

مدیریت بسته‌های RPM

بسته‌های نرم‌افزاری چه هستند و چرا به آنها نیاز داریم؟

هنگامی که توزیع لینوکس خود را نصب می‌کنید، اکثر نرم‌افزارهای مورد نیازتان همراه آن نصب می‌شود. ولی با این حال پیش می‌آید که نیاز به بسته‌های دیگری داشته باشید و یا بسته‌های نرم‌افزارهای موجود را بخواهید به نسخه‌های جدیدتر ارتقا دهید. در چنین مواردی است که باید بسته‌های جدید را روی CD و یا از اینترنت دریافت نموده و نصب نمایید.

بسته‌های نرم‌افزاری کار نصب نرم‌افزارهای کاربردی را در لینوکس آسان کرده‌اند. دیگر لازم نیست برای نصب هر نرم‌افزار، کاربر کدهای منبع را دریافت کرده، آنها را کامپایل نموده و نصب کند. کاری که بسیاری از کاربران آن انجام آن ناتوان هستند.

همانطور که متوجه شدید، بسته‌های نرم‌افزاری حاوی برنامه‌های قابل اجرا و آماده برای نصب هستند. ممکن است در اینترنت به انواع مختلفی از بسته‌های نرم‌افزاری برخورد نمایید که البته همه آنها حاوی نرم‌افزارهای آماده برای نصب نیستند که جلوتر درباره آنها توضیح خواهیم داد. پرکاربردترین بسته‌های آماده نصب نرم‌افزارهای کاربردی، بسته‌های نوع rpm و deb هستند. در این مقاله ما به بررسی بسته‌های نرم‌افزاری با فرمت rpm و مدیریت آنها خواهیم پرداخت و در مقاله جداگانه‌ای، مدیریت بسته‌های deb را که مخصوص لینوکس debian و توزیع‌های مبتنی بر آن است را بررسی خواهیم کرد.

درک نامگذاری و فرمت بسته‌های نرم‌افزاری

وقتی بسته‌ای را از اینترنت دانلود می‌کنید، می‌بینید که از یک نام، تعدادی شماره، نقطه، gz و tar تشکیل شده است. خوب اینها چه معنی دارند؟ ابتدا این اعداد و حرفها پیچیده به نظر می‌رسند در صورتی که اگر برایتان شرح داده شوند، بسیار ساده هستند. پس از مدتی به این شیوه نامگذاری عادت خواهید کرد. بیشتر بسته‌های نرم‌افزاری لینوکس از قاعده نامگذاری GNU استفاده می‌کنند. مثالهای زیر فرمت‌های گوناگون بسته‌های نرم‌افزاری را نشان می‌دهد:

```
mycoolapp-4.2.3.i386.rpm  
mycoolapp-4.2.3.tar.gz  
mycoolapp-4.2.3.src.tar.gz  
mycoolapp-4.2.3.bin.SPARC.tar.gz  
mycoolapp-4.2.3.bin.ELF.static.tar.gz
```

این مثالها بسته‌های مختلف یک نرم‌افزار هستند. نام این بسته نرم‌افزاری mycoolapp است. پس از نام بسته، شماره نسخه آن که ۴,۲,۳ است قرار دارد. اولین شماره شماره اصلی یا major، دومین شماره شماره فرعی یا minor و آخرین شماره شماره وصله یا patch است. پس از شماره نسخه، نقطه‌ای قرار دارد که بعد از آن بخش‌های انتخابی قرار دارند که نشانگر محتوای بسته و چگونگی فشرده سازی آن هستند.

خط نخست یک بسته نرم‌افزاری rpm را نشان می‌دهد. rpm مخفف Manager Redhat Package است. i386 که قبل از rpm قرار دارد، نشان دهنده این است که بسته حاوی نرم‌افزار اجرایی برای کامپیوترهای PC معمولی است. فرمت rpm که ابتدا توسط Redhat ارائه شد، اکنون تبدیل به یک استاندارد برای پخش بسته‌های نرم‌افزاری لینوکس شده است. بیشتر توزیع‌های لینوکس مانند ردهت، زوزه، ماندریک و... از این سیستم بسته بندی نرم‌افزار استفاده می‌کنند.

rpm در مقابل کد منبع

بسته‌هایی که بصورت فرمت rpm ارائه می‌شوند به سادگی قابل نصب بوده و برداشتن آنها نیز به سادگی امکان پذیر است. برای مدیریت بسته‌های rpm هم ابزارهای گرافیکی و هم ابزارهای متنی وجود دارند. در صورتی که در دنیای لینوکس تازه کار هستید، همیشه سعی کنید از این فرمت برای دریافت نرم‌افزارها استفاده نمایید.

