

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MULTIPARAMETRICKÉ PREHLADÁVANIE S3
CLOUDOVÝCH ÚLOŽÍSK POMOCOU RELAČNEJ
DATABÁZY
BAKALÁRSKA PRÁCA

2024

ONDREJ MAKŠI

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MULTIPARAMETRICKÉ PREHĽADÁVANIE S3
CLOUDOVÝCH ÚLOŽÍSK POMOCOU RELAČNEJ
DATABÁZY

BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná Informatika

Študijný odbor: Aplikovaná Informatika

Školiace pracovisko: Katedra informatiky

Školiteľ: RNDr. Andrej Blaho, PhD.

Bratislava, 2024

Ondrej Makši



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:

Študijný program:

Študijný odbor:

Typ záverečnej práce:

Jazyk záverečnej práce:

Sekundárny jazyk:

Názov:

Anotácia:

Vedúci:

Katedra:

Vedúci katedry:

Dátum zadania:

Dátum schválenia:

garant študijného programu

.....
študent

.....
vedúci práce

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
I Východisková časť	3
1 Objektové úložiská	5
1.1 Popis technológie	5
1.2 AWS S3	6
1.3 Vyhľadávanie v AWS S3	6
2 Použité technológie	9
2.1 Postgres	9
2.2 B+tree	9
2.3 AWS S3 API	10
2.4 React, Node, Express	10
3 Existujúce riešenia	11
3.1 SeaweedFS	11
3.2 GarageS3	12
3.3 s3fs	12
II Vlastné riešenie	13
4 Návrh riešenia	15
5 Implementácia	17
6 Testovanie použiteľnosti	19
7 Jadro a členenie práce	21
7.1 Jadro práce podľa smernice	21
7.1.1 Súčasný stav	22

7.1.2	Cieľ práce	22
7.1.3	Metodika práce a metódy skúmania	22
7.1.4	Výsledky práce a diskusia	22
8	LaTeX	23
8.1	Obrázky	23
	Záver	25
	Príloha A	29
	Príloha B	31

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.

Časť I

Východisková časť

Kapitola 1

Objektové úložiská

Objektové úložisko predstavuje moderný prístup k ukladaniu dát, ktorý sa líši od tradičných súborových a blokových úložísk. Táto technológia je zvlášť vhodná pre prácu s veľkými množstvami neštrukturovaných dát, ako sú multimédiá, dokumenty a zálohy dát. Jej hlavnou výhodou je schopnosť efektívne spravovať dáta vo veľkom rozsahu a poskytovať vysokú mieru škálovateľnosti a dostupnosti.

1.1 Popis technológie

V objektovom úložisku sú dáta uložené ako objekty, pričom každý objekt pozostáva z troch základných častí: dát, metadát a globálneho jedinečného identifikátora. Dáta predstavujú samotný obsah (napríklad textový súbor alebo obrázok), metadáta obsahujú informácie o dátach, ako napríklad typ súboru, vlastník alebo prístupové práva, a identifikátor, ktorý umožňuje jednoznačne identifikovať a adresovať každý objekt v úložisku.

Tento prístup umožňuje, aby boli dáta adresovateľné a prístupné z akéhokoľvek miesta cez HTTP alebo HTTPS protokoly, čo je ideálne pre cloudové aplikácie. Objekty sú nezávislé a nevyžadujú hierarchickú štruktúru súborov, čo eliminuje potrebu správy komplexných súborových systémov a zjednodušuje správu dát. Vďaka tomu je možné dáta efektívne replikovať, migrovať a škálovať.

Výhody objektového úložiska zahŕňajú nielen jeho škálovateľnosť, ale aj odolnosť voči chybám. Systémy často automaticky replikujú dáta na viaceré lokality, čím zaisťujú ich vysokú dostupnosť a ochranu pred stratou dát. Táto funkčnosť robí objektové úložisko ideálnym riešením pre podnikové aplikácie a služby vyžadujúce vysokú dostupnosť dát.

