Building RPMs (Lab) | Coursera

coursera.org/learn/linux-tools-for-developers/supplement/ipAAI/building-rpms-lab

Exercise

Note: If you are on a RPM packaging system, you shoujld to this exercise. If youare on an APT packaging system, youi should do the following one.

We give you the source files for a trivial **Hello world** application:

Makefile
myhello.c
README
We also give you a slightly modified version of the source:

Makefile - modified
myhello.c - modified.c
README - modified
Note the source directory includes a **README** file; without it, some versions of RPM will bail out in error; it is always good practice to have one anyway.

Construct the patch file, i.e. by doing something like:

```
$ diff -Nur my_app_1.0.0 my_app_1.0.0_PATCHED > my_app-1.0.0.patch
```

Write a **spec** file. Construct source and binary RPM's using **rpmbuild**.

Install and test the binary **rpm** by doing:

```
$ sudo rpm -ivh $HOME/rpmbuild/my_app-1.0.0.x86_64.rpm

/usr/local/binmyhello
```

and then remove it with:

```
 $ sudo rpm -e my_app
```

Also, try rebuilding from the source package:

```
$ rpm --rebuild $HOME/rpmbuild/SRPMS/my_app-1.0.0-1.src.rpm
```

and test as before.

Note the script **build_rpm.sh** will demonstrate one possible way to do the procedure.

Solution

build_rpm.sh
nomake.sh
my_app-1.0.0.spec
RHEL7_my_app-1.0.0.spec