

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

TVORBA VLASTNÍCH MAP Z PODKLADU OSM – ONLINE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DOMINIKA FABČINOVÁ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

TVORBA VLASTNÍCH MAP Z PODKLADU OSM – ONLINE

OWN OSM MAP GENERATOR APPLIANCE – ONLINE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DOMINIKA FABČINOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ KAŠPÁREK

BRNO 2015

Abstrakt

Práca rozoberá jednotlivé kroky postupu vytvárania máp z OpenStreetMap dát. Skúma nástroje, pomocou ktorých tieto kroky môžu byť realizované. Hodnotí ich varianty s cieľom vybrať vhodnú pre použitie v následnej implementácii. Implementuje celé riešenie pomocou vybraných nástrojov, od získania dát, cez vykreslenie máp, po ich poskytnutie koncovému užívateľovi. Postup dôkladne dokumentuje. Ukazuje, ako do máp vkladať vlastné vrstvy, konkrétne vrstevnice a značené turistické chodníky a ako v mapách vyhľadávať trasy.

Abstract

Thesis analyzes steps of creating maps from OpenStreetMap data. It examines tools, that can be used to perform those steps, evaluates them and their variations with the goal of choosing one suitable to be used in implementation. Then it implements the entire proposed solution using the selected tools from obtaining the data, through rendering map images, to serving them to end users. The procedure is documented thoroughly. It shows how to insert your own layers to the map, in this case counour lines and walking trails and how to search for routes.

Klíčová slova

maps, OpenStreetMap, vrstevnice, routování, klub českých turistů, mapnik, mod_tile, TileMill, OSRM, PostgreSQL

Keywords

maps, OpenStreetMap, contour lines, routing, Czech Tourist Club, mapnik, mod_tile, TileMill, OSRM, PostgreSQL

Citace

Dominika Fabčínová: Tvorba vlastních map z podkladu OSM–online, bakalářská práce, Brno, FIT VUT v Brně, 2015

Tvorba vlastních map z podkladu OSM – online

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracovala samostatne pod vedením Ing. Tomáša Kašpárka.

Uviedla som všetky literárne pramene a publikácie, z ktorých som čerpala.

.....
Dominika Fabčínová
20. května 2015

Poděkování

Na tomto mieste by som rada poďakovala svojmu vedúcemu, Ing. Tomášovi Kašpárkovi za jeho odborné vedenie, čas, ústretovosť a cenné rady a pripomienky. Ďalej by chcela poďakovať svojim rodičom za podporu pri štúdiu a svojmu priateľovi Jánovi Strnadovi tiež za cenné rady a predovšetkým psychickú podporu.

© Dominika Fabčínová, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Úvod	3
1 O projekte OpenStreetMap	4
1.1 Štandardy poskytovania máp	4
1.2 Súradnicové systémy a projekcie	5
2 Analýza	6
2.1 Proces servírovania online máp	6
2.2 Formáty OpenStreetMap dát	6
2.2.1 OSM XML	6
2.2.2 PBF	7
2.2.3 Esri Shapefile	7
2.3 Zdroje OpenStreetMap dát	7
2.3.1 Planet OSM	7
2.3.2 Extrakty	8
2.3.3 Zdroje dát z tretích strán	8
2.4 Databáza	8
2.4.1 PostgreSQL	9
2.5 Import do databázy	9
2.5.1 osm2pgsql	9
2.5.2 Imposm	9
2.6 Renederovanie	10
2.6.1 Mapnik	10
2.6.2 TileMill	10
2.6.3 MapBox Studio	10
2.6.4 Maperitive	10
2.7 Servírovanie a cachovanie	11
2.7.1 Mod_tile	12
2.8 Zobrazenie v prehliadači	12
2.8.1 Leaflet	13
2.8.2 OpenLayers	13
2.9 Routovanie	13
2.9.1 OSRM	13
2.9.2 osm2pgrouting	13
2.10 Map Server	14

3	Implementácia	15
3.1	Hardware a operačný systém	15
3.2	Zdroje dát	17
3.3	Databáza	17
3.3.1	Inštalácia PostgreSQL	17
3.3.2	Optimalizácia výkonu PostgreSQL databázy	18
3.4	Import do databázy	19
3.4.1	osm2pgsql	19
3.5	Renederovanie a poskytovanie	20
3.5.1	Mapnik	20
3.5.2	Inštalácia mod_tile	20
3.5.3	Generovanie stylesheet-u	20
3.5.4	Nastavenie renderovacieho démona	21
3.5.5	Inštalácia Apache	21
3.6	Zobrazenie v prehliadači	21
3.7	Routovanie	22
3.7.1	OSRM backend	22
3.7.2	OSRM frontend	23
3.8	Vrstevnice	23
3.9	Turistické trasy	24
3.10	Pravidelná aktualizácia	25
3.10.1	check_for_updates.sh	25
3.10.2	update_db.sh	26
3.10.3	update_routing.sh	27
4	Záver	28
A	Obsah CD	31

Úvod

Cieľom tejto práce je popísať jednotlivé kroky procesu tvorby vlastných máp z OpenStreetMap dát, preštudovať možnosti, ktorými môžu byť uskutočnené a vybraný postup potom implementovať a dôkladne zdokumentovať pre zjednodušenie riešenia ostatným záujemcom.

OpenStreetMap je projekt, ktorého cieľom je zmapovať celú zemeguľu a získané dáta voľne poskytovať. To znamená, že tak, ako všetko zo sveta open source, aj OpenStreetMap dáta ponúkajú okrem technickej dostupnosti dát a väčšiny potrebného podporného software, aj dostupnosť legálnu - licenciu, ktorá umožňuje užívateľom dáta a zdrojový kód nie len využívať, ale aj prehliadať, upravovať a priamo sa tak podieľať na ich vývoji a údržbe.

Avšak, cesta od dát k vytvoreniu vlastnej mapy je dlhá, rozvetvená a pre laika je náročné sa na nej orientovať. Bežným problémom totiž je, že o mnohých oblastiach existuje neprieberné množstvo informácií, často sú však neusporiadané, neexistujú komplexnejšie návody, či aspoň úvody do problematiky. Na to, ako z OSM dát vytvoriť a poskytovať vlastné mapy tiež existuje mnoho návodov, väčšina z nich sa však zaoberá len jednou časťou, nie celým postupom a počítajú s bližšou znalosťou problému, čo môže byť problém. Situáciu neuľahčuje ani množstvo existujúcich štandardov. Táto práca si kladie za cieľ celý postup nie len skompletizovať, ale aj vysvetliť, prečo je potrebné ten ktorý krok vykonať.

Kapitola 1

O projekte OpenStreetMap

OSM¹ založil v roku 2004 angličan Steve Coast s cieľom vytvoriť voľne dostupné kartografické dáta, pretože v tej dobe existovali síce podrobné mapové dáta, avšak s rôznymi reštrikciami ohľadom ich použitia. Preto sú OSM dáta šírené pod licenciou Open Data Commons Open Database License (ODbL)^{[2][7]} (do 12. septembra 2012 pod licenciou Creative Commons Attribution-ShareAlike 2.0). Znamená to, že ktokoľvek ich môže kopírovať, upravovať a distribuovať, pokiaľ ako zdroj uvedie „©Prispievatelia OpenStreetMap“. Akékoľvek dielo vytvorené s použitím OSM potom musí byť šírené pod rovnakou licenciou. Pred samotným použitím OSM dát, je teda potrebné preštudovať licencie a stránku www.openstreetmap.org/copyright.

Na vytváraní OSM dát a celého OSM ekosystému sa dobrovoľne podieľa komunita užívateľov, napríklad tým, že chodia do terénu a zbierajú GPS dáta, alebo tým, že upravujú dáta podľa aktuálnych lokálnych zmien. Zapojiť sa môže ktokoľvek a stačí malá zmena, napríklad pridať otváracie hodiny obchodu v susedstve. Okrem fyzických osôb prispievajú i firmy a rôzne združenia, napríklad Geofabrik, či Klub českých turistů. Prispievatelia okrem samotných dát vytvárajú aj software a množstvo návodov ako pracovať s OSM.

1.1 Štandardy poskytovania máp

WMS² je štandard spoločnosti OGS³, ktorý definuje rozhranie pre šírenie digitálnych máp vykreslených z priestorových dát v prostredí internetu^[22]. Inak povedané, definuje akým spôsobom by spolu mali komunikovať skript vo webovom prehliadači a server, ktorý mapy poskytuje. Pre server ja záväzné dokázať odpovedať na 2 požiadavky. Prvým je **GetCapabilities** – server odpovie akými dátami disponuje, koľko a za akých podmienok ich dokáže poskytnúť. Druhým je **GetMap** – server pošle dáta a mapa je potom vykreslená až na užívateľskej strane. Tretím nepovinným požiadavkom je **GetFeatureInfo**, ktorý funguje iba na vrstvách mapy, ktoré sú dotazovateľné – server pošle informácie o nejakom prvku na mape, ktoré sa potom zobrazia v bubline, či postrannom paneli. Štandard je však vždy iba akýmsi dopručením, nie je zaručené, že ho budú všetci dodržiavať.

