灵跃桌面云

QEMU 修改及安装包编译

成/都/虫/洞/奇/迹/科/技/有/限/公/司

版权声明

商标声明

 和其他成都虫洞奇迹科技有限公司商标均为成都虫洞奇迹科技有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受成都虫洞奇迹科技有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，成都虫洞奇迹科技有限公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

成都虫洞奇迹科技有限公司

电话：400-090-2980

邮箱：contact@lingyuecloud.com

网址：www.lingyuecloud.com

1. 前言

上一篇文章《虚拟化环境搭建及虚拟机创建》中，我们简单介绍了虚拟化环境的搭建和虚拟机创建过程，本文我们将继续介绍虚拟化环境中重要组件 QEMU 的修改和安装包编译。

我们使用上一篇文章中安装的 CentOS 7 X86_64 位版本系统作为编译环境。

2. QEMU 修改及安装包编译

2.1 源码包下载

CentOS 官方会跟随 Red Hat 官方的节奏发布各个组件的新版本源码（CentOS 发布的源码相比于 Red Hat 发布的源码，其中最主要的改变其实是将源码中的 Red Hat 替换为 CentOS，避免侵权问题），读者可以从 http://vault.centos.org 上下载，本文以当前最新的 qemu-kvm-1.5.3-141.el7_4.4.src.rpm 源码包为例进行修改和编译。

2.2 源码包直接编译安装包

下载得到源码包之后，使用 rpmbuild 命令直接就可对其进行编译生成安装包，由于前文我们安装的系统中，并没有 rpmbuild 工具，这里需要手动先安装：

```
[lingyuecloud@lingyuecloud qemu-kvm]$ sudo yum install rpm-build
```

然后，使用“--rebuild”参数直接编译 QEMU 源码包：

```
[lingyuecloud@lingyuecloud qemu-kvm]$ rpmbuild --rebuild qemu-kvm-1.5.3-141.el7_4.4.src.rpm
Installing qemu-kvm-1.5.3-141.el7_4.4.src.rpm
warning: user mockbuild does not exist - using root
warning: group mockbuild does not exist - using root
……
```

```
……
warning: user mockbuild does not exist - using root
warning: group mockbuild does not exist - using root
error: Failed build dependencies:
        zlib-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        SDL-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        gnutls-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        cyrus-sasl-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        libtool is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        libaio-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        pciutils-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        pulseaudio-libs-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        libiscsi-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        ncurses-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        libattr-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        libusbx-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        usbredir-devel >= 0.6 is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        texinfo is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        spice-protocol >= 0.12.2 is needed by qemu-kvm-10:1.5.3-
141.el7.centos.4.x86_64
        spice-server-devel >= 0.12.0 is needed by qemu-kvm-10:1.5.3-
141.el7.centos.4.x86_64
        libseccomp-devel >= 1.0.0 is needed by qemu-kvm-10:1.5.3-
141.el7.centos.4.x86_64
        gperftools-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        libcurl-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        librados2-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        librbd1-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
        glusterfs-api-devel >= 3.6.0 is needed by qemu-kvm-10:1.5.3-
```

```
141.el7.centos.4.x86_64
            glusterfs-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            systemtap is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            systemtap-sdt-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            nss-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            libjpeg-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            libpng-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            libuuid-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            bluez-libs-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            brlapi-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            check-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            libcap-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            pixman-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            rdma-core-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            cpp is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            lzo-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            snappy-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
            libssh2-devel is needed by qemu-kvm-10:1.5.3-141.el7.centos.4.x86_64
```

命令输出了很多 warning（这里省略了部分）。在编译安装包之前，首先会先安装源码 rpm 包（也就是解压源码包，通常解压到~/rpmbuild/SOURCES/目录下），这里的 warning 的意思是说，我们下载的源码包是以 mockbuild 用户和 mockbuild 用户组打包的（mock 软件、mockbuild 用户以及 mockbuild 组用来在 chroot 环境下编译 rpm 包），而本地环境没有 mockbuild 用户和 mockbuild 用户组来安装源码包，将会使用 root 用户和 root 用户组，这个 warning 并不影响编译成功与否。为消除上述 warning，执行下述命令。

```
[lingyuecloud@lingyuecloud qemu-kvm]$ sudo groupadd mockbuild
[lingyuecloud@lingyuecloud qemu-kvm]$ sudo useradd -s /sbin/nologin -g mockbuild mockbuild
```