RPM - نصب نرم افزارها با استفاده از بسته های rpm بسیار آسان است. برای نصب یک بسته rpm شما نیازی به دانستن اطلاعاتی در مورد makefile ها و کامپایلرها ندارید. همانطور که در بالا نیز اشاره کردم، ابزارهایی نیز برای مدیریت این بسته ها فراهم آمده است. هنگامی که یک بسته rpm را نصب می کنید، ابزارهای rpm سایر نرم افزارهای احتمالی مورد نیاز این بسته را نیز بررسی کرده و در صورت نیاز به نصب بسته دیگری به شما اعلام میکنند.

- کد منبع (Code Source) : برخی اوقات تا ایجاد بسته های rpm یک نرم افزار، مدتی طول می کشد. در این مواقع باید از کد منبع آن استفاده کنید. همچنین می توانید با ایجاد تغییراتی در کد منبع نرم افزار، آنرا برای خود مناسبتر کنید. اکثر کاربران حرفه ای لینوکس کار کردن با کد منبع را ترجیح می دهند.

خوب به مثال بالا برمی گردیم. در دوخط بعدی بسته هایی قرار دارند که حاوی کد منبع نرم افزار هستند. همانطور که می بینید، این بسته ها توسط دستور tar آرشیو شده (tar) و توسط دستور gzip فشرده شده اند (gz). شما از این دو فرمان برای باز کردن و خارج کردن از حالت فشرده گی بسته های نرم افزار استفاده خواهید کرد.

بین شماره های نسخه و پسوند های tar.gz بخش های اختیاری می توانند قرار گیرند. این بخش ها توسط نقطه از هم جدا می شوند و اطلاعات خاصی مانند نوع پردازنده ای که کدهای اجرایی بر روی آنها اجرا خواهد شد، نوع کدهای اجرایی و ... را فراهم می کنند. برای مثال bin.SPARC نشان می دهد که بسته حاوی کدهای اجرایی است که بر روی ایستگاههای کاری مبتنی بر پردازنده های SPARC اجرا خواهد شد.

استفاده از انواع گوناگون آرشیوها و فرمت مستندات

بسیاری از بسته های نرم افزار که به یک توزیع خاص (مانند ردهت، SuSE، Debian و...) وابسته نیستند از روش tar/gzip برای فشرده سازی و آرشیو کردن فایل ها استفاده می کنند. البته ممکن است در سایتهای FTP لینوکس به فرمت های دیگری نیز برخورد کنید. لازم به ذکر است که برخی از این فایلها مستندات همراه نرم افزار بوده و کد منبع یا اجرایی نیستند.

جدول ۱ فرمت های مختلف فایلی که شما هنگام چرخش در سایتهای FTP لینوکس مشاهده خواهید کرد و جدول ۲ برخی فرمت های رایج مستندات که همراه با نرم افزارهای لینوکس ارائه می شوند را نشان میدهد.

برای اینکه بدانید یک فایل دارای چه فرمتی است، می توانید از دستور file استفاده کنید. به مثال زیر توجه کنید :

\$ file telnet.htm

telnet.htm: HTML document text

فرمت	پسوند	شرح
Gzip file	gz or .z	این فایل با استفاده از ابزار GNU gzip فشرده شده است. این فایل توسط ابزارهای gzip یا gunzip باز خواهد شد.
Tar File	tar	این فایل با استفاده از دستور tar آرشیو شده است. این فرمان برای ایجاد یک بسته از چند فایل بکار می رود. برای باز کردن این آرشیو از خود دستور tar استفاده می شود.
Bzip2	bz2	این فایل با استفاده از برنامه bzip2 فشرده شده است. میزان فشرده سازی این برنامه از gzip بیشتر است.
Tar/compressed	taz or .tz	این فایل توسط دستور compress یونیکس فشرده شده است.
Linux Software	ism	این فایل حاوی متنی است که محتویات یک

