به نام خدا

ساخت kernel بهینه از Kernel Source RPM

ساخت kvm بهینه از Source RPM

محدوده:

در این مستند شیوه ساخت RPM از روی RPM Source و به صورت ویژه شیوه ساخت یک هسته بهینه از روی Kernel RPM Source بررسی میشود.

تاریخچه:

ردیف	نویسنده	تاریخ	شماره ویرایش	توضیحات
1	تحقیق و توسعه فنی و مهندسی	۱۳۸۹/۱/۱۵	۱.۰	ساخت RPM از Source RPM ساخت Kernel از Source RPM ساخت KVM
۲	تحقیق و توسعه فنی و مهندسی	۱۳۹۱/۱۲/۲۲	۱.۱.۰	نحوه ساخت Kernel در ویرایشهای Fedora 12 به بعد

ردیف	نویسنده	تاریخ	شماره ویرایش	توضیحات
۳	تحقیق و توسعه فنی و مهندسی	۱۳۹۲/۱۲/۰۲	۱,۲,۰	تغییر در ترتیب مراحل ساخت کرنل
٤	تحقیق و توسعه فنی و مهندسی	۱۳۹۲/۱۲/۰۸	۱,۳,۰	ذکر همانند بودن کامپایل کردنلهای Redhat6 همانند fedora
٥	تحقیق و توسعه فنی و مهندسی	۱۳۹۲/۰۹/۱۲	۱,٤,۰	نحوه ساختن patch برای Radhat6 و fedora12 به بعد

فهرست مندرجات

۱ مقدمه

با توجه به در اختیار بودن سورس سیستم عامل ردهت در قالب Source RPM و قابلیت اطمینان بالای این بسته‌های نرم‌افزاری، و همچنین وجود نرم‌افزارهای ویژه در قالب Source RPM، استفاده از این سورسها و بهینه‌سازی آنها جهت دستیابی به بسته‌های نرم‌افزاری قابل اطمینان بسیار با اهمیت است.

تمامی موارد مطرح شده بر روی CentOS انجام و تست شده است و در مورد هسته مشخصا هسته CentOS 5 مورد نظر میباشد.

۲ شیوه ساخت RPM از روی Source RPM مورد نظر

برای ساخت RPM بهتر است که از کاربر root استفاده نشود. بنابراین نام کاربری ویژه ای را برای این منظور بایددر نظر گرفت (مثال pdnsoft). بعد از ساخت نام کاربری مورد نظر مراحل زیر را باید پیمود:

۱- نصب بسته‌های نرم‌افزاری مورد نیاز:

```
[root@host]# yum install rpm-build redhat-rpm-config unifdef
```

توجه: در تمام مراحل بعدی، فعالیتها در نام کاربری تعیین شده انجام خواهد شد:

۲- ساخت محیط کارگاهی(شاخه‌های مورد نیاز برای ساخت RPM) در home نام کاربری انتخاب شده؛

```
[pdnsoft@host]$ cd
[pdnsoft@host]$ mkdir -p rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
[pdnsoft@host]$ echo '%_topdir %(echo $HOME)/rpmbuild' > .rpmmacros
```

۳- دانلود Source RPM مورد نظر:

جهت دسترسی به Source RPMهای ردهت میتوان از مسیرهای زیر استفاده کرد:

http://mirror.centos.org/centos/5/updates/SRPMS/
http://mirror.centos.org/centos/5/os/SRPMS/
http://mirrors.kernel.org/redhat/redhat/linux/enterprise/5Server/en/os/SRPMS/
http://mirrors.kernel.org/redhat/redhat/linux/enterprise/5Client/en/os/SRPMS/

۴- نصب Source RPM دانلود شده:

```
[pdnsoft@host]$ rpm -i test-pac.src.rpm
```

۵- دستیابی به سورس RPM دانلود شده:

باید توجه داشت که یک Source RPM مشتمل بر یک سورس اولیه و میزان قابل توجهی patch میباشد که در هنگام ساخت RPM بر سورس اولیه اعمال میشوند و سپس از سورس patch شده RPM ساخته میشود.

بنابراین بعد از نصب Source RPM ما هنوز به سورس کامل دسترس نداریم و باید دستورات لازم برای دسترسی به Source را اجرا کنیم که به این شرح است:

```
[pdnsoft@host]$ cd ~/rpmbuild/SPECS
```

```
[pdnsoft@host]$ rpmbuild -bp test-pac.spec
```

با اجـرا این‌ـدسـتور ســورس برنامـه در شrpmbuild/BUILD قرار خواهد گرفت.

فایل spec. شامل تنظیمات ساخت RPM میباشد.

