختل و نارضایتی‌هایی را در نحوه پشتیبانی بوجود خواهد آورد. در نتیجه راه حل این است که یک سرور فعالیت آدرس دهی شبکه و پیکربندی را بطور اتوماتیک انجام دهد.

### پروتکل Bootstrap

این پروتکل (BootP) اولین پروتکل پیکربندی است که می‌تواند کلیه اطلاعات برای پیکربندی TCP/IP یک کلاینت، که تقاضای IP نموده را فراهم کند تا براحتمی با سرور شبکه و سایر کلاینت‌ها در ارتباط باشد. مکانیزم عملکرد این پروتکل بدین صورت است که یک بسته به نام BootRequest به پورت‌های UDP شماره ۶۷ همه دستگاه‌های شبکه ارسال می‌کند. کلاینت، همه اطلاعاتی که راجع به خودش دارد را در بسته BootRequest قرار می‌دهد که می‌تواند فقط آدرس لایه فیزیکی باشد. فقط هنگامی که یک سرور BootP یک بسته روی پورت ۶۷ دریافت می‌کند، یک بسته BootReply می‌سازد و داخل آنرا تا آنجایی که بتواند با اطلاعات پیکربندی مورد نیاز کلاینت پر می‌کند. سپس سرور با استفاده از پورت شماره ۶۸ UDP، بسته را روی شبکه انتشار می‌دهد. کلاینت روی پورت ۶۸ منتظر دریافت بسته است. هنگامی که یک بسته روی پورت دریافت کرد که حاوی آدرس فیزیکی‌اش است از اطلاعات داخل بسته برای پیکربندی TCP/IP خود استفاده می‌کند.

### پروتکل DHCP (Dynamic Host Configuration Protocol)

dhcp مخفف Dynamic Host Configuration است و وظیفه آن واگذاری اتوماتیک تنظیمات TCP/IP است. در لینوکس سرویس dhcpd برای همین منظور استفاده می‌شود و پیکربندی آن بسیار ساده و راحت است. یک سرویس dhcpd از یک Scope که در لینوکس به آن range گفته می‌شود به همراه یک سری از option‌ها تشکیل شده است. تعیین range و تعیین آدرس شبکه برای اختصاص دادن به کلاینت‌ها لازم و ضروری بوده اما استفاده از option‌ها اختیاری است.

DHCP روی پورت‌های مشابه BootP یعنی ۶۷ و ۶۸ کار می‌کند و کلیه سرویس‌های BootP به‌علاوه تعدادی کاربرد مهم دیگر را ارائه می‌دهد. یک خاصیت مهم اضافه در DHCP این است که آدرس‌ها می‌توانند بصورت پویا به یک سیستم داده

شوند. آدرس IP که با استفاده از BootP، به یک سیستم تخصیص می‌یابد، به صورت دائم برای آن در نظر گرفته می‌شود و سیستم دیگری در شبکه نمی‌تواند از آن استفاده کند. اما DHCP آدرس را برای یک مدت زمانی مشخص به کلاینت اجاره می‌دهد و بعد از اتمام دوره اجاره، می‌تواند آن آدرس را به کلاینت‌های دیگر واگذار کند. مزیت استفاده از آدرس‌دهی پویا، استفاده بهینه از تعداد کم آدرس است. آدرس‌های بلا استفاده برای استفاده سایر کلاینت‌ها آزاد می‌شود. آدرس‌دهی پویا بوسیله DHCP، مانند همه چیزهای دیگر بدون نقص نیست. از آنجاییکه DNS از آدرس‌هایی که توسط DHCP تخصیص می‌یابد اطلاعی ندارد در نتیجه کامپیوترهای خارج از شبکه داخلی نمی‌توانند سیستمی را که از طریق DHCP، آدرس دریافت کرده را پیدا کنند در نتیجه این سیستم نمی‌تواند به سایر سیستم‌های راه دور سرویس ارائه دهد. این یک نقص است ولی باعث اختلال در شبکه نمی‌شود.

از آنجاییکه اولاً فقط سرورها باید به سایر سیستم‌ها سرویس ارائه کنند، می‌توان پیکربندی شبکه آنها را بصورت دستی تنظیم نمود. دوم اینکه تعداد سرورها نسبت به مجموع تعداد سیستم‌ها کمتر است. در نتیجه هزینه و حجم پیکربندی سرورها به نسبت کمتر می‌باشد. پس نتیجه می‌گیریم که سیستم‌های رومیزی نسبت به سرورها گزینه‌های بهتری برای پیکربندی توسط DHCP هستند. اما تکنیک‌هایی برای انطباق آدرس‌های DHCP و DNS با استفاده از Dynamic DNS (DDNS) وجود دارد که در ادامه این مبحث توضیح داده خواهد شد.

## شروط دریافت IP توسط کلاینت لینوکسی

کلاینت لینوکسی به دو شرط می‌تواند از dhcp آدرس IP و دیگر تنظیمات را دریافت کند که این عکس ویندوز می‌باشد.

1. شرط اول بین مایکروسافت و بین لینوکس مشترک است، آنکه گزینه obtain کارت شبکه فعال شود. البته در لینوکس چون ممکن است محیط گرافیکی نداشته باشیم باید از طریق خط فرمان وارد مسیر و فایل `/etc/sysconfig/network-script/ifcfg-eth0` شده گزینه BOOTPROTO را برابر static قرار دهیم.

2. شرط دوم فقط مخصوص لینوکس است. یک کلاینت لینوکسی به شرطی می‌تواند از dhcp آدرس IP بگیرد که در فایل `/etc/sysconfig/network` گزینه NETWORKING مقداری برابر yes داشته باشد. اگر این گزینه برابر yes نباشد برقرار بودن شرط اول هم نمی‌تواند در گرفتن IP هیچ کمکی به ما نکند.

## مبحث DHCP Failover

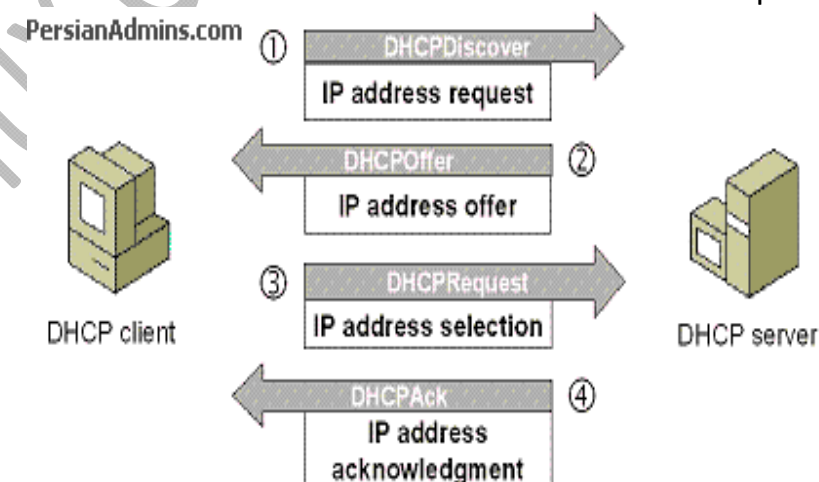
مفهوم failover مانند سرورهای بکاپ می باشد. یعنی اگر سرور اصلی ما down شد سرور دیگری باشد که به درخواستها پاسخ بدهد. در سرورهای مایکروسافت چیزی به نام failover وجود ندارد بلکه می آیند ناحیه ای که می خواهند به آن IP بدهند را به دو scop تقسیم می کنند یعنی یک رنج ip را به دو قسمت تقسیم کرده که اگر یکی از scop ها از کار افتاد scop دیگر بتواند به شبکه مشخصات tcp/ip را واگذار کند. در لینوکس برای failover دو سرور master و slave راه اندازی می شود. به روز رسانی سرور slave از طریق سرور master صورت می گیرد. فاصله زمانی این آپدیت ها هر 10 دقیقه یکبار می باشد. حال اگر برای 10 دقیقه ، سرور master آپدیتی ارسال نکند در دو مرحله توسط سرور slave درخواست آپدیت ارسال می شود اگر جوابی دریافت نکند سرور slave نقش master/slave را با هم اجرا کرده و شروع به سرویس دادن به شبکه می کند. حال اگر سرور master فعال شود ، تا زمانی که دیتابیس آن به روز نشود نقش slave به خود می گیرد و در این نقش باقی می ماند اما به محض آپدیت به نقش قبلی خود بر می گردد.

نکته مهم : تحت هیچ شرایطی نباید سرورهای شبکه ، سوئیچهای قابل مدیریت ، پرینترهای تحت شبکه و کلا دیوایس های مهم شبکه را با dhcp آدرس دهی کنیم بلکه باید آدرس آنها را در سرور dhcp رزرو کرده و به صورت static آدرس را بر روی آنها تنظیم نماییم.

## مراحل Dora

یک کلاینت طی 4 مرحله که به آن Dora گفته می شود از سرور IP اجاره می کند.