آرشیو را نشان می‌دهد.		Map
این فایل یک بسته نرم‌افزاری است که توسط ابزارهای لینوکس debian ایجاد شده است.	deb	Debian binary package
این فایل بسته نرم‌افزاری است که با استفاده از ابزارهای rpm ایجاد شده است و غیر از ردهت، توسط توزیع کنندگان دیگر لینوکس هم استفاده می‌شود.	rpm	Redhat Package Manager

جدول ۱ انواع فرمت بسته‌های نرم‌افزاری لینوکس

نصب برنامه های کاربردی در لینوکس ردهت

بیشتر برنامه هایی که نیاز به نصب آنها دارید، به صورت فرمت rpm موجود هستند. در صورتی که هیچ بسته rpm ای برای نرم‌افزار مورد نظرتان پیدا نکردید، می‌توانید از کد منبع نرم‌افزار استفاده کنید و خودتان نرم‌افزار مورد نیاز را کامپایل و نصب کنید.

فرمت	پسوند	شرح
Hypertext Markup Language	html. or .htm	این فرمت شناخته شده را می‌توانید در هر مرورگر وبی بخوانید.
PostScript	ps.	یک فایل پست اسکریپت برای ارسال به یک چاپگر پست اسکریپت.
SGML	sgml.	فایل‌های SGML برای ایجاد مستنداتی بکار می‌روند که آنها را می‌توان به فرمت‌های گوناگون تبدیل کرد.
DVI	dvi.	این فرمت خروجی برنامه LaTeX است. با استفاده از فرمان dvi از آنرا به PostScript یا HP PCL تبدیل کنید.
Plain Text		فایل‌های بدون پسوند معمولاً متن‌های ساده با فرمت ASCII هستند.

جدول ۲ انواع فرمت مستندات همراه بسته های نرم‌افزاری

نصب و مدیریت بسته های rpm

هنگامی که یک برنامه کاربردی که با فرمت rpm است به دستتان می‌رسد، این بسته معمولاً فقط شامل یک فایل است. دستوری که شما برای مدیریت و استفاده از بسته های نرم‌افزاری rpm استفاده می‌کنید، دستور rpm است.

برای مدیریت بسته های rpm ، دستور rpm گزینه‌هایی را برای لیست کردن بسته‌های نصب شده، ارتقا بسته‌های نصب شده، بدست آوردن اطلاعات از محتویات یک بسته و ... ارائه می‌کند. همچنین گزینه‌هایی وجود دارند که توسط آنها می‌توانید بررسی نمایید که کلیه فایل‌های یک بسته نصب شده وجود داشته و تغییر نکرده باشند. دستور rpm در حالت‌های زیر کار می‌کند:

- نصب نرم‌افزار (install) با گزینه i

- ارتقا نرم‌افزار (Upgrade) با گزینه U

- کسب اطلاعات از بسته (Query) با گزینه q

- چک کردن بسته (Verify) با گزینه V
 - بررسی امضا بسته (check Signature) با گزینه checksig
 - برداشتن بسته (Uninstall) با گزینه e
 - ساخت بسته (Build) با گزینه b
 - ساخت مجدد بانک اطلاعاتی (Database Rebuild) با گزینه rebuildddb
 - تصحیح مجوزها (Permissions Fix) با گزینه setperms
 - تعیین مالک/گروه (Owners/groups Set) با گزینه setugids
- با استفاده از این گزینه‌ها می‌توانید بسته‌های rpm را نصب کرده، صحت نصب بسته‌ها را بررسی کنید و اشکالات احتمالی را برطرف نمایید.

هشدار: برای نصب یا برداشتن بسته‌های rpm باید با کاربر ریشه وارد سیستم شده باشید. نمایش بسته‌های نصب شده، کسب اطلاعات از بسته و نمایش محتویات بسته نیاز به داشتن مجوز ریشه ندارد.