¹Open Street Map

²Web Map Service

³Open Geospatial Consortium

TMS⁴ je iný prístup k problému poskytovania máp. Je to štandard spoločnosti OSGeo⁵. Poskytuje mapy ako dlaždice⁶, ktoré sú vykreslené na serveri a do webového prehliadača užívateľa posielané ako obrázky.

1.2 Súradnicové systémy a projekcie

Poloha jednotlivých elementov na zemeguli je v OSM dátach udávaná vo WGS 84⁷. Je to celosvetovo uznávaný štandard vydaný ministerstvom obrany Spojených štátov amerických. Tento štandard je okrem iného používaný aj v zariadeniach GPS⁸. Bod v tomto systéme je jednoznačne určený svojou zemepisnou dĺžkou, šírkou a výškou[21].

Dôložitou otázkou je, ako zobraziť guľu (i keď Zem nie je presná guľa, jej tvar sa neustále mení a nazývame ho geoid) do 2D zobrazenia. Pri OSM je napoužívanjšou projekciou Merkator, i keď porjekcií existuje neprieberné množstvo. Je pomenovaná podľa flámskeho kartografa zo 16. storočia – Gerardusa Merkatora. Výhodu projekcie Merkator je, že zachováva uhly, čo je žiadúce hlavne pri zobrazení vo vysokom priblížení na úrovni ulíc. Pri zobrazení celej zemegule v tejto projekcii však vzniká značné skreslenie, pri ktorom sa zdá, že Grónsko má podobnú rozlohu ako Afrika (v skutočnosti však zaberá zhruba 7%). Každá krajina má pritom inú oficiálnu projekciu, ktorá záleží hlavne na tom, či je krajina bližšie k pólu, alebo rovníku a na jej rozlohe.

Slippy map je označenie pre zobrazenie mapy na webe tak, ako je to zvykom dnes, to znamená, že mapu je možné posúvať ťahom, či približovať a oddiaľovať.

⁴Tile Map Service

⁵Open Source Geospatial Foundation

⁶Po anglicky tiles.

⁷World Geodetic System 84. Svetový geodetický systém 84

⁸Global Positioning System

Kapitola 2

Analýza

Obsahom tejto kapitoly je analýza jednotlivých krokov tvorby vlastných máp z podkladu OSM a zároveň popis niekoľkých softwarových nástrojov, ktoré tieto kroky umožňujú realizovať.

2.1 Proces servírovania online máp

Existuje viacero možností, ako poskytovať svoje vlastné mapy vytvorené na základe OSM dát. Na začiatku je potrebné zvážiť s akým účelom budeme mapy poskytovať a na základe toho správne zvoliť hardware a softwarové prostriedky.

Pre vykresľovanie webových máp potrebujeme dáta niekde získať (OSM, Geofabrik a iné). Pre ďalšie spracovanie ich musíme uchovávať na serveri a to buď v pôvodnom špecializovanom formáte, alebo v databáze. Pre zobrazenie mapy užívateľovi potrebujeme ďalej z dát vytvoriť obrázky - rastrové, alebo vektorové. Tento krok sa nazýva renderovanie. Potom už ich zostáva iba zobraziť užívateľovi vo webovom prehliadači. Viď diagram 2.2.

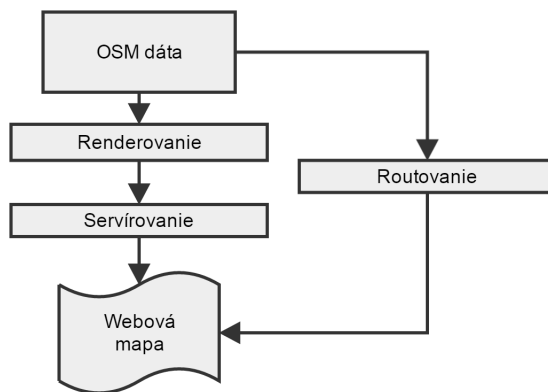
Niektoré nástroje sú určené iba pre jeden z popísaných krokov, iné poskytujú širšiu funkčnosť, napríklad je možné pomocou nich urobiť prvotný import do databázy a potom ju aj aktualizovať, alebo re-renderovať obrázky aj ich servírovať.

2.2 Formáty OpenStreetMap dát

2.2.1 OSM XML

Vektorový súbor vo formáte OSM XML obsahuje 3 základné typy elementov (datových primitív) a to uzly, cesty a relácie[4].

- **Uzly** predstavujú body na mape a sú definované svojou zemepisnou šírkou a dĺžkou v súradnicovom systéme[21]. Môžu popisovať napríklad lavičku, či semafor. Môžu byť súčasťou relácie. Aktuálne je na celej planéte popísaných viac, ako 2 miliardy uzlov.
- **Cesty** sú usporiadané zoznamy uzlov. Delia sa na otvorené, ktoré predstavujú cesty, či rieky a zatvorené, ktoré predstavujú hranice oblastí, napr. budov, mestských častí, či štátov. V prípade uzatvorenej cesty je prvý bod zhodný s posledným.



Obrázek 2.1: Diagram zjednodušeného workflow.

- **Relácie** sú viacúčelové dátové štruktúry, ktoré popisujú vzťahy medzi ostatnými prvkami, tj. uzlami, cestami, prípadne inými reláciami.

2.2.2 PBF

Súbor vo formáte PBF¹ je silne komprimovaný a optimalizovaný binárny formát OSM dát. Je takmer o polovicu menší, než OSM XML, jeho nevýhodou však je, že nie všetky nástroje s ním dokážu pracovať. Spôsob kompresie a ďalšie informácie je možné nájsť na OSM wiki[6].

2.2.3 Esri Shapefile

Shapefile je formát pre ukladanie vektorových priestorových dát². Môže obsahovať línie, body a plochy. Typickým použitím je popis iba jedného typu prvkov, napríklad pobrežných línií, siete železníc, alebo vrstveníc. Skladá sa z viacerých súborov, minimálne troch. Umožňuje editáciu, nemožno ho však aktualizovať pomocou changesetov[13].

Medzi ďalšie formáty patrí O5M, GeoTIFF, či OSM JSON³.

2.3 Zdroje OpenStreetMap dát

Zo zadania práce vyplýva nutnosť použiť dáta pre tvorbu mapy zo zdroja Open Street Map[3], preto v tejto práci nebude vytvorená podrobná analýza ostatných zdrojov, i keď niektoré stoja aspoň za krátku zmienku.

2.3.1 Planet OSM

Hlavný zdroj OSM dát, ktorý sa nachádza na adrese <http://planet.openstreetmap.org/> je týždenne aktualizovanou kópiou celej OSM databázy. Obsahuje dáta celej planéty a týždenné zmenové súbory vo formáte OSM XML a PBF. Ďalej dáta celej planéty vrátane kompletnej histórie zmien a Shapefile so spracovanými obrysmi pobrežia.

¹Protocolbuffer Binary Format

²Po anglicky spatial data

³http://wiki.openstreetmap.org/wiki/OSM_file_formats

2.3.2 Extrakty

Pracovať s celou planétou je veľmi náročné na čas a systémové prostriedky. Našťastie existuje niekoľko zdrojov kde sa dajú získať extrahované mapové dáta podľa potreby. Každý je vhodný na trochu iné účely, avšak však ich využiteľnosť sa prekrýva.

- **Geofabrik.de** je nemecká firma zaoberajúca sa spracovaním máp. Poskytuje denne aktualizované extrakty planéty rozdelené na kontinenty, jednotlivé krajiny, prípadne menšie špecifické regióny. Ponúka ich vo formátoch OSM XML, PBF a Shapefile. Navyše k nim, a to je veľmi dôležité, poskytujú tiež denné a týždenné rozdielové súbory typu OSC⁴ pre jednoduchú aktualizáciu.
- **Openstreetmap.org** poskytuje editačné REST API⁵[12]. Možno využiť pre ručné exporty menších oblastí, kvôli výkonu a možnému obmedzeniu ostatných používateľov, nie je vhodné pre väčšie exporty. API je optimalizované pre získavanie a editáciu jednotlivých prvkov priamo.
- **Overpass API** je webové „databázové“ API optimalizované na čítanie väčšieho množstva dát podľa špecifických kritérií pomocou účelového dotazovacieho jazyka (Overpass XML alebo Overpass QL⁶). Dáta sú neustále aktualizované užívateľmi a k API je k dispozícii klientská knižnica v mnohých programovacích jazykoch.
- **Ďalšie zdroje**⁷
 - **Extrakty Metro** sú extrakty pre hlavné svetové mestá a ich okolie.
 - **Thinkgeo.com** komerčne poskytuje spracované dáta spolu s SDK⁸ pre mobilné a webové aplikácie.
 - **Osm.kyblsoft.cz** poskytuje denne aktualizovanú mapu Českej republiky vo formáte OSM XML a PBF.