۶- اعمال تغییرات مورد نیاز در Source:

در صورتیکه نیاز باشد تا در سورس تغییراتی داده شود، تغییرات در سورس بدست آمده از مرحله قبل باید به صورت یک patch آماده شوند. لازم است که این patch در شاخه rpmbuild/SOURCES/ قرار گیرد و جهت اعمال Patch در هنگام ساخت RPM نیاز است تا در فایل spec. مواردی به این شرح اضافه گردد:

* معرفی patch مورد نظر: اگر به فایل spec نگاهی بیاندازی متوجه خواهید شد که در قسمتی از آن patch ها به ترتیب با ذکر شماره معرفی شده‌اند:

```
Patch40000: our-patch.patch
```

در انتخاب شماره انتهای کلمه Patch باید دقت شود تا با patchهای موجود تلاقی پیدا نکند. بهتر است که از روی آخرین patch بزرگترین عدد را انتخاب کنیم یا مثلا از شماره‌ای مانند 40000 استفاده شود که خیلی بزرگ است.

* اعمال patch: همچنین درفایل spec در مکانی نحوه اعمال patch ها مشخص شده است که به اینصورت در انتهای آن patch مورد نظر ما اضافه میشود:

```
%patch40000 -p1
```

۷- ساخت RPM

```
[pdnsoft@host]$ rpmbuild -bb --target `uname -m` test-pac.spec
```

بعد از اجرای این دستور RPM مورد نظر در شاخه rpmbuild/RPMS قرار خواهد گرفت.

۳ شیوه ساخت Kernel RPM از روی Source RPM مربوط به هسته

تمامی مراحل عنوان شده در ساخت RPM از روی Source RPMدر اینجا صادق است. مگر مواردی که مربوط به کانفیگ کرنل است. مراحل کانفیگ کرنل بعد از دستیابی به سورس آن (انجام مراحل ۱ الی ۵ از قسمت قبلی) موارد زیر را باید انجام داد:

۱- نصب بسته‌های مورد نظر جهت ساخت هسته:

```
    * yum groupinstall "Development Tools" # This will ensure that you have all
the required tools for the build.

    * yum install hmaccalc # This is required to supply the tools which can
    calculate HMAC values for files.

    * yum install ncurses-devel # This is required to enable a make *config
    command to execute correctly.
```

۲- انجام تغییرات در فایل kernel-2.6.spec

یکی از تغییرات مهم در فایل spec مربوط به هسته تعیین buildid میباشد. عبارتی که در اینجا ذکر شود به انتهای نام

هسته اضافه خواهد شد و به عبارتی شاخه کرنل تغییر داده شده ما خواهد بود. برای این منظور لازم است که خط:

```
%define buildid .your_identifier
```

را جستجو کرده و به جای عبارت your_identifier عبارت مورد نظر قرار گیرد (در ساخت RPM که توسط ما انجام شد از واژه pkv1 که مخفف PDNSoftCo. Kernel Version 1 هست استفاده شده است). باید توجه داشت که بین % و define نباید هیچگونه فاصله‌ای وجود داشته باشد.

در نسخه‌های جدید این خط کامنت شده است، به دنبال عبارت buildid باشید.

همچنین در ساخت هسته CentOS 5 عنوان شده است که موارد زیر از حالت Comment خارج شود که در CentOS 5.4 به صورت پیش‌فرض از کامنت خارج شده بودند: (این موضوع برای نسخه‌های لینوکس ۶ و بعد از آن صادق نیست)

```
#if a rhel kernel, apply the rhel config options
#%if 0%{?rhel}
#  for i in %{all_arch_configs}
#  do
#    mv $i $i.tmp
#    $RPM_SOURCE_DIR/merge.pl $RPM_SOURCE_DIR/config-rhel-generic $i.tmp > $i
#    rm $i.tmp
#  done
#%ifarch x86_64 noarch
#  for i in kernel-%{kversion}-x86_64*.config
#  do
#    mv $i $i.tmp
#    $RPM_SOURCE_DIR/merge.pl $RPM_SOURCE_DIR/config-rhel-x86_64-generic $i.tmp >
$i
#    rm $i.tmp
#  done
#%endif
#%ifarch ppc64 noarch
#  #CONFIG_FB_MATROX is disabled for rhel generic but needed for ppc64 rhel
#  for i in kernel-%{kversion}-ppc64.config
#  do
#    mv $i $i.tmp
#    $RPM_SOURCE_DIR/merge.pl $RPM_SOURCE_DIR/config-rhel-ppc64-generic $i.tmp >
$i
#    rm $i.tmp
#  done
#%endif
#%endif
```

۳- کانفیگ کرنل: برای کانفیگ کرنل میبایست از کانفیگهای پیشفرضی که وجود دارند(در شاخه configs در سورس کرنل) استفاده و آنها را به طریقه دلخواه تغییر داد:

```
[pdnsoft@host]$ cd ~/rpmbuild/BUILD/kernel-2.6.18/linux-2.6.18.`uname -m`

[pdnsoft@host]$ cp configs/kernel-2.6.18-`uname -m`[-type].config .config
```

بعد از این با اجرای دستور make menuconfig و یا دستورات دیگر جهت تغییر کانفیگ استفاده کرده و کانفیگ کرنل را به طرز دلخواه انجام میدهیم.