نصب بسته های rpm

برای نصب بسته های rpm با استفاده از دستور rpm، باید از روش زیر استفاده کنید :

\$ rpm -i [option] package

package نام بسته rpm است. این بسته ممکن است در مسیر جاری شما قرار داشته باشد، یا در روی درایو CD-ROM و یا روی سایت FTP .

توجه: در صورتی که دستور نصب بسته‌ای را صادر کنید که روی سایت FTP قرار دارد، این بسته ابتدا از سایت دانلود شده و سپس نصب می‌گردد.

همراه با گزینه `-v` هنگام نصب می‌توانید از گزینه‌های زیر استفاده نمایید:

`-vv` - این گزینه اطلاعات Debugging را هنگام نصب چاپ میکند. با این گزینه تمام آنچه را هنگام نصب رخ میدهد خواهید دید.

`-h` - فرایند نصب بسته را بصورت تعدادی کاراکتر # نشان میدهد. این کاراکترها در کنار هم تشکیل یک نوار پیشروی (Progress bar) را میدهند که برای کسب اطلاعات از وضعیت نصب بسته مفید است.

`-percent` - درصد پیشرفت نصب بسته را حین فرایند نصب چاپ میکند.

قبل از اینکه نصب بسته شروع شود، ابزار rpm بررسی‌هایی مبنی بر نصب بودن نسخه جدیدتر بسته و یا بسته‌های دیگری که به آن نیاز می‌باشد (Dependencies) به عمل می‌آورد. در صورتی که بسته‌های جدیدتر وجود داشته باشد، بسته نصب نخواهد شد. برای عبور از این شرایط می‌توانید از گزینه‌های زیر استفاده کنید:

`--force` - با استفاده حتی اگر بسته در حال نصب قدیمی‌تر از بسته نصب شده باشد، نصب خواهد شد. این کار فایل‌های جدیدتر را با فایل‌های قدیمی‌تر جایگزین خواهد کرد. این گزینه مشابه استفاده از گزینه‌های `oldpackage`، `rplacefiles` و `replacepks` است.

-nodeps -- بسته های مورد نیاز برا نادیده گرفته و بسته را نصب میکند. البته ممکن است بسته ای که اینگونه نصب شود، کار نکند.

--ignorearch - با استفاده از این گزینه اگر بسته با معماری پردازنده شما سازگاری نداشته باشد نیز نصب خواهد شد.

--ignoreos - در صورتی که بسته با سیستم عامل کامپیوترتان سازگاری نداشته باشد، نصب خواهد شد.

مثال زیر چگونگی نصب یک بسته را نشان می دهد :

\$ rpm -i audiofile-devel-0.6-1.i386.rpm

من شخصا ترجیح میدهم هنگام نصب بسته مطلع باشم که چه چیزی رخ میدهد. برای این کار معمولا گزینه vv را هم اضافه می کنم:

\$ rpm -ivv audiofile-devel-0.6-1.i386.rpm

```
D: counting packages to install
D: found 1 packages
D: looking for packages to download
D: retrieved 0 packages
D: New Header signature
D: Signature size: 160
...
```

با استفاده از یک خروجی اینچنینی می توانید به تمام جزئیات عملیاتی که هنگام نصب بسته انجام می شود ، پی ببرید. راه دیگری که می توانید بررسی کنید که آیا بسته به درستی نصب شده است یا خیر استفاده از گزینه h است. به مثال زیر توجه کنید :

\$ rpm -ivh audiofile-devel-0.6-1.i386.rpm

```
audiofile-devel [#####]100%
```

بوسیله این گزینه، همانطور که می بینید تعدادی کاراکتر # روی صفحه ترمینال چاپ می شود که بیانگر صحت نصب بسته است. همانطور که دیدید در صورتی که اشکالی وجود نداشته باشد، نصب بسته های rpm بسیار آسان است. برخی اوقات ممکن است در نصب بسته ها اشکالاتی ایجاد شود که نمونه هایی از آنها در زیر شرح داده شده اند.

