ANRC

RedHat Linux Kernel Internals Laboratory Exercises

Lab 1: Development Setup and Exploration

Objective: Setup your RedHat workstation for Linux Kernel Development and get familiar with Eclipse.

```
[root@rhki user]# cd RHKI/
[root@rhki RHKI]# ls -alrt
total 28
drwxrwxr-x.  2 user user 4096 Jun 19 09:55 media
drwxrwxr-x.  2 user user 4096 Jun 19 09:55 slides
drwxrwxr-x.  2 user user 4096 Jun 19 09:55 labs
drwxrwxr-x.  2 user user 4096 Jun 19 09:56 apps
drwxrwxr-x.  2 user user 4096 Jun 19 09:56 workspace
drwxrwxr-x.  7 user user 4096 Jun 19 09:56 .
drwx------. 27 user user 4096 Jul  1 11:46 ..
[root@rhki RHKI]#
```

1. Login into your workstation using the credentials: **user**:**pa22w0rd** (the sudo account has the same password).
2. Open a terminal and navigate to the RHKI directory in your home folder.
3. This directory contains the files, media, and laboratory exercises you will be using for this course.
4. Change to the media directory.
5. The binaries for this distribution of RHEL are stored in these folders.
6. Install the following RPM packages to get your system ready for kernel development:

    - **sudo yum groupinstall "Development Tools"**
    - **sudo yum install kernel-devel kernel-headers**

7. Install KDump (Kernel crash dump) and trace utilities.

    - **yum install kexec-tools system-config-kdump crash crash-gcore-command crash-trace-command**

8. Reboot to get Kdump running. Verify installation in Grub boot menu by editing boot command line. Post boot-up you can also verify everything is working properly by checking the status of the kdump service.

```
[ 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.]

<8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto  KEYBOARDTYPE=pc KEYTAB>
```

```
[user@localhost RHKI]$ sudo systemctl status kdump.service
kdump.service - Crash recovery kernel arming
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled)
   Active: active (exited) since Mon 2014-09-08 12:16:46 EDT; 55min ago
  Process: 1640 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/S
 Main PID: 1640 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/kdump.service

Sep 08 12:16:45 localhost.localdomain dracut[3939]: drwxr-xr-x   3 root
Sep 08 12:16:45 localhost.localdomain dracut[3939]: drwxr-xr-x   3 root
Sep 08 12:16:45 localhost.localdomain dracut[3939]: drwxr-xr-x   2 root
Sep 08 12:16:45 localhost.localdomain dracut[3939]: lrwxrwxrwx   1 root
Sep 08 12:16:45 localhost.localdomain dracut[3939]: lrwxrwxrwx   1 root
Sep 08 12:16:45 localhost.localdomain dracut[3939]: lrwxrwxrwx   1 root
Sep 08 12:16:45 localhost.localdomain dracut[3939]: ==========
Sep 08 12:16:46 localhost.localdomain kdumpctl[1640]: kexec: loaded kdump
Sep 08 12:16:46 localhost.localdomain kdumpctl[1640]: Starting kdump: [OK]
Sep 08 12:16:46 localhost.localdomain systemd[1]: Started Crash recovery k
```

9. Prepare the Kernel source build directory.
   Note: For this course the path is /home/user/rpmbuild.

   - **cd /usr/src**
   - **sudo mkdir redhat**
   - **sudo mkdir redhat/{SPECS,SOURCES,RPMS,SRPMS,BUILD}**

```
[user@localhost src]$ sudo mkdir redhat/{SPECS,SOURCES,RPMS,SRPMS,BUILD}
[user@localhost src]$ ls -lart redhat/
total 28
drwxr-xr-x. 5 root root 4096 Jul  1 18:55 ..
drwxr-xr-x. 2 root root 4096 Jul  1 18:59 SRPMS
drwxr-xr-x. 2 root root 4096 Jul  1 18:59 SPECS
drwxr-xr-x. 2 root root 4096 Jul  1 18:59 SOURCES
drwxr-xr-x. 2 root root 4096 Jul  1 18:59 RPMS
drwxr-xr-x. 2 root root 4096 Jul  1 18:59 BUILD
drwxr-xr-x. 7 root root 4096 Jul  1 18:59 .
```

10. To build the RHEL kernel we will need the source code from the “Source” DVD ISO. Mount the ISO and install the kernel source RPM. (You can safely ignore the warnings).