2.3.3 Zdroje dát z tretích strán

Zvyčajne poskytované vo formáte Shapefile, alebo vo vlastnom formáte. Napríklad vrstevnice od spoločnosti NASA, ktoré budú použité v kapitole 3 je možné získať vo vlastnom formáte *.hgt a skonvertovať na Shapefile.

2.4 Databáza

Zo zadania práce vyplýva potrebné častejšie aktualizácie máp. Jednou z možností ako to dosiahnuť je v želaných intervaloch stiahnuť všetky dáta a mapy znovu vykresliť. Druhou a ekonomickejšiou alternatívou je uložiť dáta do databázy a v prípade potreby aktualizácie stiahnuť iba zmenový súbor a pomocou neho do databázy importovať zmeny.

Voľba technológie databázy je závislá na objeme dát a spôsobe, ktorým budú dáta následne spracovávané. Pri jej výbere je vhodné zamerať sa hlavne na tieto vlastnosti⁹:

⁴OsmChange

⁵REST - Representational State Transfer. API - Application Programming Interface.

⁶XML - eXtensible Markup Language. QL - Query Language

⁷http://wiki.openstreetmap.org/wiki/Planet.osm#Country_and_area_extracts

⁸SDK - Software Development Kit

⁹http://wiki.openstreetmap.org/wiki/Databases_and_data_access_APIs

- **Aktualizovateľnosť.** Niektoré nástroje (napríklad *osmosis*) dokážu získať súbor typu OSC, ktorý obsahuje zmeny medzi dvoma verziami dát. Nie všetky databázy však umožňujú tieto zmeny importovať. Preto všeobecne, čím väčšiu oblasť Zeme je potrebné udržiavať aktuálnu, tým dôležitejšie je zvoliť databázu, ktorá aktualizácie podporuje. V prípade malého extraktu, napr. jediného mesta môže byť výhodnejšie aktuálne dáta importovať znova namiesto aplikovania zmenového súboru.
- **Podpora geoobjektov** je užitočná najmä pri vykresľovaní máp. Geoobjekty totiž dokážu čítať niektoré programy, ktoré mapy vykresľujú.
- **Bezstratovosť** určuje, či sú v databáze zachované okrem geografických dát aj história zmien a všetky tagy. Pre vykreslenie mapy je možné zvoliť stratový prístup, kompletnú sadu dát potrebujú pre svoju funkcionálnosť editory, alebo niekto, kto chce z dát robiť analýzu.
- **hstore** je datový typ PostgreSQL databáze určený na uchovávanie dvojíc kľúč, hodnota. Používa sa na uchovávanie hodnôt, u ktorých sa predpokladá, že sa k nim bude pristupovať veľmi zriedka. V implementácii bude dôležitý pre vyhľadanie turistických trás.

2.4.1 PostgreSQL

Po vyhodnotení vyššie uvedených vlastností vychádza PostgreSQL ako najvhodnejšia voľba na uskladnenie dát pre účely vykresľovania. Okrem toho je open source a o jej použití v implementácii rozhodlo aj to, že na import do PostgreSQL existuje nástroj *osm2pgsql*, ktorý poskytuje veľa možností a je dobre zdokumentovaný.

2.5 Import do databázy

2.5.1 osm2pgsql

Dokáže importovať OSM dáta vo formátoch OSM, GZ, BZ2, PBF a O5M do PostgreSQL databáze. Dokáže aplikovať aktualizácie pomocou zmenových súborov, čo je nutné k udržiavaniu aktuálnej databázy. Je open source, so zdrojovým kódom zverejneným na serveri GitHub a je dobre zdokumentovaný. Umožňuje import do stĺpcov typu *hstore*. Viac informácií o databázovej schéme do ktorej *osm2pgsql* ukladá dáta je zverejnených na <http://wiki.openstreetmap.org/wiki/Osm2pgsql/schema>.

2.5.2 Imposm

Imposm 2 je taktiež nástroj pre import OSM dát do PostgreSQL databáze, napísaný v jazyku Python. Vytvára vlastnú databázovú schému, s ktorou potom nie je schopný pracovať *mapnik*. *Imposm 3* je vyvíjaný a kompletne reimplementuje verziu 2 v jazyku Go, nie je však ešte vydaný. Údajne by mal byť import pomocou neho rýchlejší a mal by na rozdiel od *Imposm 2* podporovať aj import aktualizácií.

Existujú aj ďalšie nástroje pre import, tie však nie sú primárne určené na vytváranie databáz pre renderovanie.

2.6 Renederovanie

Pod pojmom renderovanie sa rozumie vykresľovanie máp, prípadne dlaždíc (anglicky tiles) z ktorých je mapa poskladaná, do rastrového, alebo vektorového formátu.

2.6.1 Mapnik

Mapnik je univerzálny nástroj pre renderovanie máp. Je implementovaný v jazyku C++ ako knižnica a je predovšetkým zameraný na webový vývoj, ale môže byť využitý aj inak. Vďaka tomu, že je napísaný v C++ s podporou multi-threading-u, je veľmi rýchly. Môže byť použitý pre renderovanie na požiadanie (anglicky on demand), dávkové spracovanie a ponúka tiež rozšírenú funkcionality cez Python API, napríklad úpravu grafického štýlu vykresľovaných dlaždíc.

Na vstupe podporuje Shapefile, PostGIS, GeoTIFF, alebo OSM XML. Okrem toho je potrebné mu dať na vstup XML stylesheet, v ktorom sú definované dotazy do databázy a štýl vykresľovania jednotlivých typov objektov (napríklad hrúbka čiar, farba a font písma, atď.). Pomocou Python skriptu generuje hierarchiu zložiek, do ktorých ukladá vykreselné dlaždice.

V súčasnej dobe je to zrejme najpoužívanější bezplatný open source renderovací nástroj, pretože jeho hlavný konkurent - **Osmarender** - už nie je vyvíjaný a udržiavaný.

V implementácii bude použitý v kombinácii s `mod_tile`, ktorý potom dokáže vyrenderované dlaždice aj servírovať.

2.6.2 TileMill

TileMill je nástroj, ktorý k renderovaniu používa **mapnik**, jeho primárnym zameraním je možnosť pridávať pohodlne v užívateľsky prívetivom grafickom rozhraní vrstvy dát a upravovať štýly. Vrstvy je možné pridávať vo formátoch CSV¹⁰, Esri Shapefile, GeoJSON, KML, GeoTIFF, SQLite a PostGIS. Potom z nich TileMill dokáže vygenerovať štýl (mimo iného) vo formáte XML, ktorý ako vstup pre renderovanie používa **mapnik**. Hlavnou podporovanou projekciou je Mercator. Nie je už však naďalej vyvíjaný, firma MapBox sa totiž zamerala na vývoj svojho komerčného produktu - MapBox Studio. V implementácii bude použitý pre pridanie turistických trás a úpravu ich grafického štýlu.

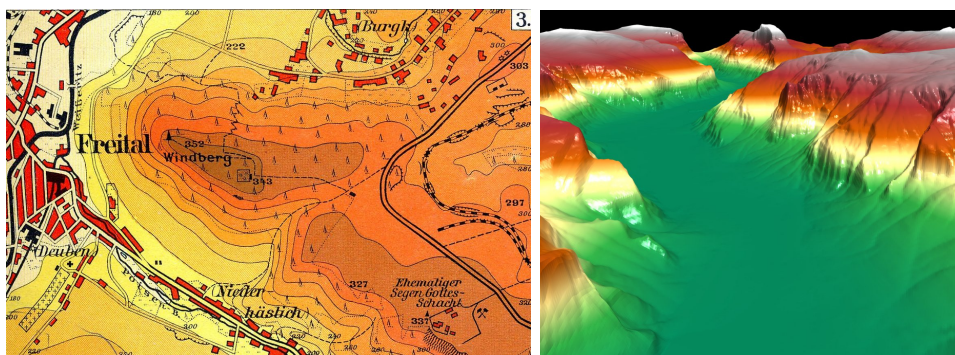
2.6.3 MapBox Studio

Je to komplexný nástroj pre vytváranie a servírovanie vlastných máp, dostupný na Linux, Windows, i MacOS. Je open source so zdrojovým kódom na GitHub-e, poskytuje však aj platenú verziu, ktorá obsahuje infraštruktúru - úložisko dát a zaručenú dostupnosť servera aj pri veľkom počte užívateľov.

