

# Building RPMs (Lab)

 [coursera.org/learn/linux-tools-for-developers/supplement/ipAAI/building-rpms-lab](https://coursera.org/learn/linux-tools-for-developers/supplement/ipAAI/building-rpms-lab)

## Exercise

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

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

 Makefile File

[Download file](#) 


 myhelloC File

[Download file](#) 


 README File

[Download file](#) 


We also give you a slightly modified version of the source:

 Makefile - modified File

[Download file](#) 

 myhello.c - modifiedC File

[Download file](#) 

 README - modified File

[Download file](#) 

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:

```
1
```

```
$ 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:

```
1
```

2

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



and then remove it with:

1

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



Also, try rebuilding from the source package:

1

```
$ 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\_rpmSH File

Download file 

 nomakeSH File

Download file 

 my\_app-1.0.0SPEC File

Download file 

 RHEL7\_my\_app-1.0.0SPEC File

Download file 



## Completed

---