

Makefile

Makefile

```
1 # use g++ if no compiler is defined
2 CXX ?= g++
3 # which standard to use in the default case
4 CXXSTD ?= -std=c++11
5
6 # and which flags
7 CXXFLAGS ?= -Wall -O2 -fPIC
8
9 # and which includes
10 CXXINCLUDES ?= -I. -I/usr/local/bin -I/opt/local/include
11
12 # and where to look for the libraries
13 CXXLIBS ?= -L/usr/local/lib -L/opt/local/lib
14
15 # headers which cause a rebuild
16 CXXHEADERS ?= /path/to/frail/header/likely/to/change.hpp
17
18 # install prefix
19 INSTALLPREFIX ?= /usr/local
20
21 compile := $(CXX) $(CXXSTD) $(CXXFLAGS) $(CXXINCLUDES)
22
23 # output all variable values, only useful for debugging
24 varsToPrint := CXX CXXSTD CXXFLAGS CXXINCLUDES CXXLIBS CXXHEADERS
25 $(foreach var,$(varsToPrint),$(info $(var) = $($var)))
26
27 # the first target make encounters is its default target
28 # --> here: data
29 # By default, generate some data
30 data: simulation
31     ./simulation > data
32
33 # link the simulation
34 simulation: libintegrate.a libdynamic.so simulation.o
35     $(compile) simulation.o $(CXXLIBS) -ldynamic -lintegrate
36
37 # Build required libraries:
38 # one static and one dynamic
39 libintegrate.a: integrate.o
40     ar ruc $@ $<
41     ranlib $@
42
43 libdynamic.so: dynamic.o
44     $(compile) -shared -o $@ $<
45
46 # rule how to make any object file blabla.o from its corresponding cpp file blabla.cpp
47 # recompiles if any headers changes
48 %.o: %.cpp $(CXXHEADERS)
49     $(compile) -c $< -o $@
50
51 # The ".PHONY" line makes the clean rule apply even if by chance there is a file named 'clean'
52 .PHONY: clean install
53 clean:
```

```
54     rm *.o
55
56 bindir := $(INSTALLPREFIX)/bin
57 libdir := $(INSTALLPREFIX)/lib
58 incldir := $(INSTALLPREFIX)/include
59 installflags := --compare --backup=existing --suffix=.bak
60 install: simulation
61     install --directory $(bindir) $(incldir) $(libdir)
62     install $(installflags) --strip simulation $(bindir)
63     install $(installflags) libintegrate.a libdynamic.so $(libdir)
64     install $(installflags) $(CXXHEADERS) $(incldir)
65
66 # if you have any problems with your rules not being applied, they could be shadowed by a
67 # builtin rule, so try adding this:
68 .SUFFIXES:
```