2.6.4 Maperitive

Maperitive je desktopová aplikácia, ktorá na vstupe očakáva dáta v súbore OSM XML (prípadne archíve typu .bz2), alebo GPX. Je vhodná na jednorázové renderovanie menších extraktov, napríklad mesta, dokáže však renderovať tieňovanie, vrstevnie, či zafarbenie na základe výškových údajov (anglicky Hypsometric tinting).

¹⁰Comma Separated Values



Obrázek 2.2: Ilustračné obrázky zafarbenia za základe výškových údajov

2.7 Servírovanie a cachovanie

Pod servírovaním sa rozumie poskytovanie vyrenderovaných dlaždíc užívateľovi na základe požiadavku z jeho webového prehliadača. Ideálne podľa štandardu WMS, s použitím Apache, alebo IIS.

Najjednoduchšou variantou sa na prvý pohľad zdá, že stačí vykresliť všetky dlaždice a servírovať ich priamo z adresárovej štruktúry. Tento prístup by síce mohol fungovať pre menšie oblasti, ale pre väčšie je veľmi nevhodný, ako vyplýva z tabuľky.

Zoom	Tile view count	Maximum (4^{zoom})	% viewed on tile
0	1	1	100
1	4	4	100
2	16	16	100
3	64	64	100
4	256	256	100
5	1 024	1 024	100
6	4 096	4 096	100
7	16 384	16 384	100
8	65 536	65 536	100
9	262 144	262 144	100
10	1 048 576	1 048 576	100
11	4 194 304	4 194 304	100
12	13 475 072	16 777 216	80.32
13	35 640 512	67 108 864	53.11
14	87 820 928	268 435 456	32.72
15	163 872 384	1 073 741 824	15.26
16	287 448 064	4 294 967 296	6.69
17	429 535 936	17 179 869 184	2.50
18	617 515 264	68 719 476 736	0.90
Total	1 640 900 565	91 625 968 981	1.79

Tabuľka 2.1: Prevzaté z http://wiki.openstreetmap.org/wiki/Tile_disk_usage. Údaje sú z Marca 2011. Absolútne počty dlaždíc sú počítané pre celú planétu.

Z tabuľky je vidieť, že sa naozaj nejedná o zanedbateľné množstvo dát, ktoré by bolo vykreslené a skladované zbytočne. Iba 1.79 % z celkového množstva možných existujúcich dlaždíc si užívatelia reálne zobrazia, i keď v prípade Českej republiky toto číslo nebude tak nízke, pretože 70% Česka nezaberá oceán. Je teda jasné, že zvoliť správnu kombináciu metód servírovania a cachovania je kľúčové.

2.7.1 Mod_tile

Mod_tile renderuje mapy pomocou knižnice `mapnik` a servíruje ich pomocou Apache¹¹. Kód je rozdelený na 2 časti:

- **Mod_tile** je časť, ktorá sa stará o servírovanie. De facto je to modul rozširujúci Apache, z čoho vznikol i názov, pretože väčšina Apache modulov sa volá `mod.<niečo>`. Prijíma teda požiadavky od koncového užívateľa, ktorý si chce zobrazíť mapu vo svojom webovom prehliadači. Posiela požiadavky back-end časti na dlaždice, ktoré nie sú ešte vyrenderované, alebo ktoré boli zmenené od poslednej aktualizácie.
- **Renderovací back-end**, ktorý prijíma požiadavky od `mod_tile` a renderuje dlaždice pomocou knižnice `mapnik`.
 - **Renderd** je démon, ktorý používa na renderovanie `mapnik`. Má implementovanú prioritnú frontu požiadaviek, kde na prvé miesto radí požiadavky na dlaždice, ktoré ešte nie sú vyrenderované, až potom na expirované dlaždice (tie, ktoré majú byť zmenené od poslednej aktualizácie). Je implementovaný ako viacvláknový proces, čo znamená, že dokáže efektívne využívať procesor na to, na čo je v danej chvíli potrebné[20].
 - **Tirex** dokáže pracovať s viacerými back-end nástrojmi. Vydíjala ho spoločnosť Geofabrik, vývoj už však nie je aktívny.
 - * **Mapnik** používa na samotné renderovanie. Je o ňom bližšie pojednané v kapitole 2.6.1.
 - * **Test** používa iba na overenie správnej konfigurácie, bez nutnosti mať nainštalovaný `mapnik`, či databázu.
 - * **WMS** sám o sebe nerenderuje žiadne dlaždice, používa sa na získavanie dlaždíc z iných WMS serverov.

Mod_tile využíva cachovaciu stratégiu meta dlaždíc. Táto metóda spája typicky 16 dlaždíc do jednej veľkej meta dlaždice o veľkosti 8x8 dlaždíc. To zvyšuje rýchlosť renderovania, znižuje počet potrebných indexov a tým zvyšuje rýchlosť servírovania.

2.8 Zobrazenie v prehliadači

V momente, keď už je celý server pripravený na poskytovanie máp, je potrebné umožniť užívateľom tieto mapy zobrazíť, či už vo webovom prehliadači, alebo mobilných aplikáciách. Ja som sa v tejto práci zamerala iba na jednoduché zobrazenie cez web pomocou HTML stránky s využitím jednej z JavaScript knižníc.

¹¹ Apache HTTP Server. <http://httpd.apache.org/>

2.8.1 Leaflet

Open source JavaScript-ová knižnica vhodná pre zobrazenie máp na webe, i s podporou zobrazenia na mobilných zariadeniach. Ponúka bohaté API s mnohými užívateľskými prvkami a možnosťami, napríklad obmedzenie zobrazenia mapy iba na určitú oblasť, inteligentné prednačítanie dlaždíc u ktorých je predpoklad, že ich užívateľ zobrazí (napr. dlaždice vedľa už zobrazených, alebo v o 1 väčšom zoome). Podporuje štandard WMS, vrstvy, vykresľovanie vlastných polygónov a ciest.

2.8.2 OpenLayers

OpenLayers je moderná JavaScript-ová knižnica s prehľadným API¹² šírená pod BSD licenciou. Ponúka veľké množstvo funkcií od vektorových vrstiev a podpory mobilných zariadení po možnosti prispôsobenia užívateľského rozhrania, alebo úpravu mapy pomocou CSS¹³.

OpenLayers je oproti Leaflet knižnici robustnejšia, avšak väčšinu bežne potrebnej funkcionality implementujú obe.

2.9 Routovanie

Routovanie je hľadanie najkratšej cesty v sieti ciest na mape, alebo širšom slova zmysle v akomkoľvek grafe.

2.9.1 OSRM

OSRM¹⁴ pre vyhľadávanie najkratšej cesty implementuje algoritmus Contraction Hierarchies urýchľujúci vyhľadávací algoritmus. Vďaka urýchleniu Contraction Hierarchies dokáže vypočítať a poskytnúť trasu v priebehu milisekúnd[8]. Na rozdiel od `osm2pgrouting` sa pri výpočte trasy nedotazuje do databázy, ale používa vlastnú predpripravenú štruktúru „indexov“, ktorú je schopný vytvoriť z OSM dát (napr. PBF súboru). Je napísaný v C++. Ako konkrétne vytvoriť routovanie pre rôzne dopravné prostriedky (automobil, bicykel, pešie trasy) bude bližšie popísané v kapitole implementácia. V tejto práci sa zameriavam (s ohľadom na zadanie) na pešie a turistické trasy.

OSRM neumožňuje inkrementálne updaty, je potrebné celú „databázu“ (súbory `*.osrm`, `*.osrm.edges`, `*.osrm.fileIndex`, `*.osrm.geometry`, `*.osrm.hsgr`, `*.osrm.names`, `*.osrm.nodes`, `*.osrm.ramIndex`, `*.osrm.restrictions`, `*.osrm.timestamp`) pripravovať opakovane. Keďže sa ale cesty nemenia zas tak často, nie je to potrebné robiť každý deň. Záleží na uvážení a hardwarových prostriedkoch.

2.9.2 osm2pgrouting

Tento nástroj pracuje s vlastnou PostgreSQL databázou, do ktorej sa dotazuje pri hľadaní trasy. Neumožňuje však importovanie aktualizácií a prvotný import dát umožňuje iba

¹²<http://openlayers.org/en/v3.5.0/apidoc>

¹³<http://openlayers.org/en/v3.5.0/doc/tutorials/introduction.html>

¹⁴Open Source Routing Machine

z OSM XML súboru[9]. Preto som sa rozhodla tento spôsob routovania vo svojej implementácii nepoužiť.

2.10 Map Server

Map server je open source platforma určená na publikovanie interaktívnych mapových aplikácií na web, ktorá plne podporuje WMS štandard. Pochádza už z polovice 90. rokov a je možné ju spustiť na všetkých platformách (Windows, Linux, Mac OS X). Nie je to však plnohodnotný GIS systém[19].