接下来我们看后面的 error，这些 error 指出编译 QEMU 安装包依赖指出的

软件，而他们并没有被安装。后面我们还将介绍到，rpmbuild 命令之所以知道编译 QEMU 需要这些软件包，是因为这些依赖软件被写在了 spec 文件中。需要注意的是，编译 QEMU 安装包需要的这些依赖包，与运行 QEMU 需要的依赖包（安装 QEMU 的时候会自动检查）并不相同，虽然有些软件包既在编译 QEMU 的时候需要，也在运行 QEMU 的时候需要。

使用以下命令安装所有依赖包：

```
[lingyuecloud@lingyuecloud qemu-kvm]$ sudo yum install zlib-devel SDL-devel gnutls-devel cyrus-sasl-devel libtool libaio-devel pciutils-devel pulseaudio-libs-devel libiscsi-devel ncurses-devel libattr-devel libusbx-devel usbredir-devel texinfo spice-protocol spice-server-devel libseccomp-devel gperftools-devel libcurl-devel librados2-devel librbd1-devel glusterfs-api-devel glusterfs-devel systemtap systemtap-sdt-devel nss-devel libjpeg-devel libpng-devel libuuid-devel bluez-libs-devel brlapi-devel check-devel libcap-devel pixman-devel rdma-core-devel cpp lzo-devel snappy-devel libssh2-devel
```

然后再运行 rpmbuild 命令即可编译成功：

```
[lingyuecloud@lingyuecloud qemu-kvm]$ rpmbuild --rebuild qemu-kvm-1.5.3-141.el7_4.4.src.rpm
Installing qemu-kvm-1.5.3-141.el7_4.4.src.rpm
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.wu0DxD
+ umask 022
+ cd /home/lingyuecloud/rpmbuild/BUILD
+ cd /home/lingyuecloud/rpmbuild/BUILD
+ rm -rf qemu-1.5.3
+ /usr/bin/tar -xf -
+ /usr/bin/bzip2 -dc /home/lingyuecloud/rpmbuild/SOURCES/qemu-1.5.3.tar.bz2
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ cd qemu-1.5.3
+ /usr/bin/chmod -Rf a+rX,u+w,g-w,o-w .
+ cp /home/lingyuecloud/rpmbuild/SOURCES/bios-256k.bin pc-bios
```

```
+ tar -xf /home/lingyuecloud/rpmbuild/SOURCES/sample_images.tar
+ echo 'Patch #1 (0000-libcacard-fix-missing-symbols-in-libcacard.so.patch):'
Patch #1 (0000-libcacard-fix-missing-symbols-in-libcacard.so.patch):
+ /usr/bin/patch -p1 --fuzz=0
+ /usr/bin/cat /home/lingyuecloud/rpmbuild/SOURCES/0000-libcacard-fix-missing-symbols-in-libcacard.so.patch
patching file libcacard/Makefile
……
……
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp.x6IUsO
+ umask 022
+ cd /home/lingyuecloud/rpmbuild/BUILD
+ cd qemu-1.5.3
+ buildarch=x86_64-softmmu
+ extraldflags=-Wl,--build-id
+ buildldflags=VL_LDFLAGS=-Wl,--build-id
+ dobuild --target-list=x86_64-softmmu
+ ./configure --prefix=/usr --libdir=/usr/lib64 --sysconfdir=/etc --interp-prefix=/usr/qemu-%M --audio-drv-list=pa,alsa --with-confsuffix=/qemu-kvm --localstatedir=/var --libexecdir=/usr/libexec --with-pkgversion=qemu-kvm-1.5.3-141.el7.centos.4 --disable-strip --disable-qom-cast-debug '--extra-ldflags=-Wl,--build-id -pie -Wl,-z,relro -Wl,-z,now' '--extra-cflags=-O2 -g -pipe -Wall -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -fPIE -DPIE' --enable-trace-backend=dtrace --enable-werror --disable-xen --disable-virtfs --enable-kvm --enable-libusb --enable-spice --enable-seccomp --disable-fdt --enable-docs --disable-sdl --disable-debug-tcg --disable-sparse --disable-brlapi --disable-bluez --disable-vde --disable-curses --enable-curl --enable-libssh2 --enable-vnc-tls --enable-vnc-sasl --enable-linux-aio --enable-smartcard-nss --enable-lzo --enable-snappy --enable-usb-redir --enable-vnc-png --disable-vnc-jpeg --enable-vnc-ws --enable-uuid --disable-vhost-scsi --disable-guest-agent --disable-live-block-ops --disable-
```