بعد از انجام کانفیگ یکی از دو عبارت زیر را در خط اول (ابتدای) کانفیگ قرار میدهیم:

```
# i386
# x86_64
```

چنانکه مشخص است این عبارتها به صورت کامنت شده در ابتدای فایل قرار میگیرند که اولی برای سیستمهای ۳۲ بیتی و دومی برای سیستمهای ۶۴ بیتی است.

بعد از انجام تغییرات فایل کانفیگ را میبایست در شاخه SOURCES کپی کرد که برای این منظور مراحل زیر را انجام میدهیم:

```
[pdnsoft@host]$ cp .config configs/kernel-2.6.18-`uname -m`[-type].config
[pdnsoft@host]$ cp configs/* ~/rpmbuild/SOURCES
```

بدینصورت در هنگام سخت RPM از کانفیگ مورد نظر ما استفاده خواهد شد.

۴- ساخت RPM: دستور استفاده شده توسط ما به این شرح میباشد:

```
rpmbuild -bb --without xen --without debug --without debuginfo
--target=`uname -m` kernel-2.6.spec
```

option هایی که میتوانند جهت ساخت کرنل استفاده شوند به این شرح است:

```
--with baseonly
--with xenonly
--without up
--without xen
--without debug
--without debuginfo
--without fips
--without kabichk
```

بعد از این مرحله RPMها در شاخه RPMS قرار خواهند گرفت.

۳.۱ ABI در هسته CentOS

یکی از خصوصیات هسته CentOS آن است که در طول تولید یک محصول ABI کرنل ثابت میماند و این به آن معنی است که ماژولهای خارجی مستقل از ویرایش کرنل کامپایل میشوند و در صورت تغییر کرنل با همان ABI نیاز به کامپایل مجدد ماژول نیست.

۴ شیوه ساخت kvm

برای KVM هم میتوان از نسخه پیشفرض CentOS استفاده کرد و یا اینکه آن را از Source RPM بسازیم. مراحل ساخت قبل ذکر شد تنها مورد یک تغییر در kvm.spec میباشد. با توجه به اینکه این kvm برای کرنل ویرایش شده ما باید کامپایل شود (pkv1) لازم است که خط زیر به صورت زیر تغییر یابد:

```
%{!?kversion: %define kversion 2.6.18-164.11.1.el5.pkv1}
```

۵ ساخت کرنل در Fedora 12 و نسخه‌های SL6 به بعد

ساخت کرنل از فدورا ۱۲ به بعد تغییراتی کرده است به این شرح: [۱]

فعالیتها شبیه به همان وضعیت توضیح داده شده در «شیوه ساخت Kernel RPM از روی Source RPM مربوط به هسته» با یک تفاوت اصلی.

1. cp configs/<desired-config-file> .config
2. make menuconfig
3. Add this to the first of .config: # x86_64
4. cp .config ~/rpmbuild/SOURCES/config-`uname -m`-generic

تفاوت اصلی مرحله چهار است که با شیوه قبل تفاوت چشمگیری دارد.

نحوه ساخت هم میتواند به این شکل باشد:

```
rpmbuild -bb --define="buildid .BID" --without xen --without debug --without debuginfo --target=`uname -m` kernel.spec
```

۵.۱ ساخت Patch

جهت نحوه سخت patch برای کرنلهای redhat6 به بعد ... (ابتدا یک کپی از فایل مورد نظر خود با پسوند .ORIG تهیه و تغییرات را در فایل اصلی اعمال کنید)

```
cd kermel.../
diff -up linux-x.x.x/patch/to/file{.ORIG,} > patch_file
```

نوبت به تغییر در kernel.spec است. ابتدا باید patch را معرفی کنید که شبیه به حالت قبل است.

سپس نوبت به اعمال patch است که میبایست خطی به این شرح به فایل اضافه شود:

```
ApplyPatch patch-pkv64_r1
```

اگر خطی برای ApplyPatch یا ApplyOptionalPatch قرار دارد این خط را در انتهای آنها قرار دهید. در مورد عملی برای pkv64_r1 این خط قبل از خط زیر قرار گرفت:

```
chmod +x scripts/checkpatch.pl
```

۶ منابع

[۱] http://fedoraproject.org/wiki/Building_a_custom_kernel