1. Dhcp Discovery
2. Dhcp Offer
3. Dhcp Request
4. Dhcp Acknolegment





## Dhcp discovery

در مرحله اول کلاینت بعد از up شدن متوجه می شود با وجود فعال بودن خاصیت obtain هنوز آدرس IP ندارد لذا یک درخواست تولید کرده و آدرس مبدا آن را 0/0/0/0 و آدرس مقصد آن را 255/255/255/255 قرار می دهد. وقتی که آدرس مقصد چهار عدد ، 255 باشد یعنی این بسته از نوع Broadcast بوده و از روی پورت 68 بر روی شبکه ارسال می شود. این پکت در شبکه فریاد می زند که این سیستم IP ندارد آیا کسی هست به آن IP واگذار کند. چون کارت شبکه IP ندارد از آدرس 0/0/0/0 استفاده می کند.

## Dhcp Offer

در دومین مرحله ، اولین dhcp سروری که پکت offer را دریافت کند از scop خودش یک IP به صورت random انتخاب کرده و آن را به صورت unicast از روی پورت 67 به کلاینت پیشنهاد می دهد. علت اینکه پیام سرور unicast میباشد این است که سرور از بسته دریافتی آدرس mac فرستنده را استخراج کرده است به چنین پکتی در دنیای شبکه Dhcp offer گفته می شود.

## Dhcp Request

در مرحله سوم ، اگر کلاینت IP ارسالی از سرور را قبول کند یک پکت به نام dhcp request تولید کرده و به صورت broadcast در شبکه اعلام می کند که این IP متعلق به خودش بوده و آن را قبول کرده است. دلیل این کار این است که اگر dhcp سرور دیگری در شبکه وجود داشته باشد متوجه شود که این کلاینت از چه IP ای استفاده می کند.

## Dhcp Acknowledgment

در مرحله چهارم سرور dhcp با پکت dhcpAck تأیید می کند که IP مورد نظر به چه کلاینتی تعلق دارد . این پکت هم از نوع unicast می باشد.

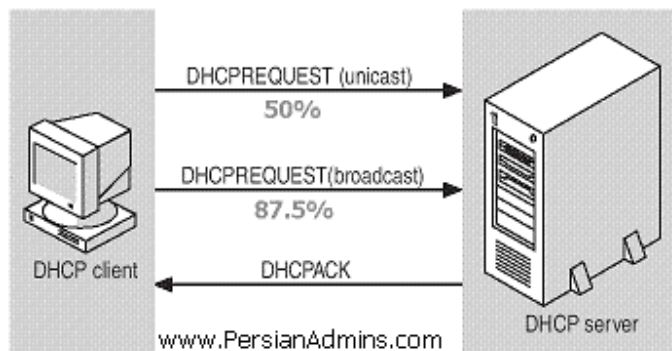
## خلاصه مراحل Dora

پس از اینکه گزینه obtain ip address automatically را در client انتخاب کردیم مراحل زیر اتفاق می افتد :

1. DHCP Client یک بسته DHCP Discover را برای جستجوی DHCP Server ، به صورت همگانی منتشر می کند.
2. DHCP server بوسیله بسته DHCP Offer یک address IP به client تقدیم میکند.
3. DHCP Client یک بسته را که DHCP Request مینامیم به DHCP Server به منظور تحقیق اینکه آیا

DHCP Server معتبر است یا خیر خواه فرستاد. 4. DHCP server بوسیله بسته DHCP acknowledgement جواب client را خواهد داد.

## مکانیزم Lease Duration



IP هایی که سرور dhcp به کلاینتها می دهد دائمی نبوده و اجاره ای می باشد و برای خودشان زمان اجاره دارند که به آن Lease Duration گفته می شود.

در سرورهای مایکروسافتی زمان پیش فرض اجاره IP ، هشت روز ، در سرورهای لینوکسی چهار روز و در سیستم های سیسکو یک روز می باشد.

### زمان T1:

کلاینت وقتی IP را گرفت باید تا 50 درصد زمان پیش فرض که معادل 2 روز می باشد از آن استفاده کند. به این زمان 50 درصدی ، T1 گفته می شود.

کلاینت در روز دوم که مساوی است با همان 50 درصد استفاده شده از IP ، باید نسبت به تمدید زمان اجاره اقدام کند.

### زمان T2:

حال اگر سرور dhcp فعال نباشد که این آدرس را تمدید کند ، کلاینت از این IP استفاده می کند تا به 87.5 درصد زمان اجاره برسد. به این زمان T2 گفته می شود. در این تایم باید زمان اجاره IP تمدید شود، در صورتی که سرور فعال نباشد کلاینت آدرس IP را آزاد کرده و آدرس 0/0/0/0 را می گیرد و مراحل Dora را دوباره انجام داده تا یک

سرور dhcp جدید پیدا کند. در ویندوز دستور `ipconfig /renew` کار تمديد اجاره ip و دستور `ipconfig /release` آدرس قبلی را آزاد کرده و آدرس 0/0/0/0 را به کارت شبکه می دهد تا دوباره دنبال یک سرور dhcp بگردد .

در اینجا مراحل Dora هر 32 و 16 و 8 و 4 و 0 ثانیه یکبار تکرار می شود، اگر سرور dhcp را پیدا نکرد که از آن IP بگیرد کارت شبکه می رود سراغ آدرس های Apipa که از رنج 169.254.0.0 یک IP بگیرد. بعد از اینکه کلاینت از Apipa آدرس IP گرفت هر 5 دقیقه یکبار در شبکه دنبال سرور dhcp می گردد که این کار ترافیک شبکه را بالا برده و کارایی آن را پائین می آورد.

با رنج Apipa نمی توان بر روی اینترنت رفت ، زیرا Apipa فقط IP واگذار می کند و آدرس default gateway اعلام نمی کند و دلیل دیگر این است که رنج Apipa در اینترنت مسیر یابی نمی شود.

## مکانیزم Duplicate Address Detection

کلاینت قبل از اینکه به صورت نهایی IP را روی کارت شبکه اش set کند یک ARP به IP خودش ارسال می کند. با این کار می فهمد آیا IP پیشنهادی از سوی سرور به کلاینت دیگری تعلق دارد یا فقط توسط خودش استفاده می شود . اگر سیستمی از این IP استفاده کند ، کلاینت می فهمد که duplicate اتفاق افتاده و IP تکراری است لذا دوباره مراحل چهار گانه DORA را انجام می دهد. به این عمل DAD گفته می شود.

در این شرایط کلاینت یک پکت به نام dhcp client تولید کرده و به سمت سرور ارسال می کند تا به سرور بفهماند این آدرس Duplicate است و سرور باید یک IP دیگر را پیشنهاد بدهد. DAD یک مکانیزم برای کشف IP های تکراری است.

## نصب و راه اندازی سرویس Dhcp

اکثر توزیع های لینوکس، DHCP را در مراحل اولیه نصب در صورت درخواست کاربر، بر روی سیستم نصب می کنند. اگر لینوکس مورد نظر شما حاوی سرویس DHCP نبود یا یک نسخه جدیدتر از نسخه ای که به همراه سیستم تان است را می خواهید، می توانید به راحتی آخرین ورژن آن را توسط دستور Yum نصب کنید.

dhcpcd با هسته لینوکس ۲.۴ به بالا به خوبی کار می کند اما اگر لینوکس مورد استفاده از هسته قدیمی استفاده می کند امکان دارد مشکلاتی بوجود آید که یکی از آنها مشکل سرویس دهی به کلاینت های ویندوزی است. پس ترجیحاً از نسخه های بالاتر DHCP و کرنل های جدیدتر استفاده کنید.

در بیشتر زمان ها `dhcpd` در موقع نصب سیستم نصب می شود و شما نیازی به نصب مجدد آن ندارید اما اگر به هر نحوی در سیستم موجود نبود طبق روش زیر عمل می کنیم :

ابتدا باید از نصب بودن پکیج `dhcpd` اطمینان حاصل کنیم لذا با دستور زیر از سیستم `query` میگیریم :

```
#rpm -qa | grep dhcp
```

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب `dhcpd` از `yum` استفاده می کنیم :

```
#yum -y install dhcp
```

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج `dhcpd` بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم `query` می گیریم :

```
#rpm -qa | grep dhcp
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm -ql dhcp
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm -qi dhcp
```

سپس با دستور `chkconfig` مشخص می کنیم در چه `run level` هایی فعال باشد :

```
#chkconfig --level 35 dhcpd on
```

```
#chkconfig --list dhcpd
```

و در انتها سرویس را `reset` می کنیم :

```
#service dhcpd restart
```

در اینجا سرویس استارت نمی شود و به ما پیام `Failed` نشان می دهد. دلیل آن هم این است که فایل پیکربندی سرویس `dhcp` خالی است . برای اینکه جلوی `Failed` دادن آن را بگیریم باید فایل کانفیگ آن را مقدار دهی کنیم . این فایل در مسیر `/etc/dhcp/dhcpd.conf` قرار دارد که در ادامه آپشن های موجود در آن و شیوه تنظیمات این سرویس را توضیح خواهیم داد.