    The source kernel source tree will be expanded in”/home/user/rpmbuild”.

    - **cd ~/rpmbuild**

```
[user@localhost ~]$ cd ~/rpmbuild/
[user@localhost rpmbuild]$ ls -alrt
total 16
drwx------. 28 user user 4096 Jul  1 19:28 ..
drwxr-xr-x.  4 user user 4096 Jul  1 19:28 .
drwxr-xr-x.  2 user user 4096 Jul  1 19:28 SPECS
drwxr-xr-x.  2 user user 4096 Jul  1 19:28 SOURCES
[user@localhost rpmbuild]$ pwd
/home/user/rpmbuild
[user@localhost rpmbuild]$
```

As of RHEL 7 we also need to add the following RPMs required to build the Kernel:

```
[user@localhost rpmbuild]$ rpmbuild -bp --target=`uname -m` SPECS/kernel.spec
Building target platforms: x86_64
Building for target x86_64
error: Failed build dependencies:
        xmlto is needed by kernel-3.10.0-123.el7.x86_64
        asciidoc is needed by kernel-3.10.0-123.el7.x86_64
        hmaccalc is needed by kernel-3.10.0-123.el7.x86_64
        pesign >= 0.109-4 is needed by kernel-3.10.0-123.el7.x86_64
        elfutils-devel is needed by kernel-3.10.0-123.el7.x86_64
        zlib-devel is needed by kernel-3.10.0-123.el7.x86_64
        binutils-devel is needed by kernel-3.10.0-123.el7.x86_64
        newt-devel is needed by kernel-3.10.0-123.el7.x86_64
        python-devel is needed by kernel-3.10.0-123.el7.x86_64
        perl(ExtUtils::Embed) is needed by kernel-3.10.0-123.el7.x86_64
        audit-libs-devel is needed by kernel-3.10.0-123.el7.x86_64
        numactl-devel is needed by kernel-3.10.0-123.el7.x86_64
        pciutils-devel is needed by kernel-3.10.0-123.el7.x86_64
[user@localhost rpmbuild]$
```

Note: Some packages might be in the optional repos.

```
[user@localhost SOURCES]$ cd ..
[user@localhost rpmbuild]$ rpmbuild -bp --target=`uname -m` SPECS/kernel.spec
Building target platforms: x86_64
Building for target x86_64
error: Failed build dependencies:
        xmlto is needed by kernel-3.10.0-123.el7.x86_64
        asciidoc is needed by kernel-3.10.0-123.el7.x86_64
        hmaccalc is needed by kernel-3.10.0-123.el7.x86_64
        pesign >= 0.109-4 is needed by kernel-3.10.0-123.el7.x86_64
        elfutils-devel is needed by kernel-3.10.0-123.el7.x86_64
        zlib-devel is needed by kernel-3.10.0-123.el7.x86_64
        binutils-devel is needed by kernel-3.10.0-123.el7.x86_64
        newt-devel is needed by kernel-3.10.0-123.el7.x86_64
        python-devel is needed by kernel-3.10.0-123.el7.x86_64
        perl(ExtUtils::Embed) is needed by kernel-3.10.0-123.el7.x86_64
        audit-libs-devel is needed by kernel-3.10.0-123.el7.x86_64
        numactl-devel is needed by kernel-3.10.0-123.el7.x86_64
        pciutils-devel is needed by kernel-3.10.0-123.el7.x86_64
[user@localhost rpmbuild]$ sudo yum install --enablerepo="*optional*" xmlto asciidoc
[sudo] password for user:
Loaded plugins: langpacks, product-id, subscription-manager
rhel-7-server-optional-beta-debug-rpms
rhel-7-server-optional-beta-rpms
rhel-7-server-optional-beta-source-rpms
```

Before we can configure it (in the next lab) we need to prepare the source.

- **rpmbuild -bp --target=`uname -m` SPECS/kernel.spec**

"-bp" means build the patched kernel source files. They will be placed under "/BUILD" directory.