-خطای بسته های مورد نیاز (dependencies errors Package) : در صورتی که بسته هایی که نصب می کنید، برای کارکرد صحیح به بسته های دیگری نیاز داشته باشند، این خطا را دریافت خواهید کرد. برای حل این مشکل کافی است که بسته های مورد نیاز را نیز تهیه کرده و قبل از نصب بسته اصلی آنها را نصب کنید. البته همانطور که در بالا شرح داده شد، می توانید rpm را مجبور کنید تا بسته های مورد نیاز را بررسی نکند، ولی این کار به هیچ وجه توصیه نمی شود زیرا بسته نصب شده کار نخواهد کرد.

- خطاهای کاربر غیر ریشه : در صورتی که در حالت کاربر ریشه نباشید و دستور rpm را با گزینه ا اجرا کنید، این خطاها را دریافت خواهید کرد. برای رفع این مشکل کافی است با استفاده از دستور su بصورت کاربر ریشه وارد شوید.

ارتقا بسته های rpm

گزینه ارتقا یا U بسیار شبیه به گزینه نصب بسته جدید می باشد. با این تفاوت که ابتدا بسته قدیمی را از روی سیستم برداشته و سپس بسته جدید را نصب میکند. به روش زیر توجه کنید :

\$ rpm -U [options] package

ممکن است پس از نصب بسته جدید، به دلایلی نرم‌افزار مربوطه کار نکند. برای بازگشت به حالت قبلی و نصب بسته قدیمی تر باید از گزینه `oldpackage` استفاده نمایید. برای روشن شدن مطلب به مثال زیر توجه کنید :

\$ rpm -U --oldpackage audiofile-devel-0.5-3.i386.rpm

در صورتی که بسته جدیدتر نصب باشد، از روی سیستم برداشته شده و بسته قدیمی تر نصب خواهد شد.

نکته : از گزینه `U` می‌توانید چه برای نصب جدید بجای گزینه `i` و چه برای ارتقا استفاده نمایید. گزینه `U` همیشه کار خواهد کرد. ولی گزینه `A` در صورتی که بسته قبلاً نصب شده باشد، کار نخواهد کرد.

حذف بسته های rpm

در صورتی که دیگر به یک بسته نیازی ندارید و میخواهید برای آزاد شدن فضای دیسک سخت خود آنرا از روی سیستم برداشته و حذف نمایید، باید از گزینه `e` استفاده نمایید. روش این کار به صورت زیر است :

\$ rpm -e package

در صورتی که هیچ بسته ای به بسته مورد نظر نیازی نداشته باشد، بسته از روی سیستم حذف خواهد شد. ممکن است قبل از حذف بسته بخواهید این امر را بررسی نمایید. برای این کار باید از گزینه `q` استفاده کنید. با استفاده از این گزینه می‌توانید بررسی های مختلفی انجام داده و اطلاعات با ارزشی بدست آورید. برای بررسی اینکه بسته مورد نظر، مورد نیاز بسته ای می‌باشد یا خیر، از روش زیر استفاده کنید :

\$ rpm -q --whatrequires package

در صورتی که تصمیم گرفتید تا بسته‌ای را حذف نمایید، من توصیه می‌کنم تا از گزینه `vv` به همراه گزینه `e` استفاده کنید. این کار به شما نشان می‌دهد که آیا واقعا فایل‌های بسته حذف شده اند یا خیر..

بدلیل اینکه معمولا در این موارد خروجی طولانی خواهد بود و از صفحه ترمینال فراتر خواهد رفت، می‌توانید خروجی را از طریق لوله بندی به یک برنامه دیگر و یا یک فایل هدایت نمایید. به مثال زیر توجه کنید:

\$ rpm -evv xpilot | more

در مثال بالا خروجی rpm به برنامه `more` هدایت می‌شود و از طریق آن می‌توانید خروجی را بصورت صفحه به صفحه مطالعه نمایید. برای این کار می‌توانید از کلید `Space` استفاده کنید.

سایر گزینه هایی که می‌توانید همراه با گزینه `e` برای نادیده گرفتن شرایطی که مانع حذف شدن یک بسته می‌شوند، استفاده کنید به شرح زیر هستند :

`--nodeps` بسته را بدون توجه به سایر بسته هایی که به آن نیاز دارند حذف میکند.

`--nodcripts` بسته را بدون اجرای هر گونه اسکریپتی حذف میکند.