## پیکربندی و تنظیمات سرویس Dhcp

همانطور که گفته شد فایل کانفیگ dhcp در مسیر `/etc/dhcp/dhcpd.conf` قرار دارد و درون آن به صورت پیش فرض خالی بوده و کانفیگ خاصی ندارد. علت آن هم این است که سرویس dhcp به محض فعال شدن بدون کانفیگ خاصی شروع به کار کردن می کند، لذا اگر یک dhcp دیگر در شبکه وجود داشته باشد شبکه دچار Collision و چند دستگی می شود.

در لینوکس هزاران فایل راهنما و نمونه برای راه اندازی یک سرویس وجود دارد، dhcp هم از این قاعده مستثنی نیست و برای راه اندازی و کانفیگ این سرویس باید به آنها رجوع شود.

فایل نمونه پیکربندی dhcp در مسیر زیر قرار گرفته است که باید آن را به دایرکتوری اصلی سرویس کپی کنیم:

**# cp/usr/share/doc/dhcp-4.1.1/dhcpd.conf.sample /etc/dhcp/dhcpd.conf**

قبل از کپی، سیستم عامل پیغام می دهد آیا مایل به overwrite فایل موجود هستید یا خیر که ما با تایپ کلمه yes این کار را تأیید می کنیم. بعد از کپی وارد مسیر اصلی فایل کانفیگ شده و آن را با یکی از ادیتورهای موجود باز می کنیم. این فایل از بخش های اصلی و option هایی تشکیل شده است که در پایان هر بخش و یا option علامت سمی کولون (;) قرار می گیرد. option ها میان {} قرار گرفته و تمامی دستورات با حروف کوچک شروع می شوند.

زبان پیکربندی شامل امکانات زیادی است که عملکرد سرور و پروتکل DHCP را کنترل می کند. لغات Allow و Deny کنترل می کنند dhcpd درخواستهای کلاینت ها را چگونه مورد کنترل قرار دهد. هر عبارت با دستور allow یا deny و به دنبال آن یک کلمه کلیدی که بیان می کند چه درخواستی مورد قبول یا رد قرار گیرد، آغاز می شود. سه کلمه کلیدی ممکن، عبارتند از:

1. unknown\_clients
2. bootp
3. booting

**unknown\_clients:** درخواست های پیکربندی، از کلاینت هایی که برای سرور نا آشنا هستند و تعریف نشده اند، می توانند قبول یا رد شوند. به طور پیش فرض این کلاینتها مورد قبول واقع می شوند در غیر اینصورت برخی از قابلیت های DHCP از بین می رود.

**bootp:** سرور می تواند پیکربندی درخواست هایی که از کلاینت های bootp می آیند را قبول یا رد کند. به طور پیش فرض کلاینتهای bootp مجازند که توسط این سرور پیکربندی شوند در نتیجه تمام کلاینتهای bootp و DHCP توسط یک سرور آدرس دهی خواهند شد.

**booting**: عبارت deny booting در قسمت host استفاده میشود و به سرور می گوید که سرور نباید درخواست یک کلاینت مشخص را بررسی کند. پیش فرض این است که درخواست های کلاینت جواب داده شود و احتیاجی به عبارت allow booting نیست. علاوه بر سه عبارتی که در بالا توضیح داده شد، پارامترهای پیکربندی متفاوتی وجود دارند که عملیات پروتکل را کنترل می کنند، در اینجا به برخی از آنها اشاره می گردد.

### # vi /etc/dhcp/dhcpd.conf

با استفاده از فایل dhcpd.conf، میتوان هر اطلاعات پیکربندی که مورد نیاز هر میزبان یا زیرشبکه ای که از سیستم شما سرویس می گیرد را فراهم نمود. در ادامه بعضی از قسمتهای این فایل را توضیح می دهیم:

**ddns-update-style interim;  
ignore client-update;**

این خط معرف ddns یا همان داینامیک dns است. این فیچر به dhcp می فهماند به هر کسی IP داد باید نام آن را در داخل dns ثبت کند. این خط می تواند سه مقدار زیر را داشته باشد:

- **interim**: این گزینه در dhcp های ورژن بالا مورد استفاده قرار می گیرد که به ما فیچرهایی مثل dhcp failover را می دهد و معمولاً هم از همین گزینه استفاده می شود.
- **adhoc**: این گزینه یک فیچر قدیمی است و در dhcp سرورهای قدیمی مورد استفاده قرار می گرفت.
- **none**: این عبارت ویژگی ddns را غیر فعال می کند.

**authoritative;**

از این گزینه برای جلوگیری از حملات dhcp Rouge استفاده می شود. در این نوع از حملات یک dhcp تقلبی توسط یک نفوذگر وارد شبکه شده و باعث چند قسمتی شدن شبکه می گردد. برای جلوگیری از این نوع حملات باید به صورتی به کلاینت ها بفهمانیم که به غیر از سرور اصلی از سرور دیگری IP نگیرند. این گزینه دقیقاً همین کار را انجام می دهد. به طور مثال می توانیم این خط را به این صورت 192.168.10.1 authoritative مقدار دهی کنیم. در این خط مشخص کرده ایم که اگر کلاینتها نتوانستند از سرور اصلی IP بگیرند فقط از این آدرس درخواست IP کنند. می توانیم چندین آدرس را پشت سر هم و یا در چندین خط وارد کنیم.

**default-lease-time 84600;**

زمان اجاره IP بر اساس ثانیه را مشخص می کند. (زمان پیش فرض) اگر کلاینت مدت زمان اجاره را مشخص نکند. سیستم این زمان را به طور پیش فرض برای مدت زمان اجاره یک آدرس در نظر می گیرد.

**max-lease-time 604800;**

بیشترین مدت زمان اجاره IP را بدون در نظر گرفتن مدت زمان درخواستی کاربر بیان می کند. این زمان به ثانیه بیان میشود.

**get\_lease\_hostnames flag false;**

اگر مقدار flag برابر true باشد، dhcpd یک جستجوی معکوس برای هر آدرسی که اجاره می دهد انجام داده و نام میزبان را از DNS دریافت نموده و به کلاینت ارسال می کند. این فرایند می تواند باعث بالا رفتن حجم فعالیت سرور روی شبکه های بزرگ شود. این flag به صورت پیش فرض false است و هیچ جستجویی در DNS صورت نمی گیرد.

## آپشن ها (Options)

آپشن ها تنظیمات اضافه تری هستند که dhcp علاوه بر IP به کلاینتها واگذار می کند. تنظیماتی مثل dns,gw,ntp و .... خطوطی که جزو آپشن محسوب می شوند حتما باید با کلمه options شروع گردند خطوطی مثل :

**option domain-name tohid.com;**

این خط نام domain شبکه را مشخص می کند.

**option domain-name-server 4.2.2.4;**

آدرس dns را مشخص و اعلام می کند.

**option router 10.6.4.1;**

آدرس default gateway را مشخص و اعلام می کند.

**option time-offset +12000;**

این آپشن محدوده زمانی location ما را اعلام می کند.

**option nntp-server 192.168.1.100;**

آدرس تایم سرور را اعلام می کند. اکتیو دایرکتوری در شبکه نقش تایم سرور را به عهده می گیرد.

**option subnet-mask 255.255.255.0**

این آپشن آدرس زیر شبکه را اعلام می کند.

**option broadcast-address 10.10.10.255**

آدرس اعلام همگانی شبکه را به کلاینتها اعلام می کند.

## Option Levels

آپشن ها در محل های مختلفی می توانند اعمال شوند . پنج سطح آپشن وجود دارد که در زیر به ترتیب اولویت آمده است. ترتیب اعمال آپشن ها از بالا به پایین می باشد.

- Host
- Group
- Subnet
- shared-network
- Global

### Option at Host level

معرفی می کنیم مثل رزرو IP. آپشن host، آپشنی است که به یک کلاینت خاص اعمال می شود. مثلاً در این سطح از آپشن می توانیم مشخص کنیم چه IP ای به چه mac آدرسی تعلق بگیرد (Reservation IP) و یا چه مشخصاتی به آن داده شود. این آپشن جزو Policy های امنیتی شبکه می باشد. آدرس اترنت که در عبارت host قرار دارد به عنوان کلیدی برای شناسایی اینکه چه کلاینتی چه اطلاعات پیکربندی دریافت می کند استفاده می شود. در مثال، نام میزبان و آدرس IP کلاینت مشخص شده است. به کلاینت آدرس ایستا داده می شود بنابراین احتیاج به تمدید مدت اجاره آدرس ها ندارند. به غیر از اطلاعاتی که در عبارت host برای کلاینت مشخص شده، dhcpd برای آنها الگو زیرشبکه، نام دامنه، آدرس های سرور DNS و آدرس سرور چاپگر که در قسمت عمومی فایل پیکربندی تعریف شده است را نیز می فرستد.

