

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SPRACOVANIE DLHODOBÝCH MERANÍ
VYBRANÝCH CHARAKTERISTÍK INTERNETU
BAKALÁRSKA PRÁCA

2023

PETER TRENČANSKÝ

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SPRACOVANIE DLHODOBÝCH MERANÍ
VYBRANÝCH CHARAKTERISTÍK INTERNETU
BAKALÁRSKA PRÁCA

Študijný program: Informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra informatiky
Školiteľ: Ing. Dušan Bernát, PhD.

Bratislava, 2023
Peter Trenčanský



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Peter Trenčanský
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Spracovanie dlhodobých meraní vybraných charakteristík Internetu
Processing and visualisation of long-term measurements of selected Internet metrics

Anotácia: Analyzujte štruktúru existujúcej databázy, ktorá obsahuje výsledky dlhodobých meraní odozvy rôznych uzlov v sieti Internet (výstupu ping, traceroute a pod.). Analyzujte dostupné nástroje pre spracovanie a vizualizáciu týchto dát. Navrhňte a implementujte systém s webovým rozhraním pre prácu s nameranými dátami, ktorý používateľovi umožní predovšetkým výber, triedenie a filtrovanie (podľa adries a času), ako aj výpočet štatistík a vizualizáciu ich hodnôt, časových priebehov, prípadne geografického rozloženia. Systém musí byť modulárny v tom zmysle, aby bolo možné konkrétne štatistiky a výstupy jednoducho dopĺňať. Riešenie overte na zobrazení časového vývoja priemernej, najmenej a najväčšej nameranej doby odpovede.

Vedúci: Ing. Dušan Bernát, PhD.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: prof. RNDr. Martin Škoviera, PhD.
Dátum zadania: 28.10.2022

Dátum schválenia: 31.10.2022

doc. RNDr. Dana Pardubská, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie: Tu môžete poďakovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Obsah

Úvod	1
1 Možnosti implementácie	3
1.1 Možnosti vývoja frontendového riešenia	3
1.2 Možnosti vývoja backendového riešenia	4
1.2.1 Možnosti implementácie vo vybranom jazyku	4
2 Jadro a členenie práce	5
2.1 Jadro práce podľa smernice	5
2.1.1 Súčasný stav	6
2.1.2 Cieľ práce	6
2.1.3 Metodika práce a metódy skúmania	6
2.1.4 Výsledky práce a diskusia	6
3 LaTeX	7
3.1 Obrázky	7
Záver	9
Príloha A	13
Príloha B	15

Zoznam obrázkov

3.1 Ukážka hry Červík	8
---------------------------------	---

Zoznam tabuliek

3.1	Doba výpočtu a operačná pamäť potrebná na spracovanie vstupu XYZ	8
-----	--	---

Úvod

Cieľom tejto práce je poskytnúť študentom posledného ročníka bakalárskeho štúdia informatiky kostru práce v systéme LaTeX a ukážku užitočných príkazov, ktoré pri písaní práce môžu potrebovať. Začneme stručnou charakteristikou úvodu práce podľa smernice o záverečných prácach [6], ktorú uvádzame ako doslovný citát.

Úvod je prvou komplexnou informáciou o práci, jej celi, obsahu a štruktúre. Úvod sa vzťahuje na spracovanú tému konkrétne, obsahuje stručný a výstižný opis problematiky, charakterizuje stav poznania alebo praxe v oblasti, ktorá je predmetom školského diela a oboznamuje s významom, cieľmi a zámermi školského diela. Autor v úvode zdôrazňuje, prečo je práca dôležitá a prečo sa rozhodol spracovať danú tému. Úvod ako názov kapitoly sa nečísluje a jeho rozsah je spravidla 1 až 2 strany.

V nasledujúcej kapitole nájdete ukážku členenia kapitoly na menšie časti a v kapitole 3 nájdete príkazy na prácu s tabuľkami, obrázkami a matematickými výrazmi. V kapitole ?? uvádzame klasický text Lorem Ipsum a na koniec sa budeme venovať záležitostiam záveru bakalárskej práce.