برای اینکه مطمئن شوید هنگام برداشتن بسته چه اتفاقاتی رخ میدهد، می‌توانید با استفاده از گزینه `test` فقط برداشتن بسته را شبیه سازی نمایید. با این کار مشکلات احتمالی را مشاهده خواهید کرد و بهتر می‌توانید آنها را رفع کنید. همراه با گزینه `test` هم می‌توانید از `vv` استفاده کنید . به مثال زیر توجه کنید :

\$ rpm -evv --test xpilot | more

در صورتی که همه چیز به خوبی پیش رفت، می‌توانید دستور خود را بدون گزینه test اجرا نمایید تا بسته از روی سیستم برداشته شود.

بدست آوردن اطلاعات از بسته های rpm

با استفاده از گزینه q می‌توانید اطلاعات با ارزشی را از بسته های rpm دریافت نمایید. این اطلاعات می‌تواند از لیست بسته های نصب شده تا اطلاعات جزئی هر بسته متغیر باشد. روش استفاده از این گزینه به صورت زیر است :

\$ rpm -q [options]

اطلاعات مفیدی که می‌توانید با استفاده از این گزینه بدست آورید به شرح زیر هستند :

-qa لیستی از بسته‌های نصب شده را نمایش می‌دهد.

-file qf بسته‌ای که یک فایل خاص به آن تعلق دارد را چاپ می‌کند. بجای file باید مسیر کامل فایل مورد نظر را تایپ کنید و در صورتی که فایل در دایرکتوری جاری قرار دارد، تنها کافی است که نام فایل را تایپ کنید.

-qi اطلاعات جزئی یک بسته را چاپ می‌کند.

-qR اجزای مورد نیاز بسته را چاپ می‌کند.

-ql لیست فایل‌هایی که در بسته قرار دارند را چاپ می‌کند.

-qd مستنداتی که در بسته وجود دارند را چاپ می‌کند.

-qc فایل‌های پیکربندی که در بسته قرار دارند را چاپ می‌کند.

همانطوری که در بالا گفته شد برای لیست کردن تمام بسته های موجود روی کامپیوترتان می‌توانید از گزینه qa استفاده کنید. بدلیل اینکه این لیست بسیار طولانی خواهد بود، می‌توانید خروجی آنرا به یک فایل هدایت نموده و یا با استفاده از دستور grep بسته های مورد نیازتان را پیدا کنید. برای روشن شدن مطلب به مثالهای زیر توجه کنید :

```
myPackages < rpm -qa $
```

```
php rpm -qa | grep -i $
```

```
phpdoc-4.0.6-133
```

```
mod_php4-4.1.0-104
```

```
mod_php4-core-4.1.0-104
```

در مثال اول خروجی دستور به فایلی به نام myPackages هدایت می‌شود. این فایل یک فایل متنی است که می‌توانید آنرا در یک ویرایشگر متنی مانند gedit یا vi مشاهده کنید. در مثال دوم، خروجی دستور به فیلتر grep هدایت شده و خواسته می‌شود تا نامهای حاوی php نمایش داده شوند. نتیجه وجود سه بسته نصب شده است.

در صورتی که به اطلاعات جزئی در مورد یک بسته خاص نیاز داشتید، می‌توانید از گزینه qi استفاده کنید. به مثال زیر توجه کنید :

```
$ rpm -qi phpdoc
```

```
Name : phpdoc Relocations: (not relocateable)
```

```
Version : 4.3.2 Vendor: Red Hat, Inc.
```

```
Release : 133
```

```
Build Date: Sat 23 Mar 2002 11:30:27 AM PST
```

```
Install date : Thu 12 Jun 2003 11:55:13 AM PDT
```

```
Build Host: devel.redhat.com
```

Size : 8592660 License: the PHP Group
Description : Documentation to PHP as HTML, PDF and RTF.