مثال :

```
host myserver {
    hardware Ethernet xxxxxxxxxxxxxxxx;
    fixed-address 192.168.6.200;
    option router 192.168.6.1;
}
```

### Option at Group Level

این نوع از آپشن برای چند کامپیوتر مشخص شده قابل اعمال است. پیکربندی یک گروه با عبارات host خاتمه می یابد. عبارت گروه از مجموعه ای از عبارات host تشکیل شده است. می توان در عبارات host از پارامتر use\_host\_decl\_name استفاده کرد که برای همه میزبان ها اعمال می شود.

نمونه:

```
group {
    option [use-host-decl-names true];
    host PC1 {
```



```

}
host PC2 {
}
}

```

**Option at Subnet level**: این نوع از آپشن برای یک رنج IP خاص قابل اعمال می باشد. شبکه ای که dhcpd به آن سرویس ارائه می دهد توسط یک آدرس و یک الگو آدرس در عبارت subnet مشخص می شود. dhcpd تنها به کلاینتهایی که در این شبکه قرار دارند و یا مستقیماً توسط عبارات host مشخص شده اند، سرویس ارائه می کند. گزینه ها و پارامترها در عبارت subnet فقط برای زیر شبکه و کلاینتهای آن اعمال می شود عبارت subnet در مثال زیر دارای دو محدوده برای ساخت چند گروه مجزا برای آدرس های پویا است. این نشان می دهد، می توان یک فضای آدرس گسسته با استفاده از چند عبارت range داشت.

نمونه:

```

subnet 192.168.6.0 netmask 255.255.255.0 {
    range 192.168.6.1 192.168.6.200;
}
subnet 192.168.7.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.7.1 192.168.7.200;
}
subnet 192.168.9.0 netmask 255.255.255.0 {
    range 192.168.9.1 192.168.9.19;
    range 192.168.9.31 192.168.9.200;
    option domain-name-server 4.2.2.4;
    option router 192.168.9.250;
}

```

در قسمت آخر ما از IP، 20 تا 30 را جدا کردیم. آپشن هایی که در زیر هر subnet آورده شوند، مختص به همان ساب نت بوده و فقط به آن اعمال می شود. آپشن های هر ساب نت باید بین دو اکولاد اصلی آن قرار بگیرند تا به آن اعمال شوند.

**نکته:** اگر پارامتر range در یک عبارت بیان شده باشد و هر کلاینت DHCP در subnet ای درخواست آدرس نماید، تا زمانیکه آدرس موجود باشد، به آن کلاینت آدرس مورد نظر اختصاص می یابد و اگر پارامتر range بیان نشده باشد آدرس دهی پویا فعال نمی شود.

## Option at Shared-network level

تعداد رنج IP ، می توان بر روی کارت شبکه IP تعریف کرد لذا با یک سرور می توان چندین رنج آدرس واگذار کرد. بهترین مثال برای این قسمت تقسیم IP در یک سوئیچ با vlan های مختلف است.

نمونه :

```
shared-network tohid {
  option
    subnet 192.168.5.0 netmask 255.255.255.0 {
    .
    .
    .
    .
    .
  }
  subnet 192.168.6.0 netmask 255.255.255.0 {
  .
  .
  .
  .
  .
}
}
```

## Option at Global level

option هایی که بین آکولاد باز/ بسته {} باشد تنها در همان حوزه اثر دارد اما می توان option هایی را نیز تعریف کرد که بصورت سراسری یا global باشند. فرض کنید یک option مانند routers که default gateway را نشان می دهد هم بصورت سراسری و هم بصورت محلی ( داخل آکولاد باز/ بسته ) باشد اما مقادیرشان با هم متفاوت است. در این صورت اولویت با option محلی است یعنی آن option ی که در حوزه محلی می باشد الویت بالاتری نسبت به option سراسری دارد.

## شروع به کار سرویس Dhcp

بعد از پایان تعاریف و تنظیمات ابتدا باید از صحت درستی تغییرات اطمینان حاصل کرده و سپس سرویس را استارت کنیم. برای این کار از دستور `dhcpcd` بهره می‌بریم. این دستور صحیح بودن تنظیمات فایل `Config` را بررسی کرده و در صورتی که مشکلی وجود داشته باشد آن را نشان خواهد داد. این دستور به طور معمول دو `error` را گوش زد می‌کند که در زیر به توضیح آنها پرداخته ایم:

1. No Configured to listen on any interfaces
2. No Subnet declaration for ethxx

**No Configured to listen on any interfaces:** معنی این پیغام این است که ما هیچ کارت شبکه ای را روی سرور مشخص نکرده ایم تا به پکتهای Dora گوش کند. با `edit` فایل زیر برای آن اینترفیس لازم را مشخص می‌کنیم:

```
# vi /etc/sysconfig/dhcpd
DHCPDARGS=ethxxx
```

در جلوی این عبارت ، نام کارت شبکه را بدون فاصله وارد می‌کنیم. ممکن است سرور 10 عدد کارت شبکه داشته باشد ولی آن کارت شبکه ای که اینجا مشخص می‌کنیم می‌تواند به کلاینتها IP واگذار کند. در ویندوز به این کار `Binding` گفته می‌شود. در لینوکس حتما باید کارت شبکه ای که IP پخش می‌کند را مشخص کنیم که این بر خلاف ویندوز است.

**No Subnet Declaration for ethxx:** این پیام هم می‌گوید کارت شبکه سرور روی یک رنج آدرس قرار دارد اما هنوز `subnet` ای برای آن تعریف نشده است. بعد از برطرف کردن اشکالات نشان داده شده دوباره دستور `dhcpcd` را اجرا می‌کنیم. این دستور علاوه بر اینکه فایل `Config` سرویس را خوانده و صحت آن را چک و سرویس را `start` می‌کند. لذا دیگر نیازی به استفاده از دستور `start` نیست.

**نکته:** سرویس `dhcpcd` به صورت `Standalone` کار کرده و زیر مجموعه `init` محسوب می‌شود.

## آمار IP های واگذار شده

`dhcpcd` آمار آدرس های واگذار شده را در فایل `dhcpcd.leases` نگهداری می‌کند که در مسیر `/var/lib/dhcpcd/dhcpcd.leases` قرار دارد. اگر این فایل وجود نداشته باشد سرویس

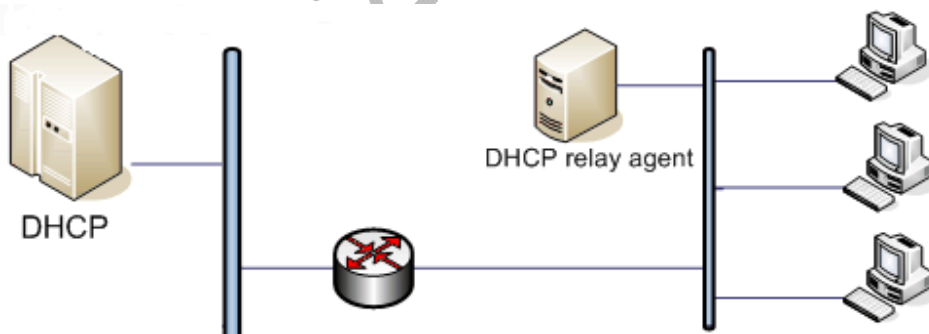
دچار مشکل جدی خواهد شد. هنگامی که سرور برای اولین بار نصب و استارت می شود یک فایل خالی `dhcpd.lease` ایجاد میکند تا سرویس از ابتدا به درستی شروع به کار کند. لازم به ذکر است که فایل `lease` فقط توسط `dhcp` استفاده شده و تغییر می یابد و احتیاجی به تغییرات کاربر ندارد. کاربر فقط باید فایل `dhcpd.conf` را مورد پیکربندی قرار دهد.

در مسیر `/var/lib/dhcpd` سه فایل به نام های زیر وجود دارد:

```
dhcpd.leases
dhcpd.leases~
dhcpd.leases.rpmshave
```

آمار آدرس های واگذار شده به کلاینتها در فایل `dhcpd.leases` قرار دارد. حال اگر سیستم به هر نحوی `crash` کند و این فایل از بین برود در `up` مجدد سیستم ، `dhcp` نمی تواند آمار IP های واگذار شده را بدست بیاورد چون دیتابیس آن پاک شده است. لذا خود سرویس در یک زمان بندی منظم می آید از این فایل یک کپی گرفته و در فایل `dhcpd.leases~` ذخیره می کند تا در صورت بروز مشکل و از بین رفتن فایل اصلی بتواند به دیتابیس خود دسترسی داشته باشد.

## مفهوم DHCP Relay Agent



`relay agent` درخواست هایی که از زیر شبکه های بدون `dhcp` ارسال می گردد را برای `dhcp` سرورهای دیگر که مشخص شده اند بازپخش میکند.