Kapitola 1

Možnosti implementácie

Každý softvér si musí určiť, pre aké platformu je vyvíjaný. Samozrejme, čím väčšiu škálu platforiem pokryje, tým je dostupnejší pre väčšie množstvo používateľov. Dnes drvivá väčšina používateľov používa istú podmnožinu zo systémov Android, Linux, Windows, iOS a MacOS X. Spoločnou črtou týchto platforiem je, že medzi nimi neexistuje taká dvojica, z ktorej by jedna platforma natívne podporovala programy napísané pre druhú platformu. Vyvíjať aplikáciu päťkrát, pre každú platformu v jej natívnom prostredí, však nie je veľmi efektívne.

Namiesto toho dnes existuje niekoľko riešení umožňujúcich zdieľať časti zdrojového kódu medzi riešeniami pre rôzne systémy. Ako príklad môžeme uviesť frameworky Flutter, React Native alebo MAUI pre .NET.

Ako univerzálne riešenie sa dnes používajú webové aplikácie. Takéto aplikácie majú niekoľko dôležitých výhod, ako napríklad rýchla dostupnosť, žiadna nutnosť inštalácie a jednoduché spustenie na všetkých platformách, na ktorých je dostupný webový prehliadač (napríklad Firefox, Chrome alebo Opera). Webové aplikácie zvyknú mávať dve oddelené časti - frontend, bežiaci vo webovom prehliadači klienta a backend, bežiaci na serveri. Tieto dve časti spolu komunikujú pomocou siete, najčastejšie pomocou protokolov HTTP(S) alebo WebSocket. Úlohou frontendu je poskytovať prívetivé rozhranie s ktorým môže používateľ pracovať, úlohou backendu je zase spracovávať dáta, komunikovať s databázou a poskytovať dáta frontendu. Práve pre takýto model som sa rozhodol.

1.1 Možnosti vývoja frontendového riešenia

Programovanie webových stránok sa dnes nezaobíde bez programovacieho jazyka JavaScript, ktorý používajú webové prehliadače. Alternatívne sa dajú použiť prekladače iných jazykov do WebAssembly (napríklad framework Blazor pre .NET) alebo jazyk TypeScript, ktorý je rozšírením JavaScriptu o typový systém a iné užitočné konštruk-

cie, zároveň je plne kompatibilný s JavaScriptom (je jeho nadmnožinou) a vo webových prehliadačoch sa spúšťa pomocou prekladu do JavaScriptu. Na vývoj webového frontendu je možnosť používať webové frameworky, ktoré umožňujú delenie stránok na moduly, ktoré sa dajú jednotlivo znovupoužívať a paraneťrizovať na rôznych podstránkach aplikácie, čím zjednodušujú spravovanie a čitateľnosť aplikácie. V súčasnej dobe sú najpopulárnejšie frameworky React, Vue a Angular. Kým React a Vue umožňujú prácu v JavaScripte ale voliteľne je možné vyvíjať aj v TypeScript, Angular prácu v TypeScript vyžaduje. Tieto frameworky sú podporované vo všetkých nových verziách moderných prehliadačov. Keďže všetky tieto frameworky fungujú na veľmi podobnom princípe, je veľmi ťažké povedať, ktorý by bola pre moju aplikáciu najvhodnejší. Vzhľadom na to som sa rozhodol pre Angular, v ktorom mám navyše praktických skúseností.