live-block-migration --enable-rbd --enable-glusterfs --enable-tcmalloc --block-drv-rw-whitelist=qcow2,raw,file,host_device,blkdebug,nbd,iscsi,gluster,rbd --block-drv-ro-whitelist=vmdk,vhdx,vpc,ssh,https --iasl=/bin/false --target-list=x86_64-softmmu

Install prefix /usr

BIOS directory /usr/share/qemu-kvm

binary directory /usr/bin

library directory /usr/lib64

libexec directory /usr/libexec

include directory /usr/include

config directory /etc

local state directory /var

Manual directory /usr/share/man

ELF interp prefix /usr/qemu-%M

Source path /home/lingyuecloud/rpmbuild/BUILD/qemu-1.5.3

C compiler cc

Host C compiler cc

Objective-C compiler cc

CFLAGS -O2 -D_FORTIFY_SOURCE=2 -pthread -I/usr/include/glib-2.0 -I/usr/lib64/glib-2.0/include -g

QEMU_CFLAGS -Werror -DHAS_LIBSSH2_SFTP_FSYNC -fPIE -DPIE -m64 -D_GNU_SOURCE -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE -Wstrict-prototypes -Wredundant-decls -Wall -Wundef -Wwrite-strings -Wmissing-prototypes -fno-strict-aliasing -O2 -g -pipe -Wall -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -fPIE -DPIE -Wendif-labels -Wmissing-include-dirs -Wempty-body -Wnested-externs -Wformat-security -Wformat-y2k -Winit-self -Wignored-qualifiers -Wold-style-declaration -Wold-style-definition -Wtype-limits -fstack-protector-strong -I/usr/include/p11-kit-1 -I/usr/include/libpng15 -I/usr/include/spice-server -I/usr/include/cacard -I/usr/include/glib-2.0 -I/usr/lib64/glib-2.0/include -I/usr/include/pixman-1 -I/usr/include/nss3 -I/usr/include/nspr4 -I/usr/include/spice-1 -I/usr/include/nss3 -

I/usr/include/nspr4 -pthread -I/usr/include/glib-2.0 -I/usr/lib64/glib-2.0/include -I/usr/include/libusb-1.0 -I/usr/include/pixman-1

LDFLAGS -Wl,--warn-common -Wl,-z,relro -Wl,-z,now -pie -m64 -g -Wl,--build-id -pie -Wl,-z,relro -Wl,-z,now

make make

install install

python python

smbd /usr/sbin/smbd

host CPU x86_64

host big endian no

target list x86_64-softmmu

……

……

Processing files: qemu-kvm-debuginfo-1.5.3-141.el7.centos.4.x86_64

Provides: qemu-kvm-debuginfo = 10:1.5.3-141.el7.centos.4 qemu-kvm-debuginfo(x86-64) = 10:1.5.3-141.el7.centos.4

Requires(rpmlib): rpmlib(FileDigests) <= 4.6.0-1 rpmlib(PayloadFilesHavePrefix) <= 4.0-1 rpmlib(CompressedFileNames) <= 3.0.4-1

Checking for unpackaged file(s): /usr/lib/rpm/check-files /home/lingyuecloud/rpmbuild/BUILDROOT/qemu-kvm-1.5.3-141.el7_4.4.x86_64

Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-1.5.3-141.el7.centos.4.x86_64.rpm

Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-img-1.5.3-141.el7.centos.4.x86_64.rpm

Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-common-1.5.3-141.el7.centos.4.x86_64.rpm

Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-tools-1.5.3-141.el7.centos.4.x86_64.rpm

Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-debuginfo-1.5.3-

```
141.el7.centos.4.x86_64.rpm
    Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.5aHHpJ
    + umask 022
    + cd /home/lingyuecloud/rpmbuild/BUILD
    + cd qemu-1.5.3
    + /usr/bin/rm -rf /home/lingyuecloud/rpmbuild/BUILDROOT/qemu-kvm-1.5.3-
141.el7_4.4.x86_64
    + exit 0
    Executing(--clean): /bin/sh -e /var/tmp/rpm-tmp.qutzZw
    + umask 022
    + cd /home/lingyuecloud/rpmbuild/BUILD
    + rm -rf qemu-1.5.3
    + exit 0
```

整个编译过程中输出的日志非常长，我们这里只截取了部分。从输出日志的开始可以看到，编译过程中首先是打 Patch（后面我们会将源码 rpm 包解开，可以看到里面有很多 Patch），然后 spec 文件会根据当前环境选择 configure 参数进行 configure，然后执行编译。最后，“Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/******.rpm”是编译完成后输出的 rpm 安装包，可使用“rpm -ivh ******.rpm”命令来安装即可。

使用源码包直接编译是最简单的编译方法，但是其缺点是我们没有对源码或者编译过程进行修改，接下来我们将介绍如何修改源码并编译。

2.3 源码包解压编译安装包

2.3.1 源码包文件分析

首先使用命令对源码包进行解压，看看其中包含的文件：

```
    [lingyuecloud@lingyuecloud qemu]$ rpm2cpio qemu-kvm-1.5.3-141.el7_4.4.src.rpm | cpio
-di
    64177 blocks
```

```
[lingyuecloud@lingyuecloud qemu]$ ls
0000-libcacard-fix-missing-symbols-in-libcacard.so.patch
80-kvm.rules
99-qemu-guest-agent.rules
bios-256k.bin
block-add-block-driver-read-only-whitelist.patch
bridge.conf
change-path-from-qemu-to-qemu-kvm.patch
disable-hpet-device.patch
……
……
e1000-Keep-capabilities-list-bit-on-for-older-RHEL-machine-types.patch
ksmctl.c
ksm.service
ksm.sysconfig
ksmtuned
ksmtuned.conf
ksmtuned.service
kvm-ac97-register-reset-via-qom.patch
……
……
qemu-1.5.3.tar.bz2
qemu.binfmt
qemu-ga.sysconfig
qemu-guest-agent.service
qemu-kvm-1.5.3-141.el7_4.4.src.rpm
qemu-kvm-Fix-migration-from-older-version-due-to-i8254-changes.patch
qemu-kvm.spec
qxl-set-revision-to-1-for-rhel6-0-0.patch
```

```
README.rhel6-gpxe-source
rename-man-page-to-qemu-kvm.patch
rhel6-e1000.rom
rhel6-ne2k_pci.rom
rhel6-pcnet.rom
rhel6-rtl8139.rom
rhel6-virtio.rom
sample_images.tar
target-i386-set-level-4-on-conroe-penryn-nehalem.patch
target-i386-update-model-values-on-conroe-penryn-nehalem-cpu-models.patch
use-kvm-by-default.patch
vga-Default-vram_size_mb-to-16-like-prior-versions-of-RHEL.patch
vmdk-Allow-reading-variable-size-descriptor-files.patch
vmdk-refuse-to-open-higher-version-than-supported.patch
……
```