Na trhu existuje množstvo riešení objektového úložiska, od cloudových služieb ako Amazon S3, Google Cloud Storage alebo Microsoft Azure Blob Storage, až po on-premise riešenia, ktoré umožňujú organizáciám udržať si kontrolu nad svojimi dátami

na vlastných serveroch. Každé riešenie ponúka rôzne úrovne konfigurácie a zabezpečenia, aby vyhovelo konkrétnym potrebám a požiadavkám organizácií.

V závere možno povedať, že objektové úložisko prináša nové možnosti v oblasti ukladania a správy dát, ktoré sú zásadné pre moderné dátové centrá a cloudové technológie. Jeho flexibilita, škálovateľnosť a efektívnosť v správe obrovských objemov dát robia z tejto technológie kľúčový nástroj pre organizácie zaoberajúce sa dátovými službami na dnešnom digitálnom trhu.

1.2 AWS S3

Amazon Web Services (AWS) S3, čiže Simple Storage Service, je objektové úložisko poskytované spoločnosťou Amazon, ktoré je navrhnuté pre ukladanie a ochranu akéhokoľvek množstva dát z akéhokoľvek zdroja. S3 je široko využívané v mnohých odvetviach pre jeho spoľahlivosť, škálovateľnosť a nízke náklady na ukladanie dát.

AWS S3 umožňuje používateľom ukladať dáta vo forme objektov v tzv. "buckets" (kbelíkoch), pričom každý objekt je jednoznačne identifikovaný svojím kľúčom. Dáta v S3 môžu byť prístupné z celého sveta pomocou standardných HTTP a HTTPS protokolov, čo umožňuje integráciu s rôznymi aplikáciami a službami.

Služba poskytuje vysokú úroveň odolnosti voči chybám, automaticky replikuje dáta na viaceré geograficky izolované miesta. AWS S3 ponúka tiež rozsiahle možnosti správy dát, vrátane lifecycle managementu pre automatizáciu presunu dát medzi rôznymi druhmi úložiska, ako sú S3 Standard, S3 Intelligent-Tiering, a S3 Glacier, ktoré sú určené pre archiváciu.

S3 je tak ideálnym riešením pre podniky, ktoré potrebujú bezpečné, dostupné a flexibilné úložisko dát, či už pre webové aplikácie, zálohovanie a obnovu dát, kontajnerové úložisko, alebo pre enterprise aplikácie.

1.3 Vyhľadávanie v AWS S3

Amazon Web Services (AWS) S3 (Simple Storage Service) poskytuje základné API operácie pre vyhľadávanie dát v objektovom úložisku. Jednou z hlavných metód na manipuláciu a vyhľadávanie objektov v bucketoch je použitie LIST operácie. Táto operácia umožňuje užívateľom získať zoznam objektov uložených v špecifickom buckete.

Možnosti vyhľadávania s LIST operáciou:

Filtre: Užívatelia môžu použiť rôzne filtre na zúženie výsledkov získaných z LIST operácie. Tieto filtre zahŕňajú napríklad prefixy, ktoré umožňujú zobrazíť iba objekty, ktorých názvy začínajú špecifickým reťazcom. Delimitery môžu byť použité na oddelenie objektov do logicky rozdelených skupín na základe spoločných znakov v názvoch.

Paginácia: Keďže LIST operácia môže vrátiť veľké množstvo objektov, AWS S3 podporuje pagináciu výsledkov. To umožňuje užívateľom získať zoznam objektov v dávkach, čo zjednodušuje spracovanie veľkých dátových súborov a optimalizuje správu zdrojov.

Obmedzenia LIST operácie:

Výkon a omeškanie: LIST operácia môže byť neefektívna a pomalá, najmä v prípade, že bucket obsahuje obrovské množstvo objektov. Vyhľadávanie v buckete bez konkrétnych prefixov alebo pri použití neoptimalizovaných filtrov môže viesť k významnému omeškaniu.

Škálovateľnosť: Keďže S3 je objektové úložisko a nie tradičná databáza, neponúka žiadne natívne indexovanie alebo rýchle vyhľadávanie. To znamená, že vyhľadávanie veľkého počtu objektov môže byť náročné a časovo neefektívne, obzvlášť ak nie sú objekty systematicky organizované.

