

Computational Physics

Development Tools

02/12/2009

Outline

1 The parameters *argc* and *argv*

2 Development Tools

- Makefiles
- Debugging Software

3 Project 5

The main Function

```
# include <iostream.h>
```

```
int main (int argc, char *argv[]) {
```

```
    ...
```

```
}
```

① The parameters *argc* and *argv* respectively give the number and value of the program's command-line arguments.

② Example: program 2 3

```
argc = 3
```

```
argv[1] = "2"
```

```
argv[2] = "3"
```

Outline

1 The parameters *argc* and *argv*

2 Development Tools
Makefiles
Debugging Software

3 Project 5

Going Beyond One File

So far: Building from a single source file that contains everything:

```
g++ <options> -o program program.cc
```

New: Building from several src files

- Compile by parts (compile just what is needed)
 - `g++ <options> -c src1.cc`
 - `g++ <options> -c src2.cc`
 - `g++ -o program src1.o src2.o ...`

External Includes & Libraries

- Including Include Files

```
g++ <options> -c src1.cc -I<include dir path>
```

Example:

```
g++ -Wall -c example.cc -I/export/home/.../comphy/include
```

External Includes & Libraries

- Including Include Files

```
g++ <options> -c src1.cc -I<include dir path>
```

Example:

```
g++ -Wall -c example.cc -I/export/home/.../comphy/include
```

- Linking with Libraries

```
g++ <options> -o program program.o -L<lib path> -l<lib>
```

Example:

```
g++ -o example example.o -I/export/home/.../comphy/lib -lComphyUtil
```

External Includes & Libraries

- Including **Include Files**

```
g++ <options> -c src1.cc -I<include dir path>
```

- Linking with Libraries

```
g++ <options> -o program program.o -L<lib path> -l<lib>
```

- Creating Libraries

- Create **Object Files**

(ThreeVector.o FourVector.o LorentzTransform.o ...)

- Use archiver to join objects in a library

```
ar r lib<name>.a src1.o src2.o ...
```

```
ranlib lib<name>.a
```


Managing Projects with *make*

Make is a command generator

- Uses a description file (i.e. Makefile) and some general templates to create a series of commands for execution
- Mostly used to sort out dependency relations among program files
- Greatly simplifies compiling and linking

Example:

```
> make sierpin
```

```
gcc sierpin.cc -o sierpin
```

```
> make sierpin
```

```
make: 'sierpin' is up to date
```

The Syntax of Makefiles

◆ Makefile Components

- ◆ Declarations
- ◆ Target
- ◆ Dependents
- ◆ Command Lines
 - ◆ start w/ tabs

◆ What “Make” Will Do For You

- ◆ Check
Dependencies
- ◆ Minimize Rebuilds
- ◆ Provide Macros
Extensions

```
1 # The duffy Makefile
2 #
3 OPTIONS = -Wall -pedantic
4 CPLIBDIR = /home/eugenio/comphy/lib/
5 CPLIB = libComphyUtil.a
6 LIBS = -L$(CPLIBDIR) -I$(CPLIB)
7
8 duffy: main.o duffy_fun.o
9     g++ -o duffy main.o duffy_fun.o
10 $(LIBS)
11 main.o: main.cc
12     g++ $(OPTIONS) -c main.cc
13 duffy_fun.o: duffy_fun.cc
14     g++ $(OPTIONS) -c duffy_fun.cc
15 clean:
16     /bin/rm -f core *.o
```

More Information on MAKE

See



GNU Make

<http://www.gnu.org/software/make/>

What is a Debugger ?

Imagine a program which could *spy* on other programs:

```
comphy % spy MyProgram
```

```
spy > execute "MyProgram" up to line #30
```

```
spy > show me the value of "x"
```

```
spy > execute the next line of code
```

```
spy > show me the current value of "x"
```

Our "spy" is a debug program

→ The Linux program "gdb" is a command line debugger.

Warning !!!

Debuggers can only operate on programs that have been compiled with the debug option “-g”.

Example:

```
g++ -g -o myprogram -c myprogram.cc
```

What is DDD ?

Data Display Debugger

DDD is a graphical front-end for **gdb** and other command-line debuggers. Using DDD, you can see what is going on “inside” another program while it executes – or what another program was doing at the moment it crashed.

For more information see:

<http://www.gnu.org/software/ddd/>

Invoking DDD

`comphy % ddd &`

`comphy % ddd <program> &`

`comphy % ddd <core file> &`

`comphy % ddd <process id> &`

Outline

1 The parameters *argc* and *argv*

2 Development Tools
Makefiles
Debugging Software

3 Project 5

Part I: mysum

Sums of powers of positive integers

$$= 1^p + 2^p + 3^p + \dots + n^p$$

Usage: mysum <integer> <power>

Example: mysum 4 3

$$1^3 + 2^3 + 3^3 + 4^3 = 100$$