

RESTRUCTUREDTEXT INTEGRATOR OF TEACHING MATERIAL

Dr Petar Marić¹; Dr Srđan Popov²

¹ Faculty of Technical Science, University of Novi Sad, Novi Sad, Serbia, petarmaric@uns.ac.rs

² Faculty of Technical Science, University of Novi Sad, Novi Sad, Serbia, srdjanpopov@uns.ac.rs

Abstract—*The integration of teaching materials and transformations into presentation forms is a problem that we face every day. We noticed that the material made for the teaching units, we often have to transform into a suitable presentation form. Preservation, correction and updating of materials, when the material in the presentation format is not suitable. Therefore, the content is generated and updated in the text format, and reStructuredText markups are responsible for generating different presentation forms.*

Keywords—*reStructuredText, ASCII txt, pdf, hml, data index .*

1. INTRODUCTION

reStructuredText and corresponding tools are very useful for generating teaching materials into different formats which can be: HTML (including Windows HTML Help), LaTeX (for printable PDF version), ePub, Texinfo, manual pages and plain text. Among these formats we can find adequate format in all cases the use of materials, which can be found during the teaching process. One of the most important is to generate a book from teaching materials. All required tasks during lecture preparation are covered and materials for a particular teaching unit are performed within a plain text formatted file.

On the basis of this formed material, materials for the classes are generated. This approach has a particularly interesting ability to generate HTML presentation based on the same source code, that is intended for online usage.

2. RELATED WORKS

In article [1] authors present how free and open source software can be incorporated into geospatial education to promote open and reproducible science. Their motive primary is to generate and share source material for courses. They chose HTML but are also using automated scripts to ensure that everything published on the website is also kept in sync with a publicly hosted code repository and allows to transform in reStructuredText, in the future. In our opinion, the transformation of the material into the reStructuredText should be done beforehand to automate the generation of the HTML.

Miller and Ranum [2] generate an interactive eTextbook that incorporates video, code editing, execution and visualization. They were motivated by drawbacks of the classic, typically static electronic book formats. In addition their eTextbook should be created with an open source authoring system. In their implementation all prose is written in restructuredText, including figures, images, and other necessary visual effects.

3. RESULTS AND DISCUSSION

To build our book we're using the reStructuredText markup language [3] in conjunction with the Sphinxdocumentation framework [4]. Thanks to Sphinx and its builders we're able to build our book in multipleformats, currently supported:

- HTML (web site)
- HTML (single page)
- plain text
- LaTeX
- camera-ready PDF, derived from LaTeX

Sphinx itself supports a lot more output formats, but we found these to be particularly useful. Sphinx implements a pluggable software architecture, making it easy to extend its features by adding new Sphinx builders, or even adapting the existing builders.

With Fabric [5] we're able to automatically compile the entire book (Figure 1) (in all of our supported formats) and upload its HTML version to the book's site [6] with a single command (please note that most of the original program output has been snipped for brevity).

```
petar@zverkan$ pipenv run fab build publish
[pjisp.petarmaric.com] Executing task 'build'
Directory '_build' already exists and will be deleted before continuing

Building lint...
[localhost] local: sphinx-build -M lint . _build
...SNIP...
build succeeded.

Building html...
[localhost] local: sphinx-build -M html . _build
Running Sphinx v1.8.1
loading translations [sr_Latn]... done
making output directory...
loading pickled environment... done
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 47 source files that are out of date
...SNIP...
build succeeded.
The HTML pages are in _build/html.

Building singlehtml...
...SNIP...
build succeeded.
The HTML page is in _build/singlehtml.

Building text...
...SNIP...
build succeeded.
The text files are in _build/text.

Building latex...
[localhost] local: sphinx-build -M latex . _build
```

Figure 1:Compiling the entire book, in all of its supported output formats

Once compiled, the book is accessible either online, through its site [6] as showed on Figure 2 or offline, via a camera-ready PDF like on Figure 3.

Table Of Contents

Zadatak “Krvna slika”

• Primer rešenja

Previous topic

Zadatak “Analiza SMS poruka”

Next topic

Zadatak “Indeks telesne mase (BMI)”

Quick search

Go

Zadatak “Krvna slika”

Iz zadate ulazne datoteke učitati statički niz struktura, pri čemu se struktura pacijent_st sastoji od sledećih polja:

- Ime (jedna reč, do 20 karaktera)
- Prezime (jedna reč, do 20 karaktera)
- Holesterol (realan broj)
- Krvni pritisak (prirodan broj)
- Šećer u krvi (realan broj)

Na osnovu zadatog tipa analize tip_analize (holesterol, pritisak ili secer) i sledećih pravila:

Tip analize	Prihvatljiv opseg
holesterol	holesterol < 5.2
pritisak	90 ≤ pritisak < 120
secer	3.9 ≤ secer < 5.6

formirati novi statički niz struktura i upisati ga u zadatu izlaznu datoteku, pri čemu se struktura analiza_st sastoji od sledećih polja:

- Upozorenje (1 znak):
 - = '+' ako je vrednost van prihvatljivog opsega
 - = '-' ako je vrednost u okviru prihvatljivog opsega
- Ime (koristiš funkciju strcpy prilikom kopiranja iz strukture pacijent_st)
- Prezime (koristiš funkciju strcpy prilikom kopiranja iz strukture pacijent_st)

Primer poziva:

```
./proveri pritisak pacijenti.txt izvestaj.txt
```

sa tip_analize=pritisak, zadatim ulazom u datoteci pacijenti.txt:

```
Homer Simpson 13.5 178 7.8
Skinny Pete 4.3 85 3.6
Johnny Bravo 4.8 90 5.4
Pizza Lover 5.2 120 3.9
```

i očekivanim izlazom u datoteci izvestaj.txt:

```
+ Homer Simpson
+ Skinny Pete
- Johnny Bravo
+ Pizza Lover
```

Primer rešenja

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_NIZ 30
#define MAX_IME 20
```

Figure 2: Bookwebsite [6]

FISP zbirka zadataka

Kategorizacija indeksa telesne mase (BMI) = $\frac{maza}{visina^2}$

Indeks	Dijagnoza
BMI < 18.5	Izdrapanost
18.5 < BMI < 25.0	Idealna težina
25.0 ≤ BMI < 30.0	Prekomerna težina
BMI > 30.0	Ležavost

Formirati novi statički niz struktura i upisati ga u zadatu izlaznu datoteku, pri čemu se struktura analiza_st sastoji od sledećih polja:

- Ime (koristiš funkciju strcpy prilikom kopiranja iz strukture pacijent_st)
- Prezime (koristiš funkciju strcpy prilikom kopiranja iz strukture pacijent_st)
- BMI (zaokružen na 2 decimale)
- Dijagnoza

Primer poziva:

```
./bmi pacijenti.txt izvestaj.txt
```

sa zadatim ulazom u datoteci pacijenti.txt:

```
Homer Simpson 13.5 178 7.8
Skinny Pete 4.3 85 3.6
Johnny Bravo 4.8 90 5.4
Pizza Lover 5.2 120 3.9
```

i očekivanim izlazom u datoteci izvestaj.txt:

```
Homer Simpson 44.78 Izdrapanost
Skinny Pete 18.58 Prekomerna težina
Johnny Bravo 19.13 Idealna težina
Pizza Lover 25.00 Prekomerna težina
```

24.1 Primer rešenja

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_NIZ 30
#define MAX_IME 20
#define MAX_PREZIME 20

struct pacijent_st {
    char ime[MAX_IME];
    char prezime[MAX_PREZIME];
    unsigned maza;
    double visina;
};
```

30 Glava 2. Primeri zadataka za T34

FISP zbirka zadataka

```
};

struct analiza_st {
    char ime[MAX_IME];
    char prezime[MAX_PREZIME];
    double bmi;
    char dijagnoza[MAX_DIJAGNOZA];
};

FILE *maza_fajler(char filename[], char mode[], int error_code);
void unesi_pacijent(FILE *fajl, struct pacijent_st pacijent[], int n);
double obradi_bmi(struct pacijent_st pacijent[], int n);
void transformiraj(struct pacijent_st pacijent[], struct analiza_st analiza[], int n);
void unesi_analize(FILE *fajl, struct analiza_st analiza[], int n);

int main(int argc, char *argv[]) {
    if (argc <= 1) {
        fprintf(stderr, "Upotreba: ./fispbmi <arg1> <arg2>\n");
        return 1;
    }

    char *in_filename = argv[1];
    char *out_filename = argv[2];

    FILE *in = fopen(in_filename, "r");
    FILE *out = fopen(out_filename, "w");

    struct pacijent_st pacijent[MAX_NIZ];
    struct analiza_st analiza[MAX_ANALIZA];
    int n;

    unesi_pacijent(in, pacijent, &n);
    transformiraj(pacijent, analiza, &n);
    unesi_analize(out, analiza, &n);

    fclose(in);
    fclose(out);

    return 0;
}

FILE *maza_fajler(char filename[], char mode[], int error_code) {
    FILE *fp = fopen(filename, mode);
    if (!fp) {
        fprintf(stderr, "Greska: %s\n", filename);
        return NULL;
    }
    return fp;
}

void unesi_pacijent(FILE *fajl, struct pacijent_st pacijent[], int n) {
    int i;
    while (fscanf(fajl, "%s %s %u %lf",
        pacijent[i].ime, pacijent[i].prezime,
        &pacijent[i].maza, &pacijent[i].visina) == 4) {
        i++;
    }
}
```

31 2.4. Zadatak “Indeks telesne mase (BMI)”

Figure 3: Camera-ready PDF

One useful feature we found missing in many similar books is a thematic index of programming assignments, which we've created by extending Sphinx (Figure 4).

