Using gfortran with external libraries and module files

Suppose we have the following directory structure

`/home/baf1/modules` (directory containing module `.mod` files)

`/home/baf1/lib` (directory containing object code libraries as `.a` files)

Compile files containing modules and external subprograms using

```
gfortran -c <source filename>
```

If file `twomods.f90` that contains modules `mod1` and `mod2` then

```
gfortran -c twomods.f90
```

produces the files `twomods.o`, `mod1.mod`, and `mod2.mod`

If file `twosubs.f90` that contains subroutine `sub1` and function `fun1` then

```
gfortran -c twosubs.f90
```

produces the file `twosubs.o`

Copy `.mod` files to the module directory

```
cp *.mod <module directory>
```

In our example

```
cp mod1.mod mod2.mod /home/baf1/modules
```

Add object files to the appropriate object code archive (library) using

```
ar -qs <library filename> <list of object file names>
```

In our example, adding files `twomods.o` and `twosubs.o` to library file `libbase1.a`

```
ar -qs /home/baf1/lib/libbase1.a twomods.o twosubs.o
```

Note: during the link phase below, the library file name is assume to be prefaced with `lib`, so use that convention when creating the file.

Note: to **replace** an object file in a archive (library) use

```
ar -rs <library filename> <list of object file names>
```

In our example, to replace existing file `twosubs.o` in library file `libbase1.a` with a new version

```
ar -rs /home/baf1/lib/libbase1.a twosubs.o
```

Compile Fortran source code that uses modules and object code stored in a library

```
gfortran <source filename> -J<path to modules directory> \
    -L<path to library directory> -l<library name> \
    -o <executable filename>
```

To compile `laba4.f90` which references `mod1`, `mod2`, and `fun1`

```
gfortran laba4.f90 -J/home/baf1/modules -L/home/baf1/lib \
    -lbase1 -o laba4
```