Kapitola 3

Implementácia

Táto kapitola popisuje jeden z možných postupov pre vytvorenie a poskytnutie vlastnej mapy. Postup zobrazený na obrázku 3.1 bude podrobne popísaný a jednotlivé jeho kroky porovnané s prípadnými možnými alternatívami. Odkazy na ukážky výsledkov sú zverejnené na <http://osm.fit.vutbr.cz/xfabci00/>.

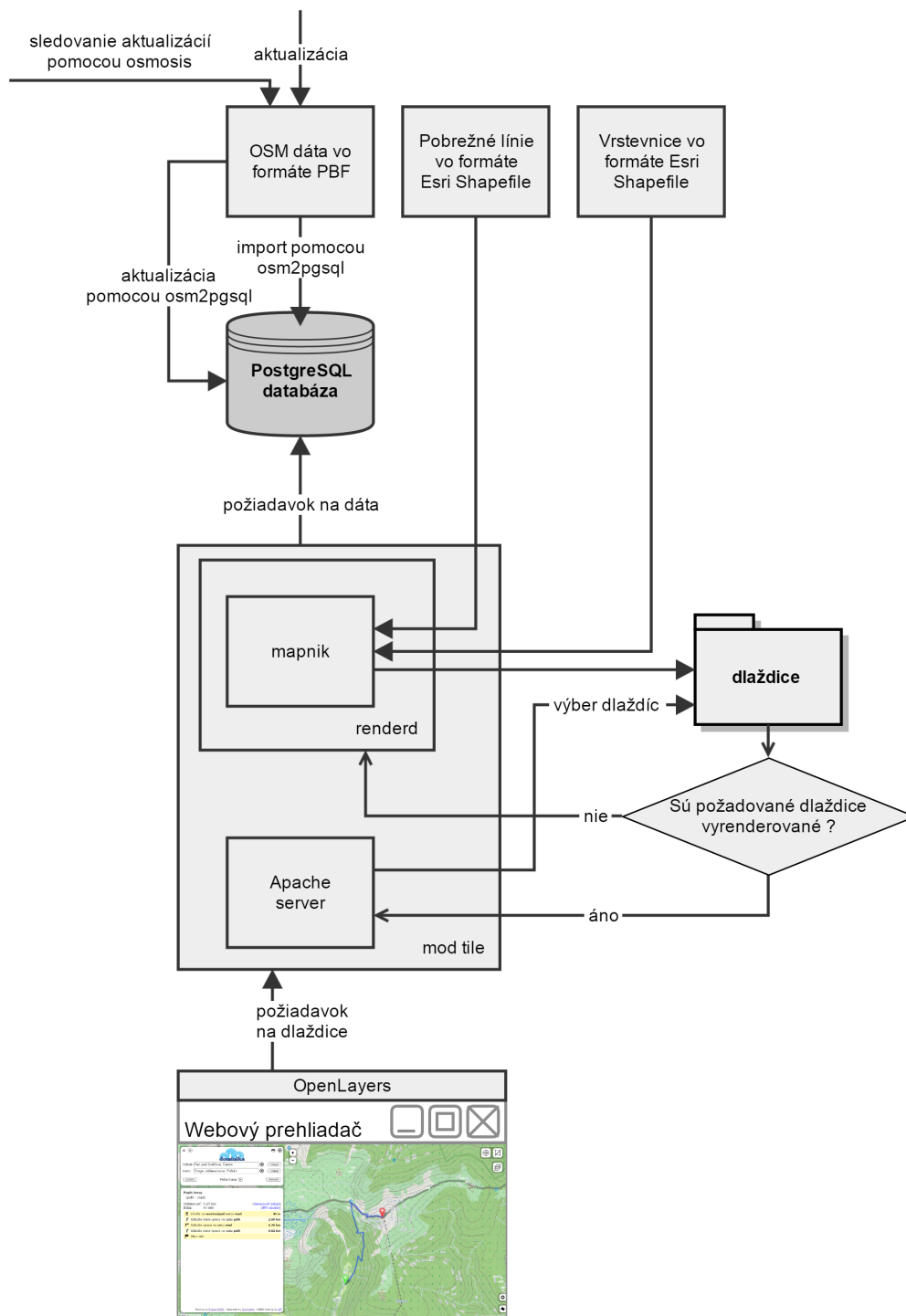
PostgreSQL, osm2pgsql, mapnik, mod_tile a Leaflet, alebo OpenLayers dohromady tvoria funkčný celok, ktorý sa vzájomne dopĺňa, výstup z jedného je možné použiť ako vstup druhého. Oproti nim stojí robustnejší a starší MapServer s inou databázovou schémou, iným spôsobom importu dát do databázy (Imposm), vlastným renderovaním a cachovaním.

3.1 Hardware a operačný systém

Výber správneho hardware je veľmi dôležitým krokom celej implementácie. Je totiž zbytočné mať veľký výkon procesoru, pokiaľ nebude stačiť veľkosť disku na uchovanie všetkých potrebných dát, alebo mať rýchly veľký disk, pokiaľ procesor nebude schopný ho dostatočne vyťažovať. Na čo sa zamerať:

- **Veľkosť a architektúra disku.** Pri OSM je treba počítať s naozaj veľkým objemom dát. Len samotný nekomprimovaný súbor `Planet.osm` má aktuálne 42 GB, týždenný súbor so zmenami 1.1 GB a komprimovaná verzia celého súboru zo formátu PBF potom 28 GB. Ja v tejto implementácii pracujem s komprimovaným PBF súborom obsahujúcim nie celú planétu, ale iba Českú republiku. Tento súbor má 650 MB. Ďalej tento súbor načítam do databázy, ktorá zaberá na disku 18 GB. Potom mám ešte vrstevnice - 1.2 GB a 900 MB, ktoré zaberajú pomocné súbory routovacej služby. Veľkou položkou sú vyrenderované dlaždice, ktoré budú zaberáť viac a viac miesta s pribúdajúcim počtom užívateľov. Len pre zaujímavosť, všetky dlaždice pre celý svet v zoomoch 1 - 18 by zabrali 54 TB diskového miesta[14]. Dôležitým aspektom je i to, že údaje je potrebné nie len uchovávať, ale aj rýchlo čítať. Preto by bolo vhodné mať aspoň na databázu, kde je rýchlosť čítania najkritickejšia, disk s architektúrou SSD¹.
- **Výkon.** Najnáročnejšími operáciami sú prvotný import do databázy, a aktualizácia routovacích dát. I pri ôsmich jadrách zaberajú niekoľko hodín pre Českú republiku. Pre celú planétu bude pravdepodobne veľmi ťažké dostať čas importu pod 6 hodín aj pri 8 jadrách a SSD disku[16].

¹Údaje o veľkosti sú približné, majú slúžiť iba pre získanie predstavy.



Obrázek 3.1: Workflow vytvárania a servírovania máp z OSM dát.

- **Ďalšie faktory.** Je ich nespočetné množstvo a popísať všetky by mohlo zadaním celej ďalšej bakalárskej práce. Pre inšpiráciu ohľadom časov importu do databázy odporúčam <http://wiki.openstreetmap.org/wiki/Osm2pgsql/benchmarks>. Ďalej je potrebné sa zamerať na výber software a jeho nastavenie. Iná kombinácia softwarových nástrojov je vhodná pre veľký import a pravidelnú aktualizáciu pomocou zmenových súborov, iná zas pre jednorázový import jediného mesta.

Úspešnosť služby stojí na jej dostupnosti a preto je rýchlosť servírovania najdôležitejšia. Pokiaľ totiž užívateľ skúsi využiť službu a tá je hneď na prvý pokus nedostupná, alebo veľmi pomalá, pravdepodobne sa už nikdy nevráti.

Keďže fyzicky nevlastním server, rozhodla som sa pre svoju implementáciu použiť virtuálny server v cloude Azure. Jeho výhodou je škálovateľnosť, tj. keď som zistila, že výkon, ktorý som si nastavila na začiatku nestačí, jednoducho som ho cez webové rozhranie pridala. Ako operačný systém som zvolila Ubuntu server 14.10 (Utopic Unicorn). Ubuntu preto, že som s ním už predtým pracovala a mám s ním dobré skúsenosti. Je to veľmi rozšírená a populárna distribúcia OS Linux a tak bolo jednoduché na akýkoľvek problém rýchlo nájsť riešenie. Taktiež väčšina nástrojov s ktorými pracujem je v Debian repozitároch a nebolo teda nutné ich kompilovať zo zdrojových kódov. Mala som možnosť pracovať i na školskom serveri, ale kvôli tomu, že by som nemohla spúšťať príkazy ako root a musela by som kvôli tomu pravidelne kontaktovať admina, by to bolo zdĺhavé a nepraktické zvlášť, keď som si zo začiatku nebola istá, že to, čo robím je správne a potrebné.