Náklady na operácie: Aj keď samotná LIST operácia môže byť považovaná za ekonomicky výhodnú, jej neefektívnosť pri spracovaní veľkých objemov dát môže viesť k zvýšeným nákladom, najmä ak sú tieto operácie vykonávané často a v rozsiahlych aplikáciách.

Výsledkom je, že pri použití LIST operácie v AWS S3 je dôležité strategicky plánovať a optimalizovať správu dát, aby sa minimalizovali výkonové obmedzenia a zlepšila efektivita operácií.

Kapitola 2

Použité technológie

Pre túto prácu sú kľúčové nasledujúce technológie: PostgreSQL, B+tree, AWS S3 API, React, Node, Express. V nasledujúcich odsekoch sa bližšie pozrieme na každú z týchto technológií a vysvetlíme, prečo sú dôležité pre túto prácu.

2.1 Postgres

PostgreSQL, často označovaný ako Postgres, je pokročilá open-source objektovo-relačná databázová systém, ktorá podporuje širokú škálu dátových typov a komplexných dotazov. Je obzvlášť obľúbený vďaka svojej robustnosti, vysokému výkonu, a rozsiahlej funkcionalite, vrátane podpory pre transakcie, konkurentné spracovanie, a rozširiteľnosť, čo ho robí ideálnou voľbou pre všeobecné použitie v rôznych aplikáciách od jednoduchých aplikácií až po komplexné systémy s veľkými objemami dát.

2.2 B+tree

B+tree index v PostgreSQL je jedným z najčastejšie používaných typov indexov, ktorý efektívne podporuje rovnosť a rozsahové dotazy na usporiadané dáta. Tento typ indexu je štruktúrovaný ako vyvážený strom, kde každý vnútorný uzol obsahuje kľúče, ktoré slúžia ako rozdeľovacie body pre dátové rozsahy, a listové uzly obsahujú ukazatele (tzv. pointers) na konkrétne záznamy v databáze. Vďaka svojej štruktúre umožňuje B+tree rýchly prístup k dátam, pretože každá operácia vyhľadávania, vloženia alebo odstránenia vyžaduje logaritmický počet krokov vzhľadom na počet prvkov v strome. Tento index je obzvlášť vhodný pre veľké objemy dát a vysokú frekvenciu dotazov, čo z neho robí kľúčový nástroj pre zlepšenie výkonu databázových operácií v PostgreSQL.

2.3 AWS S3 API

AWS S3 API je štandardný rozhraní, ktoré definuje ako interagovať s objektovým úložiskom a je široko prijímané nielen v Amazon Web Services, ale aj mnohými ďalšími poskytovateľmi cloudových služieb. Toto API umožňuje vykonávať operácie ako nahrávanie, sťahovanie a mazanie objektov, ako aj správu prístupových práv a metadát spojených s objektmi. Vďaka jeho širokému prijatiu a podpore, mnohé alternatívne objektové úložiská, napríklad MinIO, Ceph, a DigitalOcean Spaces, implementujú kompatibilitu s AWS S3 API, čo umožňuje používateľom využívať rovnaké nástroje a klienty ako by pracovali priamo s AWS S3. Táto interoperabilita zjednodušuje migráciu a integráciu medzi rôznymi cloudovými prostrediami a poskytuje väčšiu flexibilitu v voľbe poskytovateľa objektového úložiska.

2.4 React, Node, Express

Node.js je open-source, multiplatformové prostredie určené na spúšťanie JavaScriptu na strane servera, umožňujúce vývojárom vytvárať škálovateľné sieťové aplikácie. Express je minimalistický a flexibilný framework pre Node.js, ktorý poskytuje súbor robustných funkcií pre webové a mobilné aplikácie a zjednodušuje vývoj backendov často používaním middleware. Na frontendovej strane, React je deklaratívna, efektívna a flexibilná JavaScript knižnica pre vytváranie používateľských rozhraní, ktorá umožňuje vývojárom efektívne spracovať dáta meniace sa v čase pomocou reaktívnych komponentov. Spolu tieto technológie tvoria mocný základ pre moderné web aplikácie.