其中包含了几部分，主要包括：patch 文件（1746 个，此处省略了部分）、QEMU 官方 release 的源码包(qemu-1.5.3.tar.bz2)、编译 QEMU 的时候用到的其他源码及资源文件，以及控制编译的 spec 文件(qemu-kvm.spec)。

spec 文件内容非常多，关于 spec 文件的语法等知识，有兴趣的读者可以参阅 Fedora 官方文档 Creating RPM Packages with Fedora。这里，我们简单看下 spec 文件中的部分内容：

```
[lingyuecloud@lingyuecloud qemu]$ cat qemu-kvm.spec
……
%define pkgname qemu-kvm
%define rhel_suffix -rhel
%define rhev_suffix -rhev
……
Summary: QEMU is a machine emulator and virtualizer
Name: %{pkgname}%{?pkgsuffix}
```

```
Version: 1.5.3

Release: 141%{?dist}.4

# Epoch because we pushed a qemu-1.0 package. AIUI this can't ever be dropped

Epoch: 10

License: GPLv2+ and LGPLv2+ and BSD

Group: Development/Tools

URL: http://www.qemu.org/

ExclusiveArch: x86_64 %{power64} aarch64 s390x

Requires: seabios-bin >= 1.7.2.2-5

Requires: sgabios-bin

Requires: seavgabios-bin

Requires: ipxe-roms-qemu

……

……

Source0: http://wiki.qemu-project.org/download/qemu-%{version}.tar.bz2


Source1: qemu.binfmt

……

BuildRequires: zlib-devel

BuildRequires: SDL-devel

BuildRequires: which

BuildRequires: gnutls-devel

BuildRequires: cyrus-sasl-devel

BuildRequires: libtool

BuildRequires: libaio-devel

BuildRequires: rsync

BuildRequires: python

BuildRequires: pciutils-devel

BuildRequires: pulseaudio-libs-devel
```

```
BuildRequires: libiscsi-devel
BuildRequires: ncurses-devel
BuildRequires: libattr-devel
BuildRequires: libusbx-devel
……
……
```

其中，“Version 和 Release”控制了最终输出 rpm 包的版本号，其中的 dist 变量在我们环境中值为.el7.centos。“Source0”值指定了源码包的下载路径，本地编译的时候，不会从这个 url 去下载源码，只会根据这个 url 末尾的文件名从本地寻找源码包（本例中为 qemu-1.5.3.tar.bz2），若用户希望将修改后的源码打包为.tar.gz 格式或其他格式，修改此处 url 中的后缀名即可。“Requires”字段指定了编译生成的 QEMU 安装包在安装时所需要的依赖软件，“BuildRequires”指定了在编译时依赖的软件包。

2.3.2 编译安装包

要编译 rpm 包，只需将前文解压出来的所有文件复制到编译目录“~/rpmbuild/SOURCES/”（执行安装命令“rpm -ivh qemu-kvm-1.5.3-141.el7_4.4.src.rpm”也会将所需文件解压到编译目录），然后执行编译命令即可编译成功。

```
[lingyuecloud@lingyuecloud SOURCES]$ rpmbuild -bb qemu-kvm.spec
```

接下来，我们修改 qemu-1.5.3.tar.bz2 中的源码。需要特别注意的是，因为 spec 文件中有打 Patch 的过程，对 qemu-1.5.3.tar.bz2 源码的修改很可能导致后续的打 Patch 过程失败，所以，实际进行项目开发时，通常是首先将 Patch 打完，之后注释掉 spec 文件中打 Patch 的过程，最后在此基础上进行开发。

由于 Patch 较多，本文演示的改动较小，因此暂时先不打 Patch，直接修改源码进行编译。

首先解压 qemu-1.5.3.tar.bz2：

```
[lingyuecloud@lingyuecloud SOURCES]$ tar -jxf qemu-1.5.3.tar.bz2
[lingyuecloud@lingyuecloud SOURCES]$ cd qemu-1.5.3
```

编辑 vl.c 文件，在 QEMU 的 main_loop 函数中增加一行日志：

```
……
……
static void main_loop(void)
{
    bool nonblocking;
    int last_io = 0;
#ifdef CONFIG_PROFILER
    int64_t ti;
#endif
    fprintf(stderr, "Hello, welcome to lingyuecloud.com\n");
    do {
        nonblocking = !kvm_enabled() && !xen_enabled() && last_io > 0;
#ifdef CONFIG_PROFILER
        ti = profile_getclock();
#endif
        last_io = main_loop_wait(nonblocking);
#ifdef CONFIG_PROFILER
        dev_time += profile_getclock() - ti;
#endif
    } while (!main_loop_should_exit());
}
……
……
```