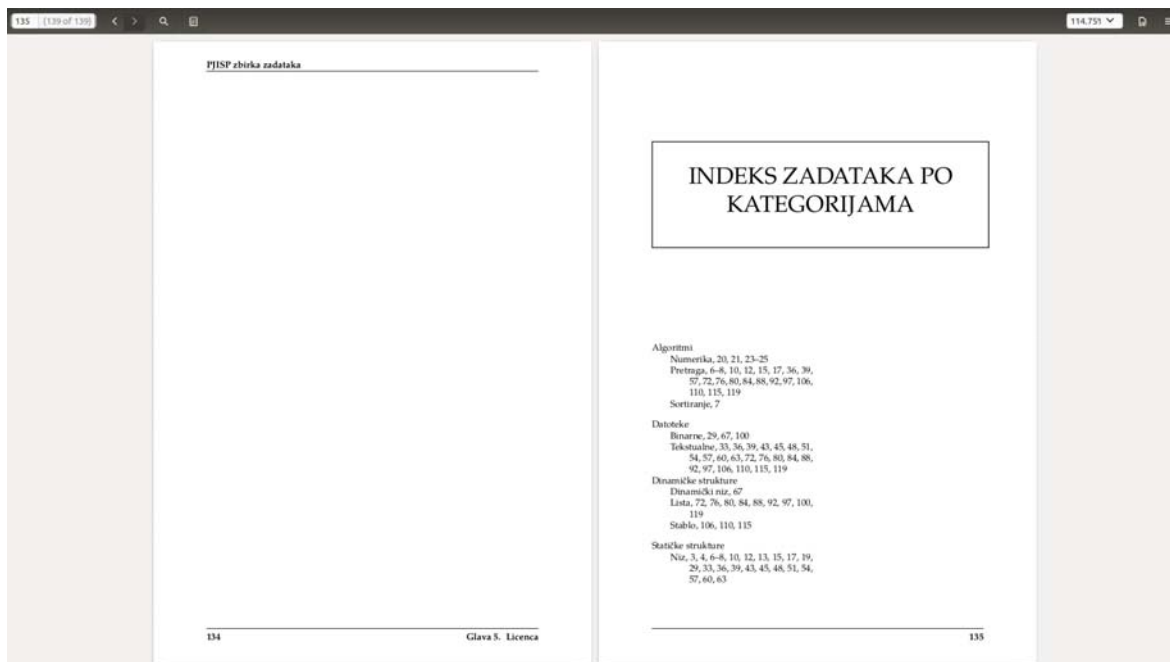


Figure 4: Thematic index

We're already beta-testing the book with our students and in the last school year alone (October 2, 2017 - September 10, 2018) we've had 3,603 users (across all the different devices they use [7]) visit the book's site, where an average user had:

- 4.94 independent sessions with the site
- 3.78 pages per session
- 5:08 minutes session duration

This demonstrates substantial engagement from both our students and the external users, as we taught 4 different courses in the last school year - totaling at about 800 students who we've told about the book.

4. CONCLUSIONS AND FUTURE WORK

We're currently in the process of evaluating different eBook formats and their existing Sphinx builders, aiming to create a platform-neutral paper-free offline version of our book.

In the same time we're integrating the book publishing process into our existing continuous integration infrastructure, which would enable us to automatically publish the book (in all of its formats) whenever a change has been made to its source code repository.

This book and its source code are publicly available, under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License (CC BY-NC-SA 4.0)[8].

REFERENCES

[1] Vaclav Petras , Anna Petrasova , Brendan Harmon , Ross K. Meentemeyer and Helena Mitsova, Integrating Free and Open Source Solutions into Geospatial Science Education, ISPRS INTERNATIONAL JOURNAL OF GEO-INFORMATION, ISSN 2220-9964 Published by MDPI AG, Basel, Switzerland

[2] B.N. Miller and D.L. Ranum. Beyond PDF and ePub: toward an interactive textbook. Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiSE'12), pages 150–155, New York, NY, USA, 2012.ACM.

- [3] D. Goodger, "An Introduction to reStructuredText"
<http://docutils.sourceforge.net/docs/ref/rst/introduction.html>
- [4] G. Brandl, "Sphinx - Python documentation generator" <http://www.sphinx-doc.org/>
- [5] J. Forcier, "Welcome to Fabric!" <http://www.fabfile.org/>
- [6] P. Marić and S. Popov, "PJISP zbirkazadataka" <http://pjisp.petarmaric.com/zbirka-zadataka>
- [7] Google, "How users are identified for users metrics - Analytics Help"
<https://support.google.com/analytics/answer/2992042?hl=en>
- [8] Creative Commons Corporation, "Creative Commons:
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)"
<https://creativecommons.org/licenses/by-nc-sa/4.0/>