3.2 Zdroje dát

Na server je potrebné stiahnuť súbor s dátami, v tomto prípade iba Českej republiky. Použila som súbor vo formáte PBF, pretože je menší ako súbor vo formáte OSM XML a nástroj pre import do databázy `osm2pgsql` s ním dokáže pracovať. Server download.geofabrik.de poskytuje extrakty, to znamená, že nie je potrebné stiahnuť celú planétu a extrahovať ručne dáta pre konkrétnu oblasť. Viac o formátoch a zdrojoch dát v kapitolách 2.2 a 2.3.

```
wget http://download.geofabrik.de/europe/czech-republic-latest.osm.pbf
```

V tejto chvíli je potrebné si poznamenať aspoň deň, v ktorý boli dáta stiahnuté. Bude to dôležitá informácia pri prvej aktualizácii.

3.3 Databáza

Je možné pracovať priamo so súbormi vo formáte OSM XML, PBF a ďalšími, ja som sa ale rozhodla dáta importovať do databázy, pretože potom je možné ich jednoduchšie a s menšími nárokmi na hardware pravidelne aktualizovať. Možnosť pravidelnej aktualizácie je jedným z bodov zadania tejto práce.

3.3.1 Inštalácia PostgreSQL

PostgreSQL je open source, plnohodnotná, spoľahlivá relačná databáza. Dokáže efektívne pracovať i s väčším množstvom dát a rozšírenie PostGIS pridáva podporu pre geografické objekty. O použití PostgreSQL rozhodlo aj to, že na import existuje nástroj `osm2pgsql`,

ktorý poskytuje veľa možností, je dobre zdokumentovaný, aktívne vyvíjaný, open source a dokáže importovať dáta zo súboru vo formáte PBF. Viac o databázach v kapitole 2.4.

Je teda potrebné nainštalovať PostgreSQL databázu. Podrobný návod pre všetky distribúcie je možné nájsť priamo na oficiálnej PostgreSQL Wiki², alebo OSM Wiki³.

Pri inštalácii Postgre by sa mal vytvoriť linuxový užívateľ postgres (možno overiť v /etc/passwd). Potom sa prihláste ako tento užívateľ pomocou príkazu:

```
su - postgres
```

Ďalej je nutné vytvoriť databázu a zároveň jej zmeniť vlastníka, pretože z bezpečnostných dôvodov nie je žiadúce, aby bolo nutné prihlasovať sa pre vykonanie akejkoľvek operácie nad databázou ako užívateľ postgres (databázový root). Na mojom serveri existuje

- užívateľ **dominika**, v návodoch k OSM nástrojom však zvyčajne nájdete **osm**
- databáza **czech**, v návodoch k OSM nástrojom však zvyčajne nájdete **gis**.

Pomenovanie záleží na každého uvážení, pokiaľ ale použijete univerzálnejšie mená užívateľa a databázy, nebudete pri využívaní návodov z internetu musieť toľko vecí upravovať.

Databázu som vytvorila príkazom:

```
createdb czech --owner="dominika"  
psql --dbname czech --command 'CREATE EXTENSION postgis; CREATE  
EXTENSION hstore;'
```

do databázy boli týmto príkazom pridané aj rozšírenia **hstore** a **PostGIS**.

- **Hstore** rozširuje PostgreSQL databázu o datový typ určený na uchovávanie dvojíc kľúč, hodnota. Toto rozšírenie sa používa na uchovávanie hodnôt, u ktorých sa predpokladá, že sa k nim bude pristupovať veľmi zriedka. V implementácii bude využité pre vyhľadanie turistických trás.
- **PostGIS** pridáva podporu geoobjektov.

Potom je potrebné sa odhlásiť z užívateľa postgres príkazom **exit**.

Pretože bude potrebné pripojenie do databázy naozaj často, je pohodlné vytvoriť súbor **.pgpass**[10].

3.3.2 Optimalizácia výkonu PostgreSQL databázy

Pre dosiahnutie optimálneho výkonu databázy je potrebné upraviť hodnoty niektorých premenných v súbore **postgresql.conf**. Pre prvotný import do databázy nie je potrebné sa optimalizáciou výkonu zaoberať úplne podrobne, pretože je to iba jednorázová operácia. Je však vhodné sa tým zaoberať bližšie pre optimalizáciu renderovania, keďže tam je potrebný prístup do databázy neustále. Ja som pre import použila toto jednoduché nastavenie:

²https://wiki.postgresql.org/wiki/Detailed_installation_guides

³<http://wiki.openstreetmap.org/wiki/PostGIS/Installation>

- `shared_buffers` nastavte približne na 25%, maximálne však na 40% celkovej veľkosti pamäte. Akú veľkú máte pamäť zistíte príkazom `free`, alebo `top`.
- `work_mem` nastavte minimálne na 16 MB.
- `maintenance_work_mem` nastavte na $\frac{1}{8}$ celkovej veľkosti pamäte[15][17].

3.4 Import do databázy

V tomto okamihu mám stiahnuté OSM dáta (popísané v sekcii 3.2), importujem ich teda do pripravenej databázy.

3.4.1 osm2pgsql

Importovať dáta pomocou tohto nástroja som sa rozhodla hlavne preto, že dokáže pracovať s komprimovaným PBF súborom obsahujúcim dáta, je open source a stále vyvíjaný, napriek tomu funguje spoľahlivo a stabilne. Je možné ho skompilovať zo zdrojového kódu, podrobný návod na inštaláciu je pripravený priamo v GitHub repozitári[11] na adrese <https://github.com/openstreetmap/osm2pgsql#osm2pgsql>. Pokiaľ nepracujete na vlastnom serveri, musíte požiadať administrátora, aby vám nainštaloval prerekvizity.

Na Ubuntu (a aj ďalších systémoch) je možné ho nainštalovať z repozitára, musíte ale mať možnosť prihlásiť sa ako užívateľ `root`, alebo takzvaný `sudoer`. Pokiaľ nepracujete na vlastnom serveri, musíte požiadať administrátora, aby vám takéto právo pridelil.

Ja som teda na vlastnom serveri zvolila alternatívu:

```
sudo apt-get install osm2pgsql
```

Pred importom preštudujte <https://github.com/openstreetmap/osm2pgsql#usage>[11]. Ja som pre prvotný import spustila `osm2pgsql` s parametrami:

```
osm2pgsql --create --database czech --slim --hstore czech-republic-latest.osm.pbf
```

Použitie povinné a nepovinné parametre:

- `--slim`. Je potrebné ho použiť, bez neho by neskôr nebolo možné databázu aktualizovať.
- `--hstore`. Bez tohto parametra by sa do databázy neimportovali tagy, čo pre základné vykreslenie mapy nevedí, ale tagy obsahujú dôležité informácie (okrem iného) o turistických trasách.
- `--number-processes`. Pokiaľ v tejto chvíli nepotrebuje na serveri spúšťať žiadne iné náročné operácie, nastavte hodnotu na o 1 menšiu, než je váš celkový počet jadier procesoru. Počet jadier zistíte napríklad príkazom `nproc`.
- `--cache`. Nastavte približne na 75% celkovej veľkosti pamäte. Akú veľkú máte pamäť zistíte príkazom `free`, alebo `top`.

- `--cache-strategy`. Použite `osm2pgsql --help --verbose` pre získanie informácií. Ja som používala predvolenú stratégiu cachovania.

Import Českej republiky do databázy s uvedenými parametrami na mojom serveri (Azure A4 Basic tier[18]) s 8 jadrami (každé má 2 GHz), 14 GB pamäte a pevnom disku s rýchlosťou zápisu 25 MB/s (Azure VHD), trvá približne 4 hodiny a 25 minút.

3.5 Renederovanie a poskytovanie

Všetky kroky v tejto sekcii je možné absolvovať podľa podrobného návodu z Debian Wiki na adrese https://wiki.debian.org/OSM/tileserver/jessie#Serving_Tiles, ten je však špecifický pre systémy založené na distribúcii Debian. Za užitočný príkaz považujem utilitu `locate`, ktorá dokáže vyhľadať súbory skrz celú adresárovú štruktúru.

3.5.1 Mapnik

V tejto implementácii bude inštalovaný ako prerekvizita k `mod_tile`, je však možné ho použiť aj samostatne.