دلیل استفاده از `DHCP Relay Agent` اینست که مسیریاب، بسته های `Broadcast` را از خود عبور نمیدهد و کلاینت ها برای گرفتن IP باید مراحل `Dora` را طی کنند و چون پکتهای `Dora` از نوع `Broadcast` است روترها آنها را از خود رد نمی کنند، پس برای حل این مشکل ما از یک `DHCP Relay Agent` استفاده میکنیم .

DHCP Relay Agent بدین ترتیب عمل میکند، ابتدا پیغام هایی که به صورت Broadcast ارسال میشود و تقاضای IP میکنند را جمع آوری کرده و سپس خود بصورت unicast با DHCP سرور درخواست مورد نظر را مطرح کرده سپس جواب گرفته شده را برای client ارسال میکند. در واقع نقش میانجی را ایفا می کند.

برای edit فایل پیکربندی به مسیر زیر رفته و فایل dhcrelay را باز می کنیم:

```
# vi /etc/sysconfig/dhcrelay
```

مهمترین قسمتهای این فایل گزینه های زیر می باشند:

**INTERFACE:"eth1"**

در اینجا مشخص می کنیم این کارت شبکه ترافیک Broadcast را دریافت کند.

**DHCPSEVER:"192.168.7.1"**

و در این خط آدرس سرور dhcp را وارد می کنیم. با تغییر این دو خط ، این سرویس فعال می شود. حال باید یک استاتیک route بنویسم و مشخص کنیم Relay Agent پکتها را به چه IP ای بدهد و در انتها سرویس را ریست کنیم:

```
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254
```

```
# service dhcrelay start
```

در این خط route مشخص کردیم پکتهای Broadcast درخواست IP را به ورودی روتر بفرستد. اگر سرویس net را ریست کنیم ، خطی که به عنوان route دستی نوشتیم پاک می شود و route پیش فرض سیستم load می شود.

## تنظیمات سرور و کلاینت dhcp

در **سرور** باید یک کارت شبکه را بعنوان سرویس دهنده DHCP انتخاب کنیم. حتی اگر چند کارت شبکه داشته باشیم تنها یکی از آنها باید برای سرویس DHCP استفاده شود و روی آن یک IP بصورت استاتیک (ثابت) تنظیم گردد. هر کارت شبکه در مسیر /etc/sysconfig/network-scripts/ یک فایل به نام ifcfg-ethx دارد که x شماره کارت شبکه می باشد و از عدد صفر شروع می شود. برای کانفیگ اولین کارت شبکه با نام eth0 از دستور زیر استفاده می کنیم:

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

برای دادن یک آدرس IP ثابت به کارت شبکه باید متغیرهای زیر را با مقادیر مناسب تکمیل کنیم. این عبارات در زیر توضیح داده شده است:

IPADDR: همان آدرس IP ثابتی است که می خواهیم به کارت شبکه اختصاص دهیم.

NETMASK: همان subnet mask است.

**NETWORK:** معرف آدرس شروع شبکه است و با توجه به range مشخص می گردد مثل آدرس 10.10.10.0.

سه عبارت بالا باید با مقادیر صحیح مقدار دهی شوند اما تغییراتی هم باید در فایل اعمال شود:

- ابتدا باید مقدار عبارت BOOTPROTO به static تغییر پیدا کند.

- سپس عبارت ONBOOT باید به yes تغییر پیدا کند.

بعد از پایان تنظیمات سرویس شبکه را ریست کنید تا تغییرات اعمال شود.

تنظیمات بخش **کلاینت** بسیار ساده است و حتی می توان گفت تنظیم خاص هم ندارد و تنها باید در فایل ifcfg-eth0 در

هر کلاینت مقدار BOOTPROTO برابر با dhcp باشد.

**BOOTPROTO=dhcp**

اگر بخواهیم به صورت موقت تیک گزینه Obtain کارت شبکه را بزنیم از دستور زیر استفاده می کنیم:

```
# dhclient eth0
```

و اگر از دستور زیر استفاده کنیم به محض اینترنت از dhcp آدرس IP میگیرد:

```
# ifconfig eth0 dynamic
```

به محض تایپ این دستور IP ای که به صورت استاتیک تنظیم شده از بین می رود و با ریست سیستم دوباره به شرایط قبل باز می گردیم.

اگر بخواهیم در ویندوز با کامند ، کارت شبکه را وادار کنیم که از dhcp سرور IP بگیرد از این دستور استفاده می کنیم :

```
netsh int ip set "local area connection" dhcp
```

حال اگر بخواهیم زمان 50% درصد را شبیه سازی (تمدید) کنیم دوباره از دستور dhclient استفاده کرده و اگر بخواهیم

زمان 87/5% شبیه سازی کنیم (درخواست IP جدید) کافی است به انتهای آن r- را اضافه کنیم. دستور dhclient همان درخواست IP را ارسال می کند:

```
# dhclient -r eth0
```

معادل ویندوزی آن دستورات ipconfig /renew و ipconfig /release می باشد.

اگر هم بخواهیم به صورت دائمی تیک Obtain کارت شبکه را فعال کنیم تا همیشه از سرور dhcp درخواست آدرس کند باید فایل مربوط به اینترفیس شبکه را باز کرده و خطوط زیر را اصلاح کنیم :

```
# vi /etc/sysconfig/network-script/ifcfg-eth0
```

```
BOOTPROTO=dhcp
```

```
PEERDNS=yes
```

اگر عبارت PEERDNS برابر yes نباشد تنظیمات dns ای که از dhcp دریافت می شود در این فایل ذخیره نشده و به

سیستم عامل اعمال نمی شود. و می توانیم با تغییر در فایل زیر مشخص کنیم کدام اینترفیس بتواند از سرور IP بگیرد:

```
# vi /etc/sysconfig/dhcpd
DHCPDARGS=eth0;
DHCPD_INTERFACE="eth0";
```

## ساختن یک فایل نمونه

در انتها یک نمونه فایل پیکربندی سرور dhcp آمده است :

```
# Define global values that apply to all systems.
max-lease-time 604800;
default-lease-time 86400;
option subnet-mask 255.255.255.0;
option domain "foobirds.org";
option domain-name-servers 172.16.55.1, 172.16.5.1;
option pop-server 172.16.18.1;
# Define the dynamic address range for the subnet.
subnet 172.16.55.0 netmask 255.255.255.0 {
option routers 172.16.55.1;
option broadcast-address 172.16.55.255;
range 172.16.55.64 172.16.55.192;
range 172.16.55.200 172.16.55.250;
}
# Use host statements for clients that get static addresses
group {
use-host-decl-names true;
host kestrel {

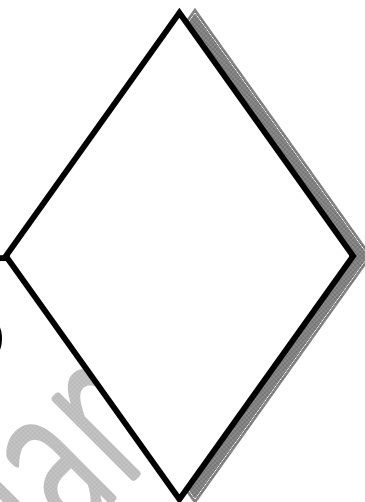
hardware ethernet 00:c7:aa:a8:04;
fixed-address 172.16.55.4;
}
host ibis {
hardware ethernet 00:00:c0:a1:5e:10;
fixed-address 172.16.55.16;
}
```





فصل دهم

بررسی Xinetd



## xinetd در لینوکس

اگر init نباشد یا درست کار نکند می توان گفت که عملاً سیستم غیر عملیاتی است. چون هر نرم افزاری بخواهد کار کند init هم به طریقی در گیر می شود. اما وظیفه init نیست که ping کند یا telnet راه اندازی کند بلکه وظیفه اش این است که یک سری نرم افزار را وارد چارت سازمانی سیستم عامل کند:

اسم مدیر شبکه در لینوکس xinetd یا super server است. نرم افزارهای تحت سیستم عامل Linux به دو صورت زیر به کاربران تحت شبکه سرویس می دهند:

- به صورت مستقل (Standalone)
  - تحت نظارت و کنترل پروسس xinetd
- بسیاری از سرویس های شبکه ای از جمله telnet تحت نظارت و سرپرستی پروسسی به نام xinetd که اصطلاحاً "Super Server خوانده می شود قرار دارند.

سوپر سرور یک سرویس قدرتمند است که سرویس های دیگر می توانند در قالب آن کار کنند. شما باید در سازمانتان تصمیم بگیرید که کدام یک از نرم افزارهایتان زیر نظر init باشد و کدام یک زیر نظر xinetd کار کند.

xinetd فقط مدیریت درخواست های یک سرویس را به عهده می گیرد، مدیریت کارهای یک سرویس به عهده خود آن سرویس است. اگر نرم افزاری به تنهایی در حافظه قرار بگیرد در مد Standalone قرار دارد.

xinetd نرم افزاری است که می تواند نرم افزارهای سرویس دهنده شبکه را مدیریت کند. مثلاً وقتی در لینوکس آپاچی سرور را راه اندازی کردیم در قسمت از پارامترهای config می توانیم مشخص کنیم که آیا به صورت standalone اجرا شود یا تحت نظر xinetd اجرا شود.

