CHAPTER

SIX

MISCELLANEOUS

6.1 Makefiles

The directory *$UWHPSC/codes/fortran/multifile1* contains a Fortran code *fullcode.f90* that consists of a main program and two subroutines:

```
! $UWHPSC/codes/fortran/multifile1/fullcode.f90

program demo
    print *, "In main program"
    call sub1()
    call sub2()
end program demo

subroutine sub1()
    print *, "In sub1"
end subroutine sub1

subroutine sub2()
    print *, "In sub2"
end subroutine sub2
```

To illustrate the construction of a Makefile, we first break this up into three separate files:

```
! $UWHPSC/codes/fortran/multifile1/main.f90

program demo
    print *, "In main program"
    call sub1()
    call sub2()
end program demo
```

```
! $UWHPSC/codes/fortran/multifile1/sub1.f90

subroutine sub1()
    print *, "In sub1"
end subroutine sub1
```

```
! $UWHPSC/codes/fortran/multifile1/sub2.f90

subroutine sub2()
    print *, "In sub2"
end subroutine sub2
```

The directory *$UWHPSC/codes/fortran/multifile1* contains several Makefiles that get successively more sophisticated to compile the codes in this directory.

In the first version we write out explicitly what to do for each file:

```
# $UWHPSC/codes/fortran/multifile1/Makefile

output.txt: main.exe
        ./main.exe > output.txt

main.exe: main.o sub1.o sub2.o
        gfortran main.o sub1.o sub2.o -o main.exe

main.o: main.f90
        gfortran -c main.f90
sub1.o: sub1.f90
        gfortran -c sub1.f90
sub2.o: sub2.f90
        gfortran -c sub2.f90
```

In the second version there is a general rule for creating *.o* files from *.f90* files:

```
# $UWHPSC/codes/fortran/multifile1/Makefile2

output.txt: main.exe
        ./main.exe > output.txt

main.exe: main.o sub1.o sub2.o
        gfortran main.o sub1.o sub2.o -o main.exe

%.o : %.f90
        gfortran -c $<
```

In the third version we define a macro *OBJECTS* so we only have to write out this list once, which minimizes the chance of introducing errors:

```
# $UWHPSC/codes/fortran/multifile1/Makefile3

OBJECTS = main.o sub1.o sub2.o

output.txt: main.exe
        ./main.exe > output.txt

main.exe: $(OBJECTS)
        gfortran $(OBJECTS) -o main.exe

%.o : %.f90
        gfortran -c $<
```

In the fourth version, we add a Fortran compile flag (for level 3 optimization) and an linker flag (blank in this example):

```
# $UWHPSC/codes/fortran/multifile1/Makefile4

FC = gfortran
FFLAGS = -O3
LFLAGS =
OBJECTS = main.o sub1.o sub2.o

output.txt: main.exe
        ./main.exe > output.txt

main.exe: $(OBJECTS)
        $(FC) $(LFLAGS) $(OBJECTS) -o main.exe

%.o : %.f90
        $(FC) $(FFLAGS) -c $<
```

Next we add a *phony* target *clean* that removes the files created when compiling the code in order to facilitate

cleanup. It is *phony* because it does not create a file named *clean*.

```
# $UWHPSC/codes/fortran/multifile1/Makefile5

OBJECTS = main.o sub1.o sub2.o
.PHONY: clean

output.txt: main.exe
        ./main.exe > output.txt

main.exe: $(OBJECTS)
        gfortran $(OBJECTS) -o main.exe

%.o : %.f90
        gfortran -c $<

clean:
        rm -f $(OBJECTS) main.exe
```

Finally we add a help message so that *make help* says something useful:

```
# $UWHPSC/codes/fortran/multifile1/Makefile6

OBJECTS = main.o sub1.o sub2.o
.PHONY: clean help

output.txt: main.exe
        ./main.exe > output.txt

main.exe: $(OBJECTS)
        gfortran $(OBJECTS) -o main.exe

%.o : %.f90
        gfortran -c $<

clean:
        rm -f $(OBJECTS) main.exe

help:
        @echo "Valid targets:"
        @echo "  main.exe"
        @echo "  main.o"
        @echo "  sub1.o"
        @echo "  sub2.o"
        @echo "  clean:  removes .o and .exe files"
```

Fancier things are also possible, for example automatically detecting all the *.f90* files in the directory to construct the list of *SOURCES* and *OBJECTS*:

```
# $UWHPSC/codes/fortran/multifile1/Makefile7

SOURCES = $(wildcard *.f90)
OBJECTS = $(subst .f90,.o,$(SOURCES))

.PHONY: test

test:
        @echo "Sources are: " $(SOURCES)
        @echo "Objects are: " $(OBJECTS)
```

6.1.1 Further reading

- http://software-carpentry.org/4_0/make/