然后修改 spec 文件，更新 release 版本号以示区别：

```
……
Summary: QEMU is a machine emulator and virtualizer
Name: %{pkgname}%{?pkgsuffix}
Version: 1.5.3
Release: 141.20171220.lingyuecloud%{?dist}.4
```

```
# Epoch because we pushed a qemu-1.0 package. AIUI this can't ever be dropped
Epoch: 10
License: GPLv2+ and LGPLv2+ and BSD
……
……
```

将修改后的源码打包并编译：

```
[lingyuecloud@lingyuecloud SOURCES]$ mv qemu-1.5.3.tar.bz2 qemu-1.5.3.tar.gz.bak
[lingyuecloud@lingyuecloud SOURCES]$ tar -jcf qemu-1.5.3.tar.bz2 qemu-1.5.3
[lingyuecloud@lingyuecloud SOURCES]$ rpmbuild -bb qemu-kvm.spec
……
……
Processing files: qemu-kvm-debuginfo-1.5.3-141.20171220.lingyuecloud.el7.centos.4.x86_64
Provides: qemu-kvm-debuginfo = 10:1.5.3-141.20171220.lingyuecloud.el7.centos.4 qemu-kvm-debuginfo(x86-64) = 10:1.5.3-141.20171220.lingyuecloud.el7.centos.4
Requires(rpmlib): rpmlib(FileDigests) <= 4.6.0-1 rpmlib(PayloadFilesHavePrefix) <= 4.0-1 rpmlib(CompressedFileNames) <= 3.0.4-1
Checking for unpackaged file(s): /usr/lib/rpm/check-files /home/lingyuecloud/rpmbuild/BUILDROOT/qemu-kvm-1.5.3-141.20171220.lingyuecloud.el7.centos.4.x86_64
Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-1.5.3-141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm
Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-img-1.5.3-141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm
Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-common-1.5.3-141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm
Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-tools-1.5.3-141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm
Wrote: /home/lingyuecloud/rpmbuild/RPMS/x86_64/qemu-kvm-debuginfo-1.5.3-
```

```
141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm
    Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.0ckghW
    + umask 022
    + cd /home/lingyuecloud/rpmbuild/BUILD
    + cd qemu-1.5.3
    + /usr/bin/rm -rf /home/lingyuecloud/rpmbuild/BUILDROOT/qemu-kvm-1.5.3-
141.20171220.lingyuecloud.el7.centos.4.x86_64
    + exit 0
```

可以看到，输出的 rpm 包版本号已经更新为我们设置的版本号。

接下来，我们更新安装编译好的 rpm 包，以检查我们的源码修改是否生效：

```
    [lingyuecloud@lingyuecloud SOURCES]$ cd
/home/lingyuecloud/rpmbuild/RPMS/x86_64/
    [lingyuecloud@lingyuecloud x86_64]$ sudo rpm -Uvh qemu-kvm-1.5.3-
141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm qemu-img-1.5.3-
141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm qemu-kvm-common-1.5.3-
141.20171220.lingyuecloud.el7.centos.4.x86_64.rpm
    [sudo] password for lingyuecloud:
    Preparing... ##################################
[100%]
    Updating / installing...
       1:qemu-kvm-common-10:1.5.3-141.2017################################
[ 17%]
       2:qemu-img-10:1.5.3-141.20171220.li################################ [ 33%]
       3:qemu-kvm-10:1.5.3-141.20171220.li################################
[ 50%]
    Cleaning up / removing...
       4:qemu-kvm-10:1.5.3-141.el7 ################################
[ 67%]
       5:qemu-img-10:1.5.3-141.el7 ################################
```

```
[ 83%]
        6:qemu-kvm-common-10:1.5.3-141.el7      ##################################
[100%]
```

