C++ für Physiker

29.04.2015

Jan Totz

jantotz@itp.tu-berlin.de

Introduction

Ein Problem ist nicht analytisch lösbar?

A) Ein neues Problem suchen

B) Problem mit cleveren Näherungen vereinfachen

C) Aufgeben

D) Numerische Lösung

Mechanik: Vielteilchen Systeme, Chaos
QM: Schrödinger-Gleichung für spannende Potentiale
Festkörperphysik: Banddiagramme, Spektren
Hydrodynamik, Molekulardynamik
Biologische Systeme
→ Validierung von physikalisch cleveren Näherungen

ReadMe

- Literatur:
  - stackoverflow.com
  - C++ numerical recipes
  - wikibooks: C++

- Kurse
  - PC-Pool: Grundlagen wissenschaftlicher Programmierung (SS)
  - Projektgruppe Praktische Mathematik: C (WS & SS)
  - Astrophysik: Numerikum I (WS)

& More

- Performante Programmiersprache für aufwändiges Problem: C/C++, Fortran
- Scriptsprache: bash, Python
- Auswertung: Mathematica, gnuplot, Matlab, Python-matplotlib, octave, Excel(?)

Was braucht man zum Programmieren?

Linux

1. guten Editor: geany
2. compiler: g++

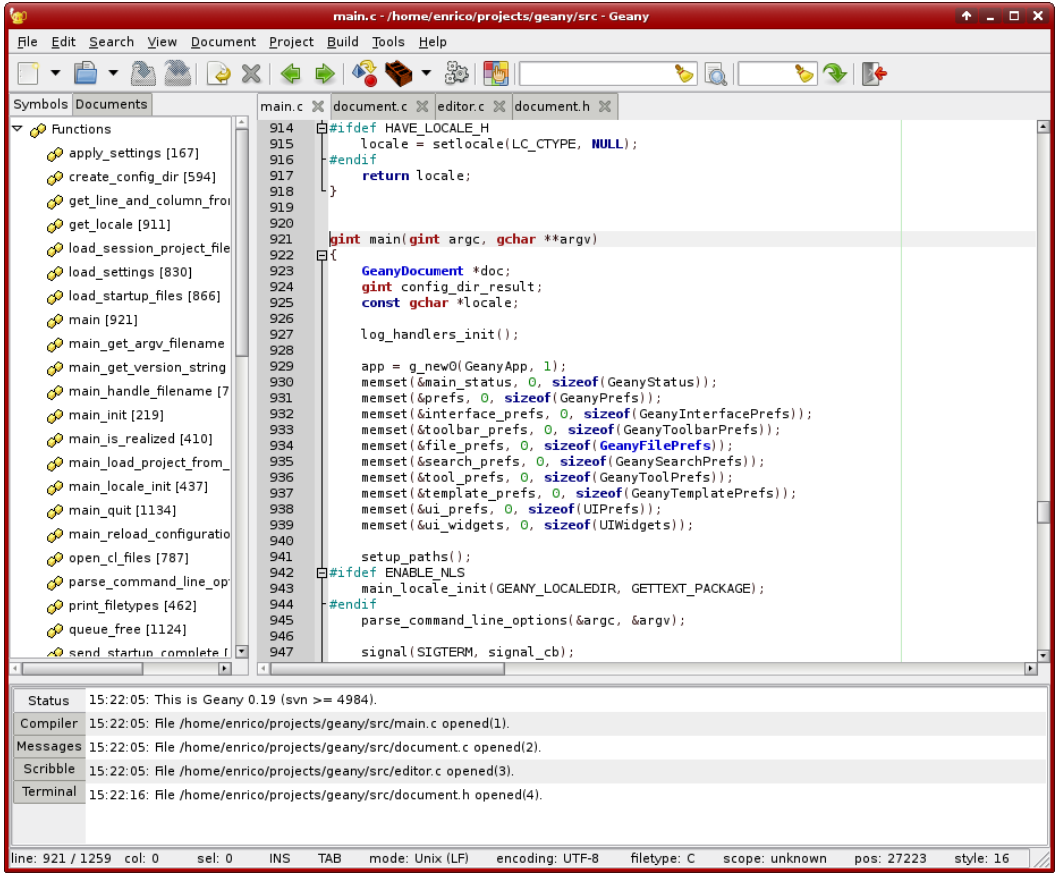
3. Installation:
   sudo apt-get install g++ geany