1.2 Možnosti vývoja backendového riešenia

Pre vývoj backendového riešenia je vhodné vybrať taký programovací jazyk, pre ktorý existuje framework umožňujúci vývoj API, ktoré bude podporovať protokol HTTP(s). Takýchto jazykov existuje veľmi veľa, dokonca pre niektoré jazyky existuje viacero rôznych knižníc umožňujúcich výstavbu takýchto API. Ako príklad môžeme uviesť jazyk Python a knižnice FastAPI alebo Flask a mnohé iné, jazyk Java a framework Spring Boot, jazyk C# a framework ASP.NET Core a mnohé iné. Všetky tieto možnosti majú dostatočné nástroje a je ťažké určiť, ktorý z nich je objektívne najvhodnejší, preto som sa rozhodol pre jazyk C# a ASP.NET Core, s ktorým mám najviac skúseností.

1.2.1 Možnosti implementácie vo vybranom jazyku

Kapitola 2

Jadro a členenie práce

V tejto kapitole si povieme niečo o jadre práce a o jej členení. V zdrojovom kóde v súbore `kapitola.tex` nájdete ukážku použitých príkazov LaTeXu potrebných na písanie nadpisov a podnadpisov a číslovaných a nečíslovaných zoznamov.

Text podkapitoly 2.1 je prebratý zo smernice o záverečných prácach [6, článok 5] a popisuje typické členenie jadra práce (text medzi kapitolami Úvod a Záver). Hoci v niektorých študijných odboroch je vyžadované členenie práce na kapitoly uvedené v smernici, v informatike nie je nutné toto členenie dodržiavať a môžete text rozdeliť do kapitol podľa potrieb konkrétnej témy. Aj tak je však potrebné uviesť súčasný stav problematiky a z práce musí byť tiež jasný váš celkový prínos ako aj detaily vašej práce. Tu uvedené podkapitoly sú len na ukážku použitia príslušných príkazov v LaTeXu, vo vašej práci by ste mali spravidla nemali mať podkapitoly s textom iba na pár riadkov.

2.1 Jadro práce podľa smernice

Jadro je hlavná časť školského diela a člení sa na kapitoly, podkapitoly, odseky a pod., ktoré sa vzostupne čísloujú. Členenie jadra školského diela je určené typom školského diela. Vo vedeckých a odborných prácach má jadro spravidla tieto hlavné časti:

- súčasný stav riešenej problematiky doma a v zahraničí,
- cieľ práce,
- metodika práce a metódy skúmania,
- výsledky práce,
- diskusia.

2.1.1 Súčasný stav

V časti súčasný stav riešenej problematiky doma a v zahraničí autor uvádza dostupné informácie a poznatky týkajúce sa danej témy. Zdrojom pre spracovanie sú aktuálne publikované práce domácich a zahraničných autorov. Podiel tejto časti práce má tvoriť približne 30 % práce.

2.1.2 Cieľ práce

Časť cieľ práce školského diela jasne, výstižne a presne charakterizuje predmet riešenia. Súčasťou sú aj rozpracované čiastkové ciele, ktoré podmieňujú dosiahnutie cieľa hlavného.

2.1.3 Metodika práce a metódy skúmania

Časť metodika práce a metódy skúmania spravidla obsahuje:

1. charakteristiku objektu skúmania,
2. pracovné postupy,
3. spôsob získavania údajov a ich zdroje,
4. použité metódy vyhodnotenia a interpretácie výsledkov,
5. štatistické metódy.

2.1.4 Výsledky práce a diskusia

Časti výsledky práce a diskusia sú najvýznamnejšími časťami školského diela. Výsledky (vlastné postoje alebo vlastné riešenia), ku ktorým autor dospel, sa musia logicky usporiadať a pri opisovaní sa musia dostatočne zhodnotiť. Zároveň sa komentujú všetky skutočnosti a poznatky v konfrontácii s výsledkami iných autorov. Výsledky práce a diskusia môžu tvoriť aj jednu samostatnú časť a spoločne tvoria spravidla 30 až 40 % školského diela.