然后，按照《虚拟化环境搭建及虚拟机创建》文章所述，使用 libvirt 启动虚拟机，查看虚拟机日志（在目录“~/.cache/libvirt/qemu/log/”下面，文件名为虚拟机名），即可看到我们加入的日志，如图所示：

```
qemu: terminating on signal 15 from pid 7661
2017-11-11 02:44:58.395+0000: shutting down, reason=destroyed
2017-12-20 14:39:19.115+0000: starting up libvirt version: 3.2.0, package: 14.el7_4.3 (Cen
tOS BuildSystem <http://bugs.centos.org>, 2017-09-07-11:27:44, c1bm.rdu2.centos.org), qemu
 version: 1.5.3 (qemu-kvm-1.5.3-141.20171220.lingyuecloud.el7.centos.4), hostname: lingyue
cloud.com
LC_ALL=C PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/lingyuecloud/.local/
bin:/home/lingyuecloud/bin HOME=/home/lingyuecloud USER=lingyuecloud LOGNAME=lingyuecloud
QEMU_AUDIO_DRV=none /usr/libexec/qemu-kvm -name lingyuecloud -S -machine pc-i440fx-rhel7.0
.0,accel=tcg,usb=off,dump-guest-core=off -m 256 -realtime mlock=off -smp 1,sockets=1,cores
=1,threads=1 -uuid e5f8aa3a-8fc1-4df9-8a9d-593e305fdeea -no-user-config -nodefaults -chard
ev socket,id=charmonitor,path=/home/lingyuecloud/.config/libvirt/qemu/lib/domain-1-lingyue
cloud/monitor.sock,server,nowait -mon chardev=charmonitor,id=monitor,mode=control -rtc bas
e=utc,driftfix=slew -global kvm-pit.lost_tick_policy=delay -no-hpet -no-shutdown -boot str
ict=on -device piix3-usb-uhci,id=usb,bus=pci.0,addr=0x1.0x2 -drive file=/home/lingyuecloud
/cirros-0.3.5-x86_64-disk.img,format=qcow2,if=none,id=drive-virtio-disk0,cache=none -devic
e virtio-blk-pci,scsi=off,bus=pci.0,addr=0x3,drive=drive-virtio-disk0,id=virtio-disk0,boot
index=1 -device usb-tablet,id=input0,bus=usb.0,port=1 -vnc 0.0.0.0:0 -k en-us -vga cirrus
-device virtio-balloon-pci,id=balloon0,bus=pci.0,addr=0x4 -msg timestamp=on
Hello, welcome to lingyuecloud.com
~
~
~
~
~
~
~
~
~
"~/.cache/libvirt/qemu/log/lingyuecloud.log" [readonly] 17L, 6155C      17,1          Bot
```

图 1 虚拟机的 QEMU 日志

2.4 Configure & Make

除了使用 spec 文件来编译完整的安装包之外，还可以在源码目录中直接使用 configure 和 make 命令进行编译。这种方式的好处是编译过一次之后，只要没有清除编译结果（执行“make clean”命令），下次编译将只会编译修改了源码的部分文件，大大节省了时间。当然，该方式也存在缺点：首先是编译过程依赖的软件包不清晰，如果遇到依赖的软件不存在，出错提示信息也并不直观；其次，使用 rpmbuild 方式编译安装包时，可以在 spec 文件中加入测试过程，对编译完成的文件进行一些测试，确保生成的安装包功能正常，而采用 Configure & Make 方式编译，测试过程需要手动执行；再次，编译生成的文件并没有很好的组织起

来，不便于安装到其他环境运行。Configure & Make 编译方式通常在开发过程中使用，软件发布时一般不会使用该编译方式。

在前文解压 qemu-1.5.3 源码目录中，执行./configure（configure 后面可加参数，如果没有加参数，configure 过程将会自动进行。读者可参考前文 rpmbuild 日志中的 configure 参数，或者执行“./configure --help”命令查看各个参数的含义），然后执行 make 命令即可进行编译。由于前文已经将编译需要的依赖软件安装完成，此处的编译将会很顺利的完成，读者可自行尝试。

3. 总结

本文简要介绍了虚拟化组件 QEMU 的安装包编译过程，简单修改了 QEMU 源码，并验证了修改的有效性。由于篇幅所限，控制 rpm 编译过程的 spec 文件，以及 QEMU 的源码分析等，都并没有深入展开叙述，后续的文章中，我们将陆续介绍，敬请期待。