برای کسب اطلاعات از محتوای یک بسته، باید از گزینه qI استفاده کنید. مثال زیر محتویات بسته phpdoc را نمایش میدهد :

\$ rpm -qI phpdoc

```
/usr/share/doc/packages/phpdoc/manual/x61684.html  
/usr/share/doc/packages/phpdoc/manual/x61688.html  
/usr/share/doc/packages/phpdoc/manual/x61694.html  
/usr/share/doc/packages/phpdoc/manual/x61720.html  
...
```

آیا دوست دارید بدانید که همراه یک بسته خاص چه مستنداتی ارائه شده است؟ کافی است از گزینه qd استفاده کنید. با این کار می‌توانید به محل مستندات یک برنامه به راحتی پی ببرید. به مثال زیر توجه کنید :

\$ rpm -qd ppp

```
/usr/share/doc/packages/ppp/FAQ  
/usr/share/doc/packages/ppp/README  
/usr/share/doc/packages/ppp/README.cbcp  
/usr/share/doc/packages/ppp/README.linux  
...
```

بیشتر بسته های نرم‌افزاری به همراه خود چندین فایل پیکربندی دارند . برای پیدا کردن آنها باید از گزینه qc استفاده کنید. به مثال زیر توجه کنید :

\$ rpm -qc ppp

```
/etc/pam.d/ppp  
/etc/ppp/chap-secrets  
/etc/ppp/options  
/etc/ppp/pap-secrets  
...
```

در صورتی که نیاز دارید بدانید که یک فایل یا برنامه خاص به کدام بسته تعلق دارد، می‌توانید از گزینه qf استفاده نمایید. به مثال زیر توجه کنید:

usr/bin/wvdial/ rpm -qf \$

```
wvdial-1.52-136
```

بررسی صحت نصب بسته های rpm

گاهی اوقات مدتی پس از نصب یک بسته به نظر می‌رسد که نرم‌افزار مربوطه بطور صحیح کار نمی‌کند. برای بررسی اینکه تمام محتویات یک بسته بدون تغییر باقی مانده اند یا خیر، باید از گزینه V استفاده نمایید. اطلاعات هر بسته نصب شده در بانک اطلاعات rpm کامپیوترتان ذخیره می‌شود. بوسیله گزینه V می‌توانید بررسی کنید که فایل‌های مربوط به بسته دچار تغییر نشده باشند.

هنگامی که از این گزینه استفاده می‌کنید تعداد زیادی آزمایش سازب فایلها و مجوزهای آنها صورت می‌گیرد. در صورتی که تمام فایلها بدون مشکل باشند، شما هیچ خروجی از دستور مشاهده نخواهید کرد. تمام بخشهایی که پس از نصب بسته دچار تغییر شده باشند، همراه با چگونگی تغییر آنها چاپ خواهند شد. به مثال زیر توجه کنید:

\$ rpm -V ppp

```
S.5.....T c /etc/ppp/chap-secrets  
S.5.....T c /etc/ppp/pap-secrets
```

خروجی مثال بالا نشان می‌دهد که از بسته ppp (نرم‌افزاری که برای ایجاد تماس dialup با اینترنت به کار می‌رود) پس از نصب ۲ تغییر در فایل‌هایش داشته است. حرف S نشان‌دهنده تغییر در اندازه فایل، عدد ۵ نشان دهنده تغییر در ترکیب MD5 فایل، حرف T نشان‌دهنده تغییر در زمان تغییر فایل و حرف c نشان می‌دهد که این فایل‌ها فایل‌های پیکربندی هستند. با بررسی فایل‌ها متوجه می‌شویم که تغییرات اعلام شده، تغییراتی است که خود من برای تنظیم برنامه ppp انجام داده‌ام. پس بسته ppp مشکلی ندارد. علائم نشان‌دهنده ای که با این گزینه با آنها برخورد خواهید کرد عبارتند از :

o : نشانگر تغییر در MD5 checksum است. نشان‌دهنده ایجاد تغییراتی در محتوای یک فایل می‌باشد.

S : نشانگر تغییر در اندازه فایل می‌باشد.

L : نشانگر این است که فایل تبدیل به یک لینک ظاهری (Symbolic) به فایل دیگری شده است.

T : زمان تغییر فایل را نشان می‌دهد.

D : فایل مورد نظر تبدیل به یک فایل مخصوص ابزار (File Device Special) شده است.

U : کاربر مالک فایل تغییر کرده است.

G : گروه کاربری فایل تغییر کرده است.

M : مالکیت یا مجوزهای دسترسی به فایل تغییر کرده است.