Kapitola 3

Existujúce riešenia

Táto kapitola sa venuje už existujúcim riešeniam daného problému. Uvádzam bakalársku prácu, ktorá sa touto témou zaoberala minulý rok a vybrala som jednu slovenskú a jednu zahraničnú webovú aplikáciu, na ktorých je k dispozícii veľké množstvo receptov nielen ako jeden blok textu, ale ako dáta, s ktorými je možné pracovať.

3.1 SeaweedFS

SeaweedFS, ako vysoko škálovateľný distribuovaný súborový systém, ponúka efektívne riešenia pre správu a vyhľadávanie veľkých objemov dát. Jednou z jeho kľúčových vlastností je schopnosť ukladať metadata spolu s dátovými objektami, čo umožňuje pokročilé vyhľadávacie operácie. V praxi to znamená, že každý súbor alebo objekt môže byť označený metadátami, ako sú tagy, dátumy, typ súboru alebo vlastné kľúčové slová. Tieto metadata potom môžu byť indexované a využité pre rýchle vyhľadávanie cez API rozhrania SeaweedFS.

Pri implementácii SeaweedFS na vyhľadávanie, užívatelia môžu využiť ElasticSearch, ktorý možno integrovať pre efektívnejšie full-textové vyhľadávanie. ElasticSearch indexuje obsah uložený v SeaweedFS, čím poskytuje možnosti komplexných vyhľadávacích dotazov a zlepšuje celkovú rýchlosť a presnosť vyhľadávania. Tento prístup je ideálny pre aplikácie, ktoré vyžadujú rýchly prístup k veľkým objemom neštruktúrovaných dát, ako sú multimediálne súbory, dokumenty alebo logy.

Okrem toho, SeaweedFS podporuje škálovateľné vyhľadávanie s použitím Filer-a, ktorý umožňuje užívateľom vyhľadávať súbory podľa názvu alebo cesty v reálnom čase. Filer pracuje ako namespace manager, ktorý spravuje cesty a metadata a môže byť konfigurovaný na prácu s rozličnými backendmi ako Redis alebo Cassandra pre rýchlejšie vyhľadávanie a lepšiu dostupnosť dát. Tento mechanizmus poskytuje flexibilitu a efektívnosť potrebnú pre dynamické a rozsiahle dátové ekosystémy.

3.2 GarageS3

3.3 s3fs

Časť II

Vlastné riešenie

Kapitola 4

Návrh riešenia

Kapitola 5

Implementácia

Kapitola 6

Testovanie použiteľnosti

Kapitola 7

Jadro a členenie práce

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

Text podkapitoly 7.1 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.

7.1 Jadro práce podľa smernice

Jadro je hlavná časť školského diela a člení sa na kapitoly, podkapitoly, odseky a pod., ktoré sa vzostupne číslujú. 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.

7.1.1 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.

7.1.2 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.

7.1.3 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.

7.1.4 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 8

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 7, na ktorú sme sa na tomto mieste odvolali príkazom `\ref`.

8.1 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`.

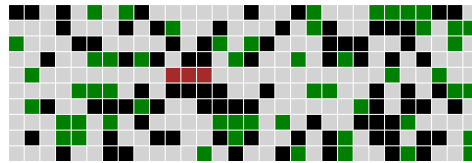
Na vkladanie obrázkov použite prostredie `figure`, ktoré obrázok umiestni na vhodné miesto, väčšinou na vrch alebo spodok stránky a tiež sa stará o automatické číslovanie obrázkov. Na každý obrázok sa treba v hlavnom texte odvolať. Napríklad ilustráciu hry Červík vidíme na obrázku 8.1. Pri odvolávaní 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 8.1 vidíme porovnanie časov niekoľkých fiktívnych programov.

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

$$\sum_{k=0}^n q^k = \frac{q^{n+1} - 1}{q - 1}. \quad (8.1)$$

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



Obr. 8.1: 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 8.1: 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 (8.1) 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 8.1.

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 8.1: 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 infromatických 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 LaTeX2e*. 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.