Note: If you are planning on building a custom kernel you must configure the kernel.spec file. We will show you how to do this in Lab 2.

```
Complete!
[user@localhost rpmbuild]$ rpmbuild -bp --target=`uname -m` SPECS/kernel.spec
Building target platforms: x86_64
Building for target x86_64
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.e1PmO6
+ umask 022
+ cd /home/user/rpmbuild/BUILD
+ patch_command='patch -p1 -F1 -s'
+ cd /home/user/rpmbuild/BUILD
+ rm -rf kernel-3.10.0-123.el7
+ /usr/bin/mkdir -p kernel-3.10.0-123.el7
+ cd kernel-3.10.0-123.el7
+ /usr/bin/tar -xf -
+ /usr/bin/xz -dc /home/user/rpmbuild/SOURCES/linux-3.10.0-123.el7.tar.xz
```

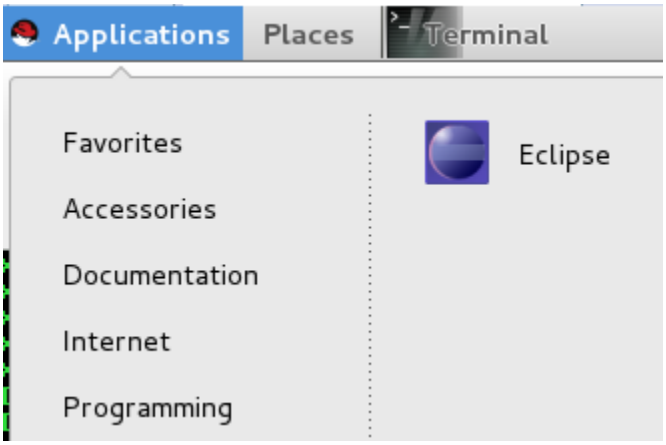
Navigate to "**/home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/**" and list the directories to view the kernel directory source tree.

```
[user@localhost rpmbuild]$ cd BUILD/kernel-3.10.0-123.el7/linux-3
[user@localhost linux-3.10.0-123.el7.x86_64]$ ls -ld */
drwxr-xr-x.  32 user user 4096 Sep  8 13:25 arch/
drwxr-xr-x.   3 user user 4096 May  5 10:58 block/
drwxr-xr-x.   2 user user   80 Sep  8 13:25 configs/
drwxr-xr-x.   4 user user 4096 May  5 10:58 crypto/
drwxr-xr-x. 100 user user 8192 Sep  8 13:25 Documentation/
drwxr-xr-x. 112 user user 4096 May  5 10:58 drivers/
drwxr-xr-x.  36 user user 4096 Sep  8 13:25 firmware/
drwxr-xr-x.  73 user user 4096 May  5 10:58 fs/
drwxr-xr-x.  27 user user 4096 Sep  8 13:25 include/
drwxr-xr-x.   2 user user 4096 May  5 10:58 init/
drwxr-xr-x.   2 user user 4096 May  5 10:58 ipc/
drwxr-xr-x.  11 user user 4096 Sep  8 13:25 kernel/
drwxr-xr-x.   9 user user 8192 Sep  8 13:25 lib/
drwxr-xr-x.   2 user user 4096 May  5 10:58 mm/
drwxr-xr-x.  56 user user 4096 May  5 10:58 net/
drwxr-xr-x.  12 user user 4096 May  5 10:58 samples/
drwxr-xr-x.  13 user user 4096 Sep  8 13:25 scripts/
drwxr-xr-x.   9 user user 4096 May  5 10:58 security/
drwxr-xr-x.  22 user user 4096 May  5 10:58 sound/
drwxr-xr-x.  17 user user 4096 May  5 10:58 tools/
drwxr-xr-x.   2 user user   80 Sep  8 13:25 usr/
drwxr-xr-x.   3 user user   16 May  5 10:58 virt/
```

11. Finally, we will be using Eclipse as the IDE for this course. To install it:

    - **sudo yum install eclipse-cdt**

12. It is located under Applications, Programming:



13. We will configure Eclipse in Lab 3 (Building our First module) for Kernel compilation.
14. Proceed to Lab 2 to configure and compile the RHEL Kernel.