هرکس از هر جای دنیا به هر سرویسی وصل شود سرویس دهنده باید به پورت گوش کند. نرم افزار stand alone خودش گوش می کند اما نرم افزارهایی که فرزند xinetd هستند xinetd به جایشان به پورت گوش می دهد و در مواقع ضروری آنها را آگاه می کند. همه نرم افزار های جدی در همه سیستم عامل ها فایل پیکربندی دارند. سرویس ها زمانی که بالا می آیند فایل conf مربوط به خود را می خوانند تا بدانند چه کاری باید انجام دهند.

سرویس xinetd که در بعضی از گونه های Linux و یا Unix با نام inetd شناخته می شود در زمان فعال شدن، به فایل ها و دایرکتوری زیر مراجعه نموده و با آنالیز نمودن اطلاعاتی که به دست می آورد آماده سرویس دهی می شود:

**etc/xinetd.conf/**

**etc/xinetd.d/**

فایل متنی xinetd.conf که همانند اکثر فایل های پیکربندی تحت دایرکتوری etc میباشد، شامل اطلاعات کلی برای سرویس دهی تحت شبکه بوده و تحت دایرکتوی etc/xinetd.d/ نیز تعداد زیادی فایل متنی قرار داشته و به ازای هر

سرویس (مثلاً "telnet") میتوان فایل متنی با همان نام مشاهده نمود. مدیر سیستم با تغییر دادن در فایل های فوق می تواند کنترل بیشتری را بر روی سرویس دهی داشته باشد. در لینوکس اکثر فایل های پیکربندی text base هستند و با ویرایشگرهای متن مانند Vim قابل تغییر هستند ولی در ویندوز این نوع فایل ها داخل رجیستری قرار دارند و با regedit باید آنها را مورد تغییر قرار داد .

shell مانند یک گارسون در رستوران است ، پشت صحنه عوامل بسیار زیادی فعالیت می کنند که آن رستوران میتواند سرویس بدهد به این عوامل Daemon می گویند که یک معنی آن پشت پرده است. xinetd هم خودش هیچ وقت شخصا برای سرویس دادن عمل نمی کند ، مانند سرپرست راننده هاست که اگر شخصی تماس بگیرد و ماشین می خواهد سرپرست هم برایش ماشین می فرستد.

در فایل xinetd.conf داخل {} مانند برنامه های C پارامترها و مقادیرشان تعریف شده اند که xinetd از روی اینها موقع بالا آمدن می فهمد که به چند نفر باید سرویس بدهد و مثلا صد نفر می توانند telnet کنند یا 50 نفر ftp کنند. در واقع xinetd فایلی دارد که از روی آن می تواند بخواند که چه کسی می تواند چه کسی نمی تواند و چه کسی نباید بتواند! همان طور که موقع ورود نگهبان از شما کارت تشخیص هویت می خواهد xinetd نیز برای هر ارتباطی identification طلب می کند و مجموعه این اطلاعات را نیز نگه داری می کند.

اگر کسی telnet کند xinetd هاستش (ip) را نگه می دارد و اینکه با چه کسی کار دارد. در نگهبانی نیز شماره شناسایی مراجعه کننده و شماره کارمندی کسی را که با او کار دارد ثبت می شود. حال اگر شخص شماره شناسایی نداشت log out failure اتفاق می افتد ولی سیستم هاست را ثبت می کند و همه این کارها توسط xinetd انجام می شود. به همین دلیل است که اگر به سایتی حمله کنید شناسایی می شوید ؛ userID ندارید ولی ip شما ثبت می شود. در لینوکس های سری 6 به بعد خانواده ردهت سرویس xinetd نصب نیست بلکه باید نصب و پیکربندی شود.

## مزایا و معایب xinetd

به طور مثال اگر روی سروری 100 سرویس داشته باشیم و هر سرویس به تنهایی بخواهد در حافظه قرار بگیرد مطمئنا آن سرور در سرویس دهی با مشکل جدی مواجه خواهد شد .

xinetd سرویس هایی که ضروری نیستند را از حافظه خارج کرده و مدیریت درخواست های آنها را به عهده گرفته و به جای آنها در حافظه قرار می گیرد. و سرویس مورد نظر را به حالت standby می برد . به محض اینکه درخواستی برای آن سرویس برسد xinetd آن را فراخوانی کرده و سرویسش مورد نظر را وارد چرخه سرویس دهی می کند. لازم به ذکر است تمام سرویس ها را نمی توان تحت xinetd راه اندازی کرد ، مثلا سرویس های پر تراکشی مثل dhcp و dns را باید به

صورت standalone راه اندازی کرد. اما سرویس های مثل ssh و ftp و یا telnet که listen دارند را می توان تحت xinetd اداره کرد. سرویس های جدی و پرتراکنش ، حتما باید به صورت standalone راه اندازی شود.

با xinetd یک لایه امنیتی به سیستم اضافه می شود و دست admin برای اعمال محدودیت در سرویس ها باز می شود. اگر سرویسی که دارای تراکنش بالائی باشد را تحت xinetd قرار دهیم مطمئنا خود xinetd با مشکل جدی مواجه خواهد شد. کانفیگ xinetd مربوط به سرویس خاصی نیست. این فایل در مسیر /etc/xinetd.conf قرار دارد و در /etc/xinetd.d هم فایل های کانفیگ سرویس هایی که تحت xinetd اداره می شوند و می خواهند یکسری خواص گلوبال را به ارث ببرند قرار دارد.

## بررسی آپشن های موجود در یک فایل xinetd

در زیر تعدادی از آپشن هایی که در یک فایل xinetd میتوان نوشت را توضیح می دهیم جهت اطلاع بیشتر به man xinetd رجوع فرمائید. مقادیر ثبت شده در جلوی هر آپشن فرضی می باشد.

**instances =60**

این آپشن مشخص می کند سرویس به چند درخواست هم زمان پاسخ می دهد. instances =60 یعنی به بیش تر از 60 نفر را سرویس نمی دهد حالا اگر 45 نفر telnet کنند و 15 نفر ftp نفر 61م که تلاش کند وصل شود با پیغام connection refused مواجه می شود. این عدد را در یک سازمان مثلا برابر 600 می گذاریم و برای شبکه های خانگی و کوچک 2 یا 3. پس این عدد در سازمان ها برای سرویس هایی که stand alone نیستند باید عوض شوند.

**log\_type =SYSLOGautjpriv**

این خط مشخص می کند تمام لاگها را تحویل syslog سرور بدهد و log هایی را ثبت می کند که از نوع احراض هویت باشند.

**log\_on\_success =HOSTPID**

اگر کلاینت موفق شود از سرویس مورد نظر استفاده کند از id پروسس های آن log گرفته می شود.

**log\_on\_failure =HOST**

اگر کلاینت موفق به سرویس گرفتن نشد از IP یا نام سیستم log برداری می کند.

**cps =25 30**

در جلوی این آپشن دو عدد وجود دارد اولین عدد مشخص می کند چند درخواست هم زمان را در ثانیه بپذیرد و دومین عدد هم بیان می کند اگر ارتباط برقرار نشد چند ثانیه بعد مجدداً **retry** کند. عدد دوم همان زمان **time out** است در این خط مشخص کرده ایم در ثانیه 25 درخواست بیشتر نمی توانند به سیستم وصل شوند حال اگر درخواست 26 رسید باید 30 ثانیه صبر کند تا اتصال برقرار شود. این کار برای جلوگیری از حملات **DDOS** می باشد. عدد **cps** باید متناسب با **instances** باشد. در مورد **cps** یا **connection per second** باید توجه شود که اگر 10000 کاربر ظرفیت داشته باشیم و همه با هم وصل شوند مثل این است که 10000 نفر یک دفعه وارد یک اتاق شوند پس منطقی این است که هر دفعه (ثانیه) مثلاً 25 نفر وارد شوند. اکثر سرویس های موجود در دایرکتوری **xinetd** (xinetd.d) تحت مدیریت **xinetd** اجرا می شوند مانند **telnet** ، **ftp** و **chargen** و بقیه سرویس ها نیز شبیه این سه سرویس کار می کنند. به ازای هر سرویسی در دایرکتوری **xinetd.d** یک فایل کانفیگ داریم.

**disable =yes or no**

اگر **yes** باشد سرویس غیر فعال می شود و اگر **no** باشد بر عکس آن را اجرا می کند.

**wait =yes or no**

این آپشن مشخص می کند نرم افزار مربوطه **Multi Thread** باشد یا **Single Thread** . اگر **yes** باشد باید ابتدا به درخواست رسیده پاسخ دهد سپس به درخواست بعدی رسیدگی کند، اما اگر **no** باشد می تواند هم زمان به چند درخواست پاسخ دهد.

