Template for C project documentation

Ville Hautamäki

24.9.2002

student id: XXXXXX
email: villeh@cs.joensuu.fi
University of Joensuu
Department of Computer Science
Programming Language C (3 cu) 173001

Contents

1 Description of the problem and solution

You should take into account that this documentation could be used in many different ways. For this reason, this section should be written in clear and straight forward language, without going into too much technical detail. Technical specifications should go to section 3.

This section is organized in such a way, that first in section 1.1 problem is defined and then in section 1.2 solution is explained.

1.1 Description of the problem

In this section you should write a short and clear description of the problem in your own words! *I.e.* you should not copy directly from assignment. If formalism helps in defining the problem, you should use it.

1.2 Description of the solution

In this section you should write a algortihmic explanation of your solution to your assigment. You should not go into too much technical detail, *i.e.* used data structures and pointers etc. Idea is that, when somebody reads this section, he/she can write a program that solves the problem in your assignment in any programming language.

$$F(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) e^{-j2\pi kn/N} \tag{1}$$

Also here you should write all equations your algorithm uses. Also you can use more formal way to describe your algorithm as is done in 1.

Algorithm 1 PNN method

Mark every data vector to be codevector ($m = n$)
repeat
  Find two clusters i and j whose merge minizes the distortion
  Merge the clusters i and j
  $m \leftarrow m - 1$
until $m = c$

2 User guide

User should be able to use your program just be reading this section of your documentation. In larger works, User guide is usually a separate documentation, and so in here also you should treat it as such.

2.1 How to use

Describe here how to use your program. In what format input files are in, how you type in input to your program. Also you should describe here how output is handled, where (file, stdout) program outputs and what output means.

Examples on how to use are also good in here.

Also it should be noted, that more advanced works handle input only in commandline and take large inputs as files. Only in extreme cases should your program ask for input. Reading input should always be buffered (meaning, that scanf() should be used only in extreme cases).

2.2 Special cases

In here your should write how the program handles special cases. And what are the its restrictions.

3 Technical details

As section 1 described algorithmicly how to solve the problem, in this section you describe that solution in the level of the C programming language.

Of main interest here are *Application Programming Interface* (API) and how you use dynamic memory.

3.1 API guide

Meaning of API guide is to present your functions to hypothetical other programmer who is in the future going to use those functions that you wrote in his program. This means that every function has to be (exception of main()) has to be described here. Function protype has to be as heading of each paragraph/ subsection.

int findmax(int *input)

This section should answer to these questions. What does this function do? What does this function take as parameters and what it returns? What are the side effects? What are the limitations of the parameters? What happens when user (that hypothetical programmer) inputs wrong parameters?

3.2 Dynamic memory

In this section you should describe allocation, using, and freeing of dynamic variables.

4 Testing

You should test for correct input, input that are in the border, erroneous input and empty input. This section should have the summary of all those tests.

Appendices

Appendix A Original assignment paper

Appendix B Program sources

Appendix C Testing material

`Just write here all testing data.`