Kapitola 3

Ukážky užitočných príkazov v systéme LaTeX

V tejto kapitole si ukážeme príklady niektorých užitočných príkazov, ako napríklad správne používanie tabuliek a obrázkov, číslovanie matematických výrazov a podobne. Konkrétne príkazy použité v tejto kapitole nájdete v zdrojovom súbore `latex.tex`. Všimnite si, že pre potreby obsahu a hlavičky stránky je v zdrojovom súbore uvedený aj skrátený názov tejto kapitoly. Ďalšie užitočné príkazy nájdete aj v kapitole 2, na ktorú sme sa na tomto mieste odvolali príkazom `\ref`.

3.1 Obrázky

Vašu prácu ilustrujte vhodnými obrázkami. Pri použití programu `pdflatex` je potrebné pripraviť obrázky vo formáte `pdf`, `jpg` alebo `png`. Vektorové obrázky (napr. `eps`, `svg`) je najvhodnejšie skonvertovať do formátu `pdf`, napríklad programom `Inkscape`.

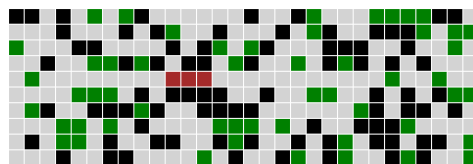
Na vkladanie obrázkov použite prostredie `figure`, ktoré obrázok umiestni na vhodné miesto, väčšinou na vrch alebo spodok stránky a tiež sa stará o automatické číslovanie obrázkov. Na každý obrázok sa treba v hlavnom texte odvolať. Napríklad ilustráciu hry Červík vidíme na obrázku 3.1. Pri odvolávaní sa na číslo obrázku používame príkaz `\ref`. Pri vložení alebo zmazaní obrázku tak nemusíme ručne všetky ostatné obrázky prečíslovať.

Podobne tabuľky vkladajte pomocou prostredia `table`, pričom samotnú tabuľku vytvoríte príkazom `tabular`. Každú tabuľku potom spomeňte aj v hlavnom texte. Napríklad v tabuľke 3.1 vidíme porovnanie časov niekoľkých fiktívnych programov.

V texte môžete tiež potrebovať dlhšie matematické výrazy, ako napríklad tento

$$\sum_{k=0}^n q^k = \frac{q^{n+1} - 1}{q - 1}. \quad (3.1)$$

Použitím prostredia `equation` bol tento výraz zarovnaný na stred na zvláštnom riadku



Obr. 3.1: Ukážka hry Červík. Červík je znázornený červenou farbou, voľné políčka sivou, jedlo zelenou a steny čiernou. Hoci tento popis obrázku je dlhší, v zdrojovom texte je aj kratšia verzia, ktorá sa zobrazí v zozname obrázkov.

Tabuľka 3.1: Doba výpočtu a operačná pamäť potrebná na spracovanie vstupu XYZ. V tomto popise môžeme vysvetliť detaily potrebné pre pochopenie údajov v tabuľke.

Meno programu	Čas (s)	Pamäť (MB)
Môj super program	25.6	120
Speedy 3.1	32.1	100
VeryOld	244.1	200

a očíslovaný. Na toto číslo sa tiež môžeme odvolať príkazom `\ref`. Napríklad rovnica (3.1) predstavuje súčet geometrickej postupnosti.

V práci tiež možno budete uvádzať úryvky kódu v niektorom programovacom jazyku. Môže vám pomôcť prostredie `lstlisting` z balíčka `listings`, v ktorom môžete nastaviť aj jazyk a kód bude krajšie sformátovaný. Ukážku nájdete ako Algoritmus 3.1.

Napokon, v texte nezabudnite citovať použitú literatúru pomocou príkazu `\cite`. Napríklad ďalšie detaily o systéme LaTeX nájdete v knihe od Tobiasa Oetikera a kolektívu [5]. Pre ukážku citujeme aj článok z vedeckého časopisu [3] a článok z konferencie [2], technickú správu [4], knihu [1] a materiál z internetu [6].