**server =/usr/sbin/sshd**

مقابل این گزینه باید آدرس اسکریپت باینری فایل اجرایی سرویس مورد نظر را وارد کنیم. این فایل حتماً باید باینری باشد یعنی اگر یک سرویس نوشتید باید فایل آن را تبدیل به باینری کنید.

**user =root**

مشخص می کند اسکریپت اجرایی سرویس توسط چه یوزری اجرا می شود.

**sock\_type =stream or dgram**

اگر مقدار این خط بر روی **stream** تنظیم شود یعنی پکتها از نوع **tcp** می باشند و اگر بر روی **dgram** باشد یعنی پکت های این سرویس از نوع **udp** خواهند بود.

**server\_args =/etc/ssh/sshd\_conf**

هر سرویسی یکسری آرگومان دارد که ممکن است xinetd آنها را نشناسد. مثلاً در یک سرویس می‌توانیم کاری کنیم که اگر یوزر یک دایرکتوری یا فایل ایجاد کرد با پرمیژن خاصی ذخیره شود. این می‌تواند قابلیت داخلی یک سرویس باشد ولی ممکن است در xinetd وجود نداشته باشد. اگر بخواهیم این قابلیت‌ها را به xinetd معرفی کنیم باید در جلوی این آپشن آدرس مسیر فایل کانفیگ سرویس مورد نظر را وارد می‌کنیم چون معمولاً آرگومانها و فیچرها در فایل کانفیگ یک سرویس تعریف می‌شوند.

**bind =192.168.1.1**

این آپشن معلوم می‌کند این سرویس درخواست‌های رسیده از کدام کارت شبکه را پاسخ دهد.

**access\_time =19:12-19:16**

در این خط مشخص می‌کنیم کاربران در چه زمانهای بتوانند از این سرویس استفاده کنند.

## نمونه ای از فایل ایجاد شده در Xinetd

سرویس vsftpd به صورت standalone کار می‌کند حال اگر بخواهیم این سرویس زیر نظر xinetd اداره شود باید گزینه listen=??? را در فایل کانفیگ آن برابر YES شده و در زیر دایرکتوری xinetd یک فایل کانفیگ بسازیم. در زیر نمونه ای از فایل آورده شده است :

Servicevsftp

```
{
socket_type      =stream      یعنی پکتها از نوع tcp می باشند
user             =root        مشخص میکند اسکریپت اجرایی سرویس توسط چه یوزری اجرا می شود
server           =/usr/sbin/vsftpd آدرس اسکریپت باینری فایل اجرایی سرویس مورد نظر
server_args      =/etc/vsftod/vsftpd.conf آدرس فایل کانفیگ جهت شناختن آرگومانها
nice             =10
disable          =no          نشان می دهد سرویس فعال است
flags            =ipv4
}
```

Linux Cookie in Persian v.2

ضمیمه یک

اضافه کردن Repository به لینوکس



# اضافه کردن Repository به لینوکس

## 1- اضافه کردن repository اینترنتی

موقعی پیش می آید نیاز به نرم افزاری دارید ولی نام آن را نمی دانید مثل نرم افزارهای پلیر یا pdf ، برای پیدا کردن نرم افزار مورد نیاز می توانید از yum و مخازن اینترنتی لینوکس استفاده کنید به طور مثال برای پیدا کردن نرم افزار نمایش pdf از دستور جستجو yum به صورت زیر استفاده می کنیم .

**# yum search pdf**

با تایپ این دستور ، yum ابتدا دیتابیس خود را که متشکل از لیست rpm مخازن مشخص شده است را به روز می کند و سپس لیست تمام پکیج هایی که کلمه pdf در آن وجود دارد را نمایش می دهد. مخازن اینترنتی مکانهای ذخیره سازی هستند که نرم افزارهای لینوکس در آنجا نگهداری می شوند. لیست این مخازن در سیستم عامل قابل تغییر است.

بر روی اینترنت مخازن زیادی وجود دارد ولی نباید همه و یا تعداد زیادی از آنها را بر روی سیستم add کرد ، چون هم می تواند خطر امنیتی ایجاد کرده و هم عملکرد پروسه yum را به شدت سنگین می کند . از مهمترین و بهترین مخازن لینوکسی ، می توان به مخزن base و epel اشاره کرد آدرس مخزن base بر روی تمام توزیع های مبتنی بر ردهت وجود دارد ولی epel به صورت پیش فرض بر روی سیستم نصب نیست و باید آن را به صورت دستی نصب کنیم . برای نصب آن ابتدا به آدرس زیر می رویم :

<https://fedoraproject.org/wiki/EPEL>



در پائین صفحه باز شده قسمتی به نام how can I use these extra packages? وجود دارد .


## How can I use these extra packages?


EPEL has an 'epel-release' package that includes gpg keys for package signing and repository information. Installing this package for your Enterprise Linux version should allow you to use normal tools such as yum to install packages and their dependencies. By default the stable EPEL repo is enabled, there is also a 'epel-testing' repository that contains packages that are not yet deemed stable.





### NOTE for RHN users.

You need to also enable the 'optional' repository to use EPEL packages as they depend on packages in that repository. This can be done by enabling the RHEL optional subchannel  for RHN-Classic. For certificate-based subscriptions see [Red Hat Subscription Management Guide](#) .

If you are running an EL7 beta version, please visit here to get the newest 'epel-release' package for EL7: [The newest version of 'epel-release' for EL7](#) 

If you are running an EL6 version, please visit here to get the newest 'epel-release' package for EL6: [The newest version of 'epel-release' for EL6](#) 

If you are running an EL5 version, please visit here to get the newest 'epel-release' package for EL5: [The newest version of 'epel-release' for EL5](#) 

You can verify these packages and their keys from the Fedora project's keys page: <https://fedoraproject.org/keys> 

با توجه به سری سیستم عامل بر روی یکی از لینکها کلیک کرده و وارد صفحه بعد می شویم :

<http://mirror.switch.ch/ftp/mirror/epel/6/i386/repoview/epel-release.html>

## epel-release - Extra Packages for Enterprise Linux repository configuration

**Website:** <http://dl.fedoraproject.org/pub/epel/>

**License:** GPLv2

**Vendor:** Fedora Project

### Description:

This package contains the Extra Packages for Enterprise Linux (EPEL) repository GPG key as well as configuration for yum and up2date.

### Packages

[epel-release-6-8.noarch](#) [14 KiB] **Changelog** by (2012-11-04):  
- Fix URL bz #870686

با ورود به صفحه جدید به قسمت Packages و بر روی لینک دانلود رفته آدرس آن را کپی کرده و سپس در صفحه terminal توسط دستور yum یا rpm آن را نصب می کنیم .

# rpm -Uvh <http://mirror.switch.ch/ftp/mirror/epel/6/i386/epel-release-6-8.noarch.rpm>

سیستم به صورت اتوماتیک پکیج مورد نظر را دانلود و نصب می کند. لیست repository ها در مسیر `/etc/yum.repo.d/` قرار دارند که پسوندشان `.repo` می باشد.

داخل این پکیج 5 فایل قرار دارد که با دستور زیر قابل مشاهده است :

```
# rpm -qlp epel-release-6-8.noarch.rpm
```

و با دستور زیر می توانیم اطلاعات کاملی در مورد این پکیج بدست آوریم :

```
# rpm -qip epel-release-6-8.noarch.rpm
```

## 2- اضافه کردن repository محلی

در بعضی از سرورها دسترسی به اینترنت برای نصب پکیجها و یا هر بسته دیگری امکان پذیر نیست و همانطور که می دانید نصب rpm بدون نصب نیازمندی ها بسیار مشکل است. لذا باید کاری کنیم yum بدون وابستگی به اینترنت و از روی یک مخزن محلی ، نرم افزار مورد نظر را به همراه نیازمندی های آن نصب کند. برای این کار یا باید تمام پکیج های مورد نیاز را جمع آوری کرده و در قسمتی از هارد ذخیره کنیم و یا تمامی محتویات dvd نصب سیستم عامل را در مکانی از هارد کپی کنیم پس طبق روش زیر ابتدا به مسیر `/etc/yum.repos.d` می رویم :

```
# mkdir -p /opt/dvd/pkg
```

```
# cp -rf /media/dvdrm-name /opt/dvd
```

```
# cp -rf /media/dvdrm-name /RPM-GPG-KEY* /opt/dvd/gpg
```

```
# cd /etc/yum.repo.d/
```

```
# mv CentOS_Media.repo dvd.repo
```

```
# vi dvd.repo
```

```
[dvd]
```

```
name=dvd
```

```
baseurl=file:///opt/dvd/
```

```
gpgcheck=1
```

```
enabled=1
```

```
gpgkey=file:///opt/dvd/gpg/RPM-GPG-KEY-CentOS6.0
```