3.5.2 Inštalácia `mod_tile`

Je potrebné skompilovať zdrojový kód, ktorý je prístupný na https://github.com/openstreetmap/mod_tile. Táto operácia vyžaduje ako prerekvizity balíčky `autoconf`, `libtool`, `libmapnik-dev` a `apache2-dev`. Pokiaľ sa pokúsíte o inštaláciu podľa oficiálneho návodu⁴ bez nainštalovaných prerekvizít, chybové výpisy v konzole vás navedú k inštalácii chýbajúcich balíčkov.

```
./autogen.sh
./configure
make
sudo make install
sudo make install-mod_tile
```

3.5.3 Generovanie stylesheet-u

Nainštalujte renderer stylesheetu `node-carto` s prerekvizitami, `curl`, `unzip`, `gdal-bin` a `mapnik-utils`. V zložke, ktorá sa jeho inštaláciou vytvorí sa potom nachádza skript `get-shapefiles.sh`, ktorý je potrebné spustiť a ktorý stiahne potrebné súbory vo formáte Shapefile pre mapnik. V súbore `project.mml` je potom potrebné zmeniť meno databázy, pokiaľ sa vaša databáza volá inak, než `gis`. Potom je potrebné spustiť utilitu `carto`, ktorá vygeneruje stylesheet `style.xml`.

```
carto project.mml > style.xml
```

Stylesheet obsahuje informácie pre mapnik o tom, aké má použiť

- dotazy do databázy na získanie všetkých prvkov mapy – ciest, budov, popiskov, atď,

⁴https://github.com/openstreetmap/mod_tile/blob/master/readme.txt

- font písma, jeho farbu a veľkosť, hrúbku ciest, farbu výplne budov, vodných plôch a celkovo všetky informácie o tom, ako má výsledná mapa vyzerieť.

3.5.4 Nastavenie renderovacieho démona

V súbore `renderd.conf` je potrebné upraviť cestu k vygenerovanému súboru `style.xml`, nastaviť pole `host` na `localhost` a zmeniť hodnotu poľa `plugins_dir` na cestu ku knižnici mapnik. Zmeňte tiež hodnoty polí `DAEMON`, `DAEMON_ARGS` a `RUNASUSER` v inicializačnom skripte `/etc/init.d/renderd` podľa návodu, vytvorte zložku, do ktorej sa budú ukladať vyrenderované dlaždice a zmeňte jej vlastníka na rovnakého, aký je v poli `RUNASUSER`.

Je potrebné znovu načítať zmenený konfiguračný súbor.

```
systemctl daemon-reload
```

Potom už zostáva iba renderovacieho démona spustiť.

```
service renderd start
```

3.5.5 Inštalácia Apache

Keďže `mod_tile` je modul Apache HTTP Servera, je potrebné tento nainštalovať a nastaviť tak, aby si propojil modul `mod_tile`. Postupujte podľa návodu na https://wiki.debian.org/OSM/tileserv/jessie#Install_Apache.

3.6 Zobrazenie v prehliadači

V tejto chvíli už je server schopný renderovať a poskytovať dlaždice, musíme však ešte vytvoriť webovú stránku, na ktorej si užívateľ bude môcť mapu zobrazieť. Zvolila som jednoduchú HTML stránku, ktorá používa JavaScript knižnicu OpenLayers. Túto som zvolila preto, že knižnicu Leaflet používa routovací engine OSRM a tak bude ukážka použitia tejto knižnice v sekci zaoberajúcej sa routovaním.

Vytvorila som najjednoduchšiu možnú stránku v HTML a JavaScripte, pretože som v podstate chcela iba vyskúšať, či celé moje riešenie funguje. Táto stránka je v prílohach. Na stránke OpenLayers je dokumentácia⁵ s príkladmi, podľa ktorej to zvládne vytvoriť ktokoľvek so základnou znalosťou jazykov HTML a JavaScript.

V tejto chvíli sme prešli celým procesom, ktorý je potrebný k vykresleniu mapy z OSM dát a jej zobrazeniu vo webovom prehliadači. Budeme do nej ešte pridávať vrstevnice, turistické značky a routovanie. V prípade, že mapové dlaždice v prehliadači nevidíte, skúste zastaviť renderovací démon a namiesto toho spustiť `renderd` v debug móde.

```
/<cesta_k>/renderd --foreground --config /<cesta_k>/renderd.conf
```

⁵http://wiki.openstreetmap.org/wiki/OpenLayers_Simple_Example

3.7 Routovanie

3.7.1 OSRM backend

Pretože neexistuje OSRM balíček, je potrebné si stiahnuť zdrojový kód a skompilovať ho podľa návodu na adrese <https://github.com/Project-OSRM/osrm-backend/wiki/Building%20OSRM>.

Potom si musí OSRM pripraviť svoje súbory. To prebieha v dvoch krokoch:

- **osrm-extract** zo súboru PBF, OSM XML, alebo OSM XML komprimovanom vo formáte BZ2 extrahuje 2 súbory – *.osrm a *.osrm.restrictions. Ktoré dáta do týchto súborov extrahovať je riadené LUA profilom.
- **osrm-prepare** zo súborov .osrm a .osrm.restrictions predpočíta trasy potrebné pre beh routovacieho serveru.

```
/osrm-backend/build/osrm-extract --profile /osrm-backend/profiles
/foot.lua /czech-republic-latest.osm.pbf
/osrm-backend/build/osrm-prepare --profile /osrm-backend/profiles
/foot.lua /czech-republic-latest.osrm
```

Tretím krokom je spustenie samotného routovacieho servera. Stačí spustiť program **osrm-routed** a predať mu cestu k súboru .osrm, za predpokladu že všetky ostatné súbory vygenerované z **osrm-extract** a **osrm-prepare** sú v tej istej zložke.

Ja som sa však rozhodla, že nechcem spúšťať routovací server manuálne po každom štarte operačného systému, preto som vytvorila démona, ktorý sa o spustenie stará automaticky. V zložke /etc/init.d/ sa na každom serveri nachádza súbor skeleton, ktorý sa použije ako šablona. Túto šablónu som skopírovala (v rámci zložky /etc/init.d/), premenovala na **osrm-backend** a upravila tak, aby spúšťala program **osrm-routed**. Tento súbor nájdete v prílohe. Potom je potrebné mu priradiť práva na spustenie príkazom **chmod** a pridať ho do zoznamu démonov, aby sa spustil automaticky po štarte systému.

```
sudo chmod 755 /etc/init.d/osrm-backend
sudo update-rc.d osrm-backend defaults
```

Aktualizáciu na pravidelnej báze som vyriešila skriptom **update_routing.sh**, viac o aktualizáciách v sekcii 3.10.

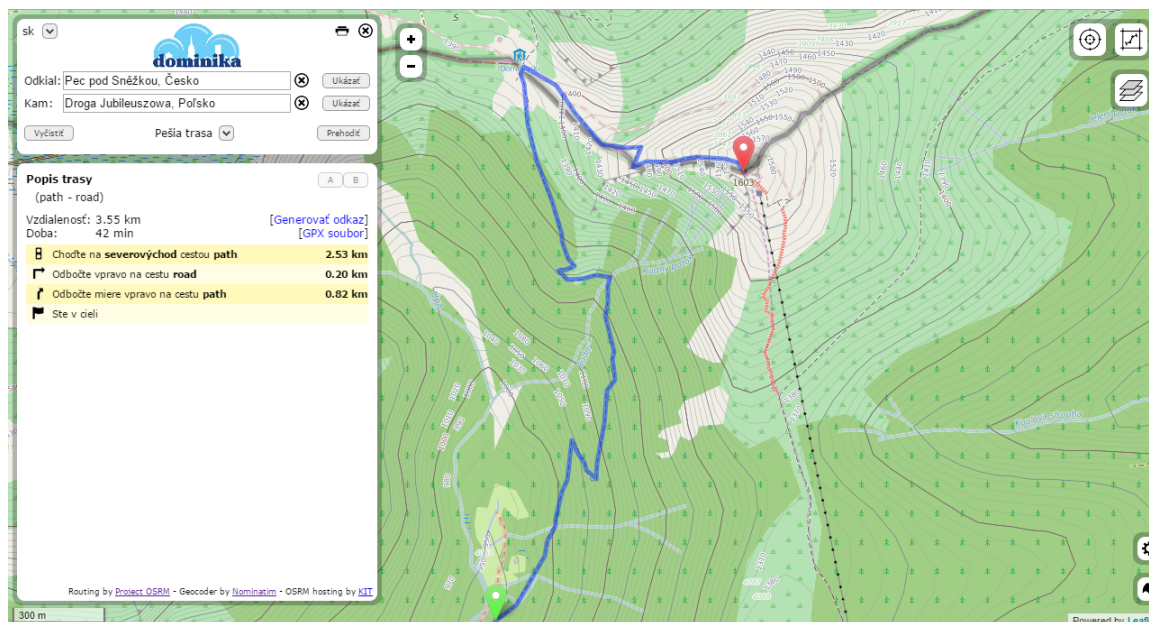
Extractor používa knižnicu **stxxl**⁶, ktorá implementuje paralelný zápis na viac fyzických diskov. To zvyšuje rýchlosť extrakcie distribúciou záťaže medzi jednotlivé disky. Extrakcia bude fungovať aj s predvoleným nastavením, ale rýchlosť nebude optimálna. Preto sa odporúča vytvoriť config⁷ popisujúci fyzické disky, ktoré potom extraktor bude schopný efektívne využiť.

