VRIJE
UNIVERSITEIT
BRUSSEL

Proef ingediend met het oog op het behalen van de graad van bachelor in de Ingenieurswetenschappen

Demo document template available on the Rapptorlab website

Course/thesis example

Laurent Segers, Tom van Rijn, Ruben De Smet

May 17, 2017

Jef Paul, Promotor One, Promotor Two, Promotor Three

Ingenieurswetenschappen

Abstract

Hieronder volgt een voorbeeld van tekst zoals gevonden kan worden in de cursus gevorderde programmeertechnieken.

Voorwoord

Met dank aan Laurent Segers, Tom van Rijn en Ruben De Smet.

Contents

1 Inleiding

Hieronder volgt een voorbeeld van tekst zoals gevonden kan worden in de cursus gevorderde programmeertechnieken.

2 Citeren in Latex

Voorbeeld van een citatie: [1]. Andere citaties zijn [2] voor de website van het Rapptorlab waar de template gedownload kan worden en [3] voor de git repository van de titelpagina en slides met de VUB template.

3 Programmeren en compileren in C

Moderne visuele programmeeromgevingen bieden een heel breed assortiment aan diverse opties. In dergelijke omgevingen is het mogelijk om een project aan te maken, nieuwe broncodebestanden aan te maken, code te schrijven en het volledige programma via een klik op een knop te compileren en eventueel uit te voeren. Sommige omgevingen laten zelfs toe om de code in te kleuren (highlighten) volgens verschillende kleurenschema's, de gewenste compiler te kiezen, code te debuggen, opslaan van de broncode in een externe repository, enz. Deze omgevingen waarin alles vanaf het begin tot het uitvoeren van het programma kan verwezenlijkt worden, worden ook *Integrated Development Environment* (IDE) genoemd. Deze omgevingen zijn meteen ook een goed voorbeeld van hoe men grotere programma's ontwikkelt: programma's worden opgedeeld in verschillende deelprogramma's waarbij elk deelprogramma zijn rol vervult. In wat volgt zullen de verschillende tussenstappen toegelicht worden om van de broncode tot een volledig werkend programma te gaan. Deze tussenstappen zullen een beter inzicht geven in hoe dergelijke moderne ontwikkelomgevingen werken. Ook kan het instellen van deze omgevingen met dit inzicht vergemakkelijkt worden.

3.1 Broncode

Code template

Snippet 3.1: Uitbreiding op Hello World.

```
1  // include the stdio.h library
2  #include <stdio.h>
3
4  int main(int argc, char* argv[])
5  {
6     // check if we have argument variables
7     // argv[0] contains the programm name, so go to argv[1] for the variables
8     if (argc>1)
9     {
10       // take the argument variables as if it is an array
11       // the %s indicates where to print the additional variable
12       // (%s = string variable, complete list of options available on internet)
13       printf("Hello %s\r\n",argv[1]);
14    }
15    // return 0 from the main
16    return 0;
17 }
```

3.1.1 Linux terminal

In dit hoofdstuk zullen we een aantal concepten aanhalen waarbij gebruik gemaakt zal worden van een terminal. Een terminal is een venster waarin een aantal commando's kunnen ingevoerd worden. In theorie is het aantal commando's dat de terminal kan aanvaarden eindeloos op voorwaarde dat de nodige programma's hiervoor op de computer aanwezig zijn. Het concept van de terminal wordt voornamelijk nog gebruikt binnen UNIX gebaseerde systemen zoals Linux. Een voorbeeld van een Linux terminal wordt in afbeelding 3.1 weergegeven. Een Windows-variant bestaat evenwel.

Figure 3.1: Terminal in Linux Mint.

3.2 De string als een array of char

Tekenreeksen of strings vormen de basis voor input-output mogelijkheden tussen de gebruiker en de computer. Hiervoor is reeds een brede waaier aan bibliotheken (*eng.* libraries) beschikbaar. Om deze libraries echter te begrijpen is enige kennis over een string vereist. Een string (in C) wordt als volgt gedefinieerd:

- Een string bevat een willekeurig aantal karakters.
- Elk afzonderlijk karakter is van het type *char*. De string is dus een array van karakters.
- Een string wordt ook gekenmerkt door de lengte van de tekenreeks. Omdat de string een char-array is, moet de lengte vervat zijn binnen de tekenreeks. Bij conventie wordt het laatste karakter van de reeks door een '0' aangeduidt (of '\0');

Een string kan in C op verschillende manieren aangemaakt worden. In codefragment 3.2 zijn enkele voorbeelden weergegeven.

Snippet 3.2: Declareren en invullen van een string.

```
1 // method 1
2 char str1[] = "tekenreeks";
```

```
// method 2
char str2[16] = "tekenreeks";
// method 3
char str3[16] = {'t','e','k','e','n','r','e','e','k','s',0,0,0,0,0,0};
```

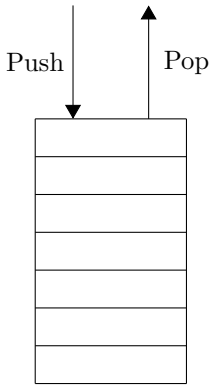
Hoewel de 3 werkwijzen een verschillende syntax hebben, is het eindresultaat hetzelfde. Merk op dat de eerste 2 methoden impliciet de nodige null-karakters in de array opslaan. Dit wordt door de compiler voor ons afgehandeld. Bij conventie gaat men ervan uit dat de string in de array past, en de overige array elementen altijd met een 0 worden aangevuld tot het einde van de array. Om de lengte van de string te kennen, is het dus voldoende om te alle tekens te tellen totdat men een nul-teken tegenkomt (codefragment 3.3). Kan deze methode ook toegepast worden op arrays van andere datatypes? Waarom (niet)? Kan deze ook toegepast worden op een array van char dat niet bedoeld is als string? Waarom (niet)? Bespreek.

Snippet 3.3: Nul-teken in een string.

```
// assign nul sign in a string, method 1
char str3[16] = {'t','e','k','e','n','r','e','e','k','s',0,0,0,0,0,0};
// assign nul sign in a string, method 2
char str3[16] =
    {'t','e','k','e','n','r','e','e','k','s','\0','\0','\0','\0','\0','\0'};
```

3.3 Opgaven

Figure 3.2: Schematische voorstelling van de stack.



$$\begin{cases} l_a = x \cdot l_{a-1} \\ \alpha_a = \alpha_{a-1} + \theta \\ \alpha_b = \alpha_{a-1} - \theta \end{cases} \tag{3.1}$$

$$\begin{cases} \overrightarrow{P_{a,\alpha}} = \overrightarrow{P_{a-1}} + l_a \cdot rot(\alpha_a) \cdot \overrightarrow{v} \\ \overrightarrow{P_{a,\beta}} = \overrightarrow{P_{a-1}} + l_a \cdot rot(\alpha_b) \cdot \overrightarrow{v} \end{cases} \tag{3.2}$$

Bibliography

[1] M. Shell. (2015). *IEEEtran homepage*, [Online]. Available: http://www.michaelshell.org/tex/ieeetran/ (cit. on p. 5).

[2] Rapptorlab. (May 2017). *Rapptorlab website*, [Online]. Available: https://rapptor.vub.ac.be (cit. on p. 5).

[3] R. D. Smet. (May 2017). *Git repo of for the title page of the vub*, [Online]. Available: https://gitlab.com/rubdos/texlive-vub (cit. on p. 5).