در صورت استفاده در شبکه ، می توانیم تنظیمات زیر را در سیستم های مورد نیاز اعمال کنیم :

```
[dvd]
name=dvd
baseurl=http://ip-address/ opt/dvd/
gpgcheck=1
enabled=1
gpgkey=http://ip-address/pkg/gpg/RPM-GPG-KEY-fedora
```

تغییرات را ذخیره کرده و فایل را می بندیم. سپس باید لیست repoهای سرویس yum را به روز کنیم تا منبع جدید را بشناسد، لذا با دستور زیر دیتابیس yum را به روز کرده و repo جدید را وارد cache سرویس yum می کند:

```
# yum repolist
# yum clean metadata
```

سپس با دستور زیر هر پکیجی را که بخواهید می توانید با دستور yum بدون نیاز به اتصال به اینترنت نصب کنید.

```
# yum -y install bind --disablerepo=* --enablerepo=dvd
```

Linux Cookie in Persian v.2

## بازیابی پسورد یوزر root در لینوکس



## بازیابی رمز عبور یوزر root در لینوکس

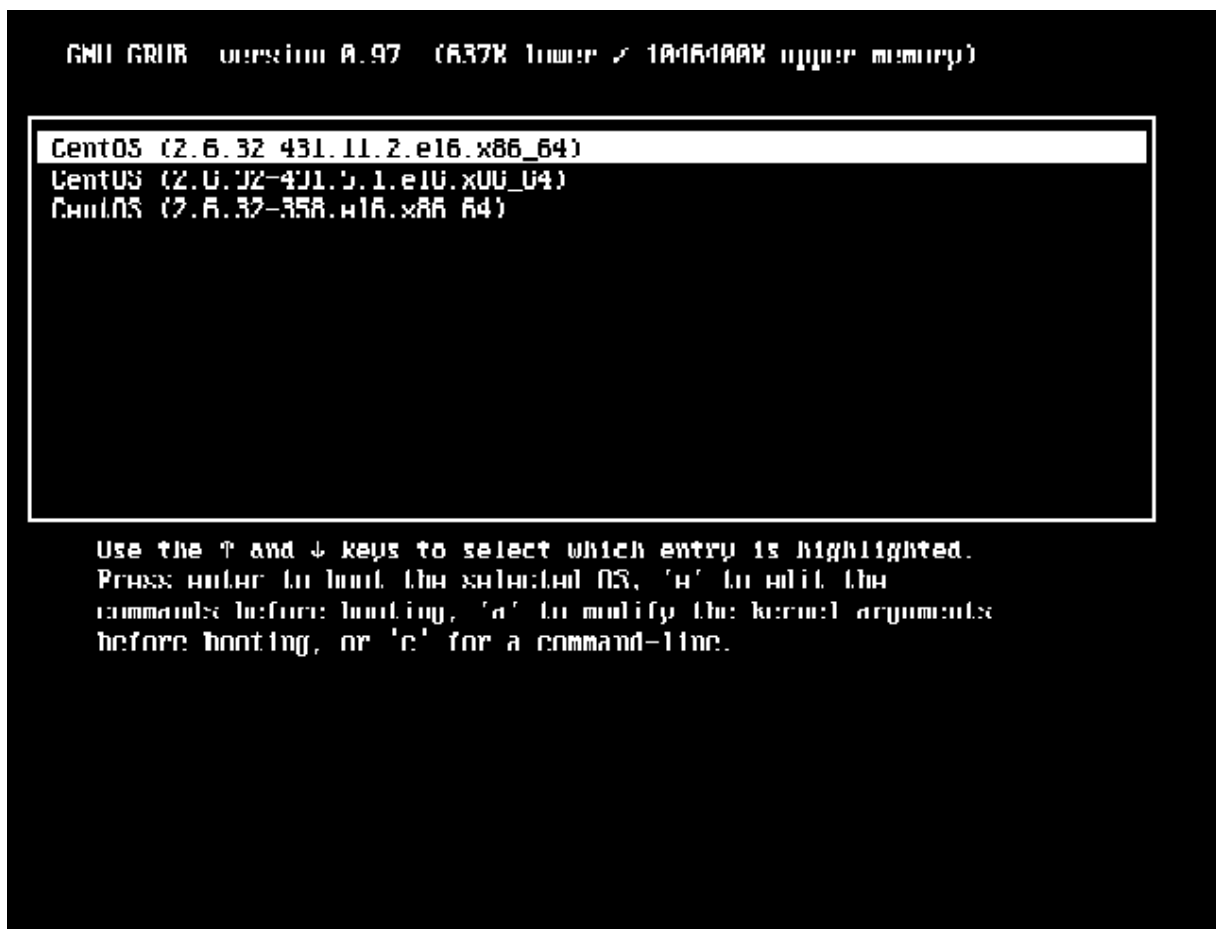
با گسترش شبکه اینترنت ، امروزه حجم تلاشها برای نفوذ به سرورهای اینترنتی به شدت افزایش یافته است مدیران سرورها به اجبار برای حفظ امنیت سرور خود اغلب از کلمات پیچیده و بلند برای رمز عبور سرور خود استفاده می کنند. در لینوکس کاربر root بالاترین سطح دسترسی دارد و فراموش کردن رمز عبور این کاربر درست در جایی که برای انجام وظایف مدیریتی سیستم شديداً به آن محتاجیم مشکل ساز می شود. در لینوکس این قابلیت وجود دارد که به راحتی نسبت به بازیابی رمز عبور root اقدام کنید و یک کلمه عبور جدید جایگزین نمایید. در این زمان است که می توان با استفاده از مد نجات براحتی پسورد را بازیابی کنید.

برای بازیابی پسورد در انواع مختلف لینوکس، روش ها متفاوت از یکدیگر می باشند ، این در حالیست که اصول یکی می باشد ولی ممکن است شیوه اجرا متمایز باشد. برای این کار باید از حالت single-user یا همان runlevel1 وارد لینوکس شوید و رمز عبور جدید را بسازید. یعنی ما باید سیگنالی به init بفرستیم تا سیستم عامل وارد محیط recovery شود.

**نکته:** دو علت برای تغییر RUNLEVEL وجود دارد : System Crash و Recovery Password

برای رفتن به حالت single-user باید سیستم ری بوت کنید. اگر بصورت پیش فرض از بوت لودر GRUB استفاده می کنید زمانی که منوی بوت به شما نشان داده می شود هسته ای که می خواهید از طریق آن سیستم خود را بوت کنید، برگزینید.





کلید **e** را بزنید تا وارد حالت ویرایش شوید. در این حالت یک محیط شبیه خط فرمان برای شما نمایش داده می شود تا دستورات را وارد کنید.

خطی که با عبارت `vmlinux` شروع می شود را انتخاب کرده کلید **e** را بزنید تا وارد یک صفحه جدید بشوید و خط را ویرایش کنید. در آخر همان خط یک فاصله ایجاد کرده و عدد 1 که به منزله ورود به `runlevel 1` است را تایپ می کنیم. سپس اینتر را زده تا تغییرات ذخیره شده ، به صفحه قبل هدایت شویم و جهت بوت سیستم دکمه **b** را فشرده تا سیستم به روند ادامه بوت برتایپ برگردد.



```
GNU GRUB version 0.97 (637K lower / 1046400K upper memory)

root (hd0,0)
kernel /tboot.gz logging=uva,serial,memory
module /umllinux-2.6.32-431.11.2.el6.x86_64 ro root=/dev/mapper/ug_ser→
module /initramfs-2.6.32-431.11.2.el6.x86_64.img

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time cancels. ENTER
at any time accepts your changes.]

<LE=us rd_NO_DM rhgb quiet 1
```

پس از انجام این کار به محیط تک کاربره خواهید رفت و از طریق `bash` می‌توانید رمز عبور خود را تغییر دهید. برای این کار کافی است از دستور `passwd` استفاده کنید. این کار سیستم را وارد حالت تک کاربره می‌کند. وقتی خط فرمان باز شد، دستور زیر را تایپ کنید:

```
passwd root
```

بعد از آن که رمز عبور خود را بازیابی کردید، می توانید دستورهای `init 3` یا `init 5` را بزنید `Init 3` به حالت چند کاربره بدون رابط گرافیکی بر می گردد و `init 5` حالت چند کاربره با ظاهر گرافیکی است بر گردید.

ادامه دارد.....



## منابع :

با تشکر از موسسه آموزشی نورانت و مرکز آموزشهای پیشرفته دانشگاه صنعتی شریف

تدریسات و راهنمایی های اساتید محترم :

مهندس مهدوی فر

مهندس جعفر عقربی

استاد حاجی غلامعلی

سایت [centos.org](http://centos.org)

سایت محترم [itpro.ir](http://itpro.ir)

سایت محترم [falearn.ir](http://falearn.ir)

سایت محترم [www.forum.persianadmins.ir](http://www.forum.persianadmins.ir)

سیلابس های درسی دوره های RHCSA و RHCE و LPic2

مطالب متفرقه منتشر شده در سایتها و وبلاگهای فارسی زبان لینوکس