Algoritmus 3.1: Algoritmus na výpočet faktoriálu v jazyku C

```

int factorial = 1;
for(int i = 1; i <= n; i++) {
    factorial *= i;
}

```

Záver

Na záver už len odporúčania k samotnej kapitole Záver v bakalárskej práci podľa smernice [6]: „V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom. Rozsah záveru je minimálne dve strany. Záver ako kapitola sa nečísluje.“

Všimnite si správne písanie slovenských úvodzoviek okolo predchádzajúceho citátu, ktoré sme dosiahli príkazom \uv.

V informatických prácach niekedy býva záver kratší ako dve strany, ale stále by to mal byť rozumne dlhý text, v rozsahu aspoň jednej strany. Okrem dosiahnutých cieľov sa zvyknú rozoberať aj otvorené problémy a námety na ďalšiu prácu v oblasti.

Abstrakt, úvod a záver práce obsahujú podobné informácie. Abstrakt je kratší text, ktorý má pomôcť čitateľovi sa rozhodnúť, či vôbec prácu chce čítať. Úvod má umožniť zorientovať sa v práci skôr než ju začne čítať a záver sumarizuje najdôležitejšie veci po tom, ako prácu prečítal, môže sa teda viac zamerať na detaily a využívať pojmy zavedené v práci.

Literatúra

- [1] X. Autor1 and Y. Autor2. *Názov knihy*. Vydavateľstvo, 1900.
- [2] X. Autor1 and Y. Autor2. Názov článku (väčšinou z konferencie). In *Názov zborníka (väčšinou názov konferencie spolu s ročníkom)*, pages 1–100. Vydavateľstvo, 1900.
- [3] X. Autor1 and Y. Autor2. Názov článku z časopisu. *Názov časopisu, ktorý článok uverejnil*, 4(3):1–100, 1900.
- [4] X. Autor1 and Y. Autor2. Názov technickej správy. Technical Report TR123/1999, Inštitút vydávajúci správu, June 1999.
- [5] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *Nie príliš stručný úvod do systému LaTeX2ε*. 2002. Preklad Ján Buša ml. a st.
- [6] Univerzita Komenského v Bratislave. Vnútorňý predpis č. 7/2018, Úplné znenie vnútorného predpisu č. 12/2013 Smernice rektora Univerzity Komenského v Bratislave o základných náležitostiach záverečných prác, rigorózných prác a habilitačných prác, kontrole ich originality, uchovávaní a sprístupňovaní na Univerzite Komenského v Bratislave v znení dodatku č. 1 a dodatku č. 2 smernica rektora Univerzity Komenského v Bratislave o základných náležitostiach záverečných prác, rigorózných prác a habilitačných prác, kontrole ich originality, uchovávaní a sprístupňovaní na Univerzite Komenského v Bratislave, 2013. [Citované 2020-10-19] Dostupné z https://uniba.sk/fileadmin/ruk/legislativa/2018/Vp_2018_07.pdf.

Príloha A: obsah elektronickej prílohy

V elektronickej prílohe priloženej k práci sa nachádza zdrojový kód programu a súbory s výsledkami experimentov. Zdrojový kód je zverejnený aj na stránke <http://mojadresa.com/>.

Ak uznáte za vhodné, môžete tu aj podrobnejšie rozpísať obsah tejto prílohy, prípadne poskytnúť návod na inštaláciu programu. Alternatívou je tieto informácie zahrnúť do samotnej prílohy, alebo ich uviesť na oboch miestach.

Príloha B: Používateľská príručka

V tejto prílohe uvádzame používateľskú príručku k nášmu softvéru. Tu by ďalej pokračoval text príručky. V práci nie je potrebné uvádzať používateľskú príručku, pokiaľ je používanie softvéru intuitívne alebo ak výsledkom práce nie je ucelený softvér určený pre používateľov.

V prílohách môžete uviesť aj ďalšie materiály, ktoré by mohli pôsobiť rušivo v hlavnom texte, ako napríklad rozsiahle tabuľky a podobne. Materiály, ktoré sú príliš dlhé na ich tlač, odovzdajte len v electronickej prílohe.