Windows

1. guten Editor: geany
2. compiler: g++ (mingw)

3. Installation guide:
   http://blog.stijn-dhaese.be/2008/03/how-to-install-mingw-with-geany/

Geany Look


main.c - /home/enrico/projects/geany/src - Geany
File Edit Search View Document Project Build Tools Help
Symbols Documents
Functions
apply_settings [167]
create_config_dir [594]
get_line_and_column_fro
get_locale [911]
load_session_project_file
load_settings [830]
load_startup_files [866]
main [921]
main_get_argv_filename
main_get_version_string
main_handle_filename [7
main_init [219]
main_is_realized [410]
main_load_project_from_
main_locale_init [437]
main_quit [1134]
main_reload_configuratio
open_cl_files [787]
parse_command_line_op
print_filetypes [462]
queue_free [1124]
send_startup_complete [
main.c
document.c
editor.c
document.h
914 #ifdef HAVE_LOCALE_H
915 locale = setlocale(LC_CTYPE, NULL);
916 #endif
917 return locale;
918 }
919
920
921 gint main(gint argc, gchar **argv)
922 {
923 GeanyDocument *doc;
924 gint config_dir_result;
925 const gchar *locale;
926
927 log_handlers_init();
928
929 app = g_new0(GeanyApp, 1);
930 memset(&main_status, 0, sizeof(GeanyStatus));
931 memset(&prefs, 0, sizeof(GeanyPrefs));
932 memset(&interface_prefs, 0, sizeof(GeanyInterfacePrefs));
933 memset(&toolbar_prefs, 0, sizeof(GeanyToolbarPrefs));
934 memset(&file_prefs, 0, sizeof(GeanyFilePrefs));
935 memset(&search_prefs, 0, sizeof(GeanySearchPrefs));
936 memset(&tool_prefs, 0, sizeof(GeanyToolPrefs));
937 memset(&template_prefs, 0, sizeof(GeanyTemplatePrefs));
938 memset(&ui_prefs, 0, sizeof(UIPrefs));
939 memset(&ui_widgets, 0, sizeof(UIWidgets));
940
941 setup_paths();
942 #ifdef ENABLE_NLS
943 main_locale_init(GEANY_LOCALEDIR, GETTEXT_PACKAGE);
944 #endif
945 parse_command_line_options(&argc, &argv);
946
947 signal(SIGTERM, signal_cb);
Status 15:22:05: This is Geany 0.19 (svn >= 4984).
Compiler 15:22:05: File /home/enrico/projects/geany/src/main.c opened(1).
Messages 15:22:05: File /home/enrico/projects/geany/src/document.c opened(2).
Scribble 15:22:05: File /home/enrico/projects/geany/src/editor.c opened(3).
Terminal 15:22:16: File /home/enrico/projects/geany/src/document.h opened(4).
line: 921 / 1259 col: 0 sel: 0 INS TAB mode: Unix (LF) encoding: UTF-8 filetype: C scope: unknown pos: 27223 style: 16

1. C++ Beispielprogramm

hello_world.cpp

```
// comment: here be a description of this program

# include <iostream>
using namespace std;

int main(){
        cout << "Hello World!" << endl;
}
```

compile with:
g++ hello_world.cpp –o hello_world.exe

Real world problem:
Love affair dynamics [1]

problem: Romeo & Julia lieben sich.
Spannender: Romeo liebt Julia.
Aber Julia mag keinen liebestollen Romeo. Sie fühlt sich erst zu ihm hingezogen, wenn er ihr die kalte Schulter zeigt. Sobald sie ihn liebt, steht er auch wieder auf sie.