⁶<https://github.com/stxxl/stxxl>

⁷http://stxxl.sourceforge.net/tags/master/install_config.html

3.7.2 OSRM frontend

Nainštalujte frontend podľa návodu na adrese <https://github.com/Project-OSRM/osrm-frontent> a potom by ste mali na adrese `http://<váš-server>/osrm/main.html` vidieť užívateľské rozhranie pre vyhľadávanie ciest založené na knižnici Leaflet. Ukážka na obrázku 3.2.



Obrázek 3.2: Cesta na Sněžku pri zoom-e 15

3.8 Vrstevnice

Výškové dáta poskytuje NASA, ktorá v roku 2000 uskutočnila misiu Shuttle Radar Topography Mission, počas ktorej získala tieto dáta pre väčšinu planéty. Sú poskytované v eponymnom formáte SRTM. S týmto formátom však nevie pracovať mapnik. Návod ako vytvoriť záznam, ktorý vložíme do mapnik stylesheet-u je na OSM Wiki⁸, kde som ho upravila, pretože bol neaktuálny a nefunkčný. Existujú 2 prístupy, buď tieto dáta nahrať do databázy, alebo z nich vytvoriť shapefiles. Vybrala som si druhý prístup a to hlavne preto, že v tomto prístupe bolo jednoduchšie vybrať si iba vrstevnice pre Českú republiku.

Súbory som stiahla z http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/Eurasia/^[1], územie Českej republiky ich pokrýva 33. Aby sme s istotou pokryli celé územie ČR, je potrebné stiahnuť súbory so súradnicami E v rozmedzí 12 - 19 a zároveň N v rozmedzí 48 - 52, ako vyplýva z tabuľky hraničných bodov Českej republiky 3.1.

Ďalej postupujte podľa návodu na OSM Wiki⁹, tj. vyrobte shapefiles pomocou utility `gdal_contour` (súčasť mapniku), indexujte vyrobené shapefiles, vyrobte mapnik stylesheet súbory a ich obsah potom prekopírujte do vášho hlavného `style.xml`. Pozor, jednotlivé

⁸<http://wiki.openstreetmap.org/wiki/Contours>

⁹http://wiki.openstreetmap.org/wiki/Contours#The_shapefiles_approach

Bod	Názov miesta	Súradnice
západný	Krásná u Aše	12° 06' východnej dĺžky
východný	Bukovec u Jablunkova	18° 52' východnej dĺžky
severný	Lobendava u Šluknova	51° 03' severnej šírky
južný	Vyšší Brod	48° 33' severnej šírky

Tabulka 3.1: Prevzaté z <http://www.zemepis.com/gpcr.php>.

vrstvy sa vykresľujú postupne, tj. pokiaľ vložíte vrstevnice na začiatok, prekreslia ich všetky ostatné elementy a nebude ich takmer vidieť. Reštartujte render démona, vyčistite cache a skúste vyrenderovať dlaždice znovu. Mali by obsahovať vrstevnice od zoomu 11 do zoomu 18.

3.9 Turistické trasy

Dáta o tursitických trasách v Českej republike vytvára a udržiava Klubu českých turistů¹⁰. v databáze sú identifikované hodnotou „cz:KČT“ v stĺpci „operator“ v tabuľke `planet_osm_line`. Podrobnejšie údaje ako ich značka, alebo odkiaľ kam vedú si uchovávajú v stĺpci „tags“ datového typu `hstore`. Preto bolo potrebné importovať dáta pomocou `osm2pgsql` s použitím parametra `--hstore`.

Teraz je potrebné vytvoriť dotazy do databázy na turistické trasy, ich štýly a tieto záznamy vložiť do mapnik stylesheetu. Dá sa to robiť ručne, ja som si však zvolila desktopový program TileMill, ktorý sa dokáže pripojiť do databázy, spustiť SQL dotaz a jeho výsledok zobrazí na obrazovke. Okrem toho umožňuje upravovať štýl zobrazených prvkov v jazyku CartoCSS, ktorý je svojou syntaxou podobný jazyku CSS¹¹. Na obrázku 3.3 môžete vidieť ukážku grafického užívateľského rozhrania TileMill-u s importovanými českými turistickými trasami.

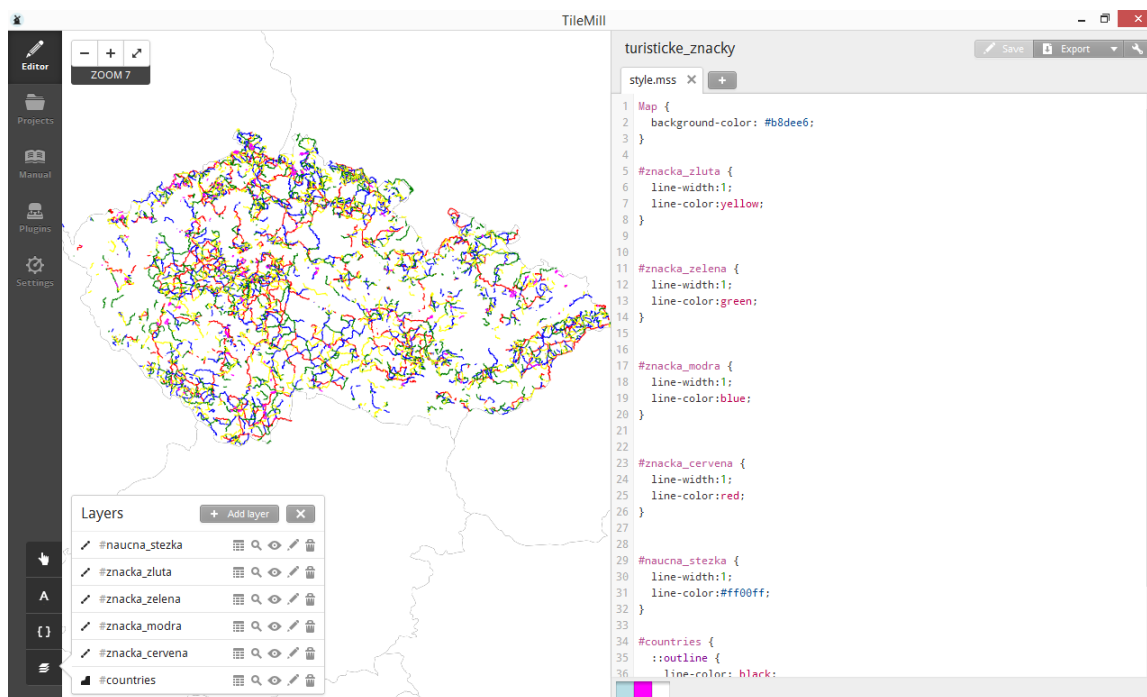
V programe TileMill najskôr vytvorte nový projekt, pomenujte ho, otvorte ho pre editáciu a vľavo dole vyhľadajte možnosť Pridať vrstvu¹². Kliknite na PostGIS a vyplňte požadované parametre. Ja som použila tento dotaz pre zelenú turistickú značku a analogicky potom červenú, žltú a modrú značku a náučný chodník.

```
(SELECT osm_id ,
        way
FROM   planet_osm_line
WHERE  (
        route = 'foot '
        OR   route = 'hiking '
        OR   route = 'ski '
        OR   route = 'horse ' )
AND    operator = 'cz:KČT'
AND    tags @> hstore( 'osmc:symbol' , 'green:white:green_bar' ))
as zelena
```

¹⁰<http://www.kct.cz/>

¹¹CSS - Cascading Style Sheets, <http://www.w3.org/Style/CSS/>

¹²Popis grafického užívateľského rozhrania TileMill na <https://www.mapbox.com/tilemill/docs/manual/interface-tour/#layers-list>



Obrázek 3.3: Turistické značky v programe TileMill

Zoznam dostupných typov turistických značiek sa nachádza na OSM Wiki http://wiki.openstreetmap.org/wiki/WikiProject_Czech_Republic/OTM_zna%C4%8Dkov%C3%BD_kl%C3%AD%C4%8D. Keď máte vytvorené všetky vrstvy, ktoré chcete a ich štýly, kliknite na export a zvolte možnosť Mapnik XML. Exportovaný súbor otvorte a jeho obsah skopírujte do vášho `style.xml`, ktorý používa mapnik (resp. renderd) na renderovanie na príslušné miesto. U mňa je až úplne na konci, tj. turistické trasy sa renderujú ako posledné a prekrývajú tak cesty, či vrstevnice.

3.10