Makefile

PHYS2G03

McMaster University

Unix and Compiling Summary
or What do you really need to know?

- How to log-in to phys-ugrad with ssh (options: Mobaxterm, ssh, MacOSX: xterm + ssh)
- How to look and move files (ls, cd, cp, mv, rm, mkdir, rmdir)
- How to edit files (gedit, xemacs; .cpp)
- How to compile a program (c++) ← Make
- How to run your programs

hello.cpp
A basic C++ source file

```
#include <iostream>


int main()
{
  std::cout << "Hello World!\n";
}
```

Making program hello (by hand)

```
[wadsley@phys-ugrad ~]$ cp -r /home/2G03/hello .
[wadsley@phys-ugrad ~/hello]$ ls
hello.cpp  Makefile
[wadsley@phys-ugrad ~/hello]$ c++ hello.cpp -c
[wadsley@phys-ugrad ~/hello]$ c++ hello.o -o hello
[wadsley@phys-ugrad ~/hello]$ ls -l
total 24
total 24
-rwxrwxr-x 1 wadsley wadsley 8800 Sep 15 13:53 hello*
-rw-r--r-- 1 wadsley wadsley   69 Sep 14 23:21 hello.cpp
-rw-rw-r-- 1 wadsley wadsley 2480 Sep 15 13:57 hello.o
[wadsley@phys-ugrad ~]$ hello
Hello World!
```

Compile

Link

run!

Compiling and Linking

```
$ c++ hello.cpp -c
$ c++ hello.o -o hello

$ c++ hello.cpp -o hello
```

Compile

Link

Compile & Link in one go!

Same outcome:
For now we can just use the all in one go approach

Advantages to doing in stages for large projects
And also, can be automated with **make**

Makefile rules:

If there is a Makefile in the directory, the **make** program can compile and link automatically. You write the compile and link instruction into the **Makefile** once and it applies them for you.

It needs rules to tell it what to do:

```
Thing_to_make: things_it_needs
<TAB> How to make it
```

e.g. The Makefile for hello in /home/2G03/hello

```
hello.o: hello.cpp
        c++ –c hello.cpp
```

Makefile for hello

```
hello: hello.o
	c++ hello.o -o hello

hello.o: hello.cpp
	c++ hello.cpp -c
```

If hello.cpp changes or hello.o doesn't exist →
make will compile to create hello.o
If hello.o changes or hello doesn't exist →
make will link to create hello

Making program hello

```
[wadsley@phys-ugrad ~]$ cd hello
[wadsley@phys-ugrad ~/hello]$ gedit hello.cpp &
[wadsley@phys-ugrad ~/hello]$ make hello
c++ hello.cpp -c
c++ hello.o -o hello
[wadsley@phys-ugrad ~/hello]$ make hello
make: `hello' is up to date.
[wadsley@phys-ugrad ~/hello]$ rm hello
[wadsley@phys-ugrad ~/hello]$ make hello
c++ hello.o -o hello
[wadsley@phys-ugrad ~/hello]$
```

edit hello.cpp

make

hello.cpp changed: Compile

hello.o changed: Link

make

Nothing needs to be done!

remove hello.cpp

make

hello needed: Link only

make vs. compile by hand

You can compile by hand for now.

For future homeworks and activities we will sometimes provide a **Makefile**

e.g. **/home/2G03/hello/Makefile**

(**make** usually fails without one)

Eventually, you will be expected to use make and Makefiles

Making a Makefile

- The Makefile must be in the same directory as your source files
- Copy an existing Makefile (e.g. from a HW folder) into the directory with your source file
- Edit to replace the old source files (e.g. hello.cpp) with the name of your new one
- Change the program name if necessary after the –o, to –o *programname*
- Then try: make programname

Makefile for many files & functions

/home/2G03/mandel/Makefile

```
mandel: MandelMain.o MandelCount.o
        c++ -o mandel MandelMain.o MandelCount.o -
  ltrapfpe -lpgplot -lcpgplot -lX11

MandelMain.o: MandelMain.cpp Mandel.h
        c++ -c MandelMain.cpp

MandelCount.o: MandelCount.cpp Mandel.h
        c++ -c MandelCount.cpp
```

This program uses two source files:
MandelCount.cpp (counter function) and
MandelMain.cpp (main function)
The final program is called **mandel**

Functions: Dependencies

MandelCount.cpp

```
#include <complex>
#include "Mandel.h"

int iCountIterations( std::complex<float> c, int
    nMaxIter ) {
  std::complex<float> z;
  int nIter;

  z=c;
  for (nIter=1;nIter<nMaxIter;nIter++) {
    if (abs(z) > 2.0) break;
    z = z*z + c;
  }
  return nIter;
}
```

Mandel.h

```
int iCountIterations(
    std::complex<float> c, int nMaxIter
    );
```

MandelMain.cpp

```
#include <iostream>
#include <complex>
#include <algorithm>
#include "cpgplot.h"
#include "Mandel.h"

int main()
{
 …
  if (cpgbeg(0,"/xwindow",1,1) != 1) return 1;
 …
  // plot loop: keep redrawing the mandelbrot set until user
     selects exit
  for (;;) {
    // create an array of pixel colours based on the current
       position (x0,y0)
    // and viewing window size (xrange, yrange)
    for (j=0;j<nPix;j++) {
      cImag = (j-(nPix-1)*.5)*yrange/nPix + y0;
      for (i=0;i<nPix;i++) {
        cReal = (i-(nPix-1)*.5)*xrange/nPix + x0;

        // Find number of interations
        nIter = iCountIterations( std::complex<float>(cReal,cImag),
      nMaxIter );
```

MandelMain.cpp uses iCountIterations

MandelCount.cpp **defines** the function iCountIterations

Mandel.h contains the **declaration** for iCountIterations so Mandel.h is needed to compile both these source files

Makefile for 2 .cpp files

/home/2G03/mandel/Makefile

```
mandel: MandelMain.o MandelCount.o
        c++ -o mandel MandelMain.o MandelCount.o
  -ltrapfpe -lpgplot -lcpgplot -lX11

MandelMain.o: MandelMain.cpp Mandel.h
        c++ -c MandelMain.cpp

MandelCount.o: MandelCount.cpp Mandel.h
        c++ -c MandelCount.cpp
```

Link depends on .o files
creates runnable program

Compile depends on .cpp, .h
creates MandelMain.o file

Compile depends .cpp, .h
creates MandelCount.o file

Remember how Makefile rules work:

Thing_to_make: things_it_needs

<TAB> How to make it

If make does not see something it needs it looks for a rule to make that thing further down

Makefile for 2 .cpp files

/home/2G03/mandel/Makefile

```
mandel: MandelMain.o MandelCount.o
        c++ -o mandel MandelMain.o MandelCount.o
  -ltrapfpe -lpgplot -lcpgplot -lX11

MandelMain.o: MandelMain.cpp Mandel.h
        c++ -c MandelMain.cpp

MandelCount.o: MandelCount.cpp Mandel.h
        c++ -c MandelCount.cpp
```

```
[wadsley@phys-ugrad ~/mandel]$ make mandel
c++ -c MandelMain.cpp
c++ -c MandelCount.cpp
c++ -o mandel MandelMain.o MandelCount.o -
  ltrapfpe -lpgplot -lcpgplot -lX11
[wadsley@phys-ugrad ~/mandel]$
```

Compile makes MandelMain.o

Compile makes MandelCount.o

Link makes **mandel** program

Linking a complicated program

/home/2G03/mandel/Makefile

Link command

```
mandel: MandelMain.o MandelCount.o
        c++ -o mandel MandelMain.o MandelCount.o
   -ltrapfpe -lpgplot -lcpgplot -lX11

MandelMain.o: MandelMain.cpp Mandel.h
        c++ -c MandelMain.cpp

MandelCount.o: MandelCount.cpp Mandel.h
        c++ -c MandelCount.cpp
```

The **mandel** program needs:

- iCountIterations (from MandelCount.o, .cpp)
- main function (from MandelMain.o, .cpp)
- pgplot functions (from –lpgplot -lcpgplot)
- X Windows function (needed for pgplot, from –lX11)
- floating point exception (from –ltrapfpe)

The link step must provide everything in this list

Makefile variables

Set Variable: CPPFLAGS

/home/2G03/mandel/Makefile

```
CPPFLAGS = -O3
#CPPFLAGS = -g -O0

mandel: MandelMain.o MandelCount.o
        c++ $(CPPFLAGS) -o mandel MandelMain.o MandelCount.o -ltrapfpe
   -lpgplot -lcpgplot -lX11

MandelMain.o: MandelMain.cpp Mandel.h
        c++ $(CPPFLAGS) -c MandelMain.cpp

MandelCount.o: MandelCount.cpp Mandel.h
        c++ $(CPPFLAGS) -c MandelCount.cpp
```

Comment (ignored)

make supports text variables. You set them with
NAME =
and paste them into the code with
$(NAME)

Makefile variables

Set Variable: CPPFLAGS

/home/2G03/mandel/Makefile

```
CPPFLAGS = -O3
#CPPFLAGS = -g -O0

mandel: MandelMain.o MandelCount.o
        c++ $(CPPFLAGS) -o mandel MandelMain.o MandelCount.o -ltrapfpe
   -lpgplot -lcpgplot -lX11

MandelMain.o: MandelMain.cpp Mandel.h
        c++ $(CPPFLAGS) -c MandelMain.cpp

MandelCount.o: MandelCount.cpp Mandel.h
        c++ $(CPPFLAGS) -c MandelCount.cpp
```

Comment (ignored)

CPPFLAGS = -O3 Fast (optimized) compile

CPPFLAGS = -g –O0 Debugging compile

Comment out the one you don't want. Note you may need to touch the file to compile again.

Makefile: debug compile

/home/2G03/mandel/Makefile edited

```
#CPPFLAGS = -O3
CPPFLAGS = -g -O0

mandel: MandelMain.o MandelCount.o
	c++ $(CPPFLAGS) -o mandel MandelMain.o MandelCount.o -ltrapfpe -
   lpgplot -lcpgplot -lX11

MandelMain.o: MandelMain.cpp Mandel.h
	c++ $(CPPFLAGS) -c MandelMain.cpp

MandelCount.o: MandelCount.cpp Mandel.h
	c++ $(CPPFLAGS) -c MandelCount.cpp
```

```
[wadsley@phys-ugrad ~/mandel]$ gedit Makefile &
[wadsley@phys-ugrad ~/mandel]$ make mandel
make: `mandel' is up to date.
[wadsley@phys-ugrad ~/mandel]$ touch *.cpp
[wadsley@phys-ugrad ~/mandel]$ make mandel
c++ -g -O0 -c MandelMain.cpp
c++ -g -O0 -c MandelCount.cpp
c++ -g -O0 -o mandel MandelMain.o MandelCount.o -ltrapfpe -
   lpgplot -lcpgplot -lX11
[wadsley@phys-ugrad ~/mandel]$
```