variables: R – Romeos Liebe (zu Julia)

J- Julias Liebe (zu Romeo)

dynamical system:

$$\dot{R} = aJ$$
$$\dot{J} = -bR$$

[1] Strogatz: Nonlinear Dynamics & Chaos, p. 138

Numerical solver for love

dynamical system: $\dot{x} = f(x)$

finite difference approximation: $\frac{x_{t+1} - x_t}{\Delta t} = f(x)$

rewrite: $x_{t+1} = x_t + \Delta t\, f(x)$

computer code:

```
xneu = x + dt*( f(x) );
```

RungeKutta: Strogatz: Nonlinear Dynamics & Chaos, p. 32

Complete program I

solver.cpp

```
// comment: here be a description of this program
// compile with: g++ -O3 -std=c++0x solver.cpp -o solver.exe

#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
```

Complete program II

```
int main(){
        // parameters
        float dt=0.01;
        int nsteps=10000;
        int savesteps=10;
        int len=nsteps/savesteps;
        vector <float> r_out(len);
        vector <float> j_out(len);
        float rneu, jneu;

        // initial condition
        float r0=1.0;
        float j0=0.0;
```

```
        // model parameters
        float a=1.0;
        float b=1.0;


        // time integration
        float r=r0;
        float j=j0;

        int counter=0;
```

Complete program III

```
for(int step=0; step<nsteps; step++){

        // status update
        if(!(step%100)) cout << "step: " << step << endl;

        // save data for output
        if(!(step%savesteps)){
                r_out[counter] = r;
                j_out[counter] = j;
                counter++;
        }

        // euler step
        rneu = r + dt*( a*j );
        jneu = j + dt*( -b*r );

        // update state
        r=rneu;
        j=jneu;
}
```

Complete program IV

```
        // save data
        savedata(r_out,j_out,len);

}
```

```
void savedata(vector<float> &r_out, vector<float> &j_out, int len){

        ofstream out("/home/jan/Desktop/output.dat");
        for(int i=0; i<len; i++) out << r_out[i] << " " << j_out[i] << endl;

}
```

Mathematica

```
data=Import["/home/jan/Desktop/output.dat"];
Dimensions[data]

plot=ListPlot[data,
        Frame->True,
        FrameLabel->{"Romeo's love R", "Julia's love J"},
        LabelStyle->Directive[20,FontFamily->"Helvetica",Black],
        AspectRatio->1,
        ImageSize->500
];

Export["/home/jan/Desktop/pic.png",plot]
```

gnuplot

call gnuplot from console: „gnuplot“ or use a bash script

```
#! /bin/bash
file="/home/jan/Desktop/output.dat"

#gnuplot call
gnuplot << EOF

# output as png pictures
set term pngcairo enhanced size 640,480 font "Arial"

set out "$HOME/Desktop/pic.png"

# Options
unset key     # no legend
set grid      # grid lines
```

```
# Plotrange
set xrange [-2.0:2.0]
set yrange [-2.0:2.0]

# AxesLabel
set xlabel "Romeo's love R"
set ylabel "Julia's love J"


# Plot
plot "${file}" u 1:2

EOF
```

Common C++ errors

- “;“ am Ende vergessen
- Klammern “{“, “}“ vergessen
- falsche Datentypen: float f =3/5 (0), float f=3.0/5.0 (0.6)
- pointer Fehler bei Array Übergabe/Verwendung in Funktionen -> besser STL vector verwenden
- Index Fehler in Arrays (±1) -> segfaults

Debuggen:

- erwartete Variablenwerte im Programm ausgeben:
  cout << “name: “ << variable << endl;

Outlook

- OpenMP: CPU Parallelisierung
- boost: useful functions: commandline parameters, OS portability, etc
- C++11, 14: new features: auto, lambda expressions, for-each loop, exakte Zeitmessung
- CUDA: High Performace Computing instead of Gaming
- git: version control software – never lose code again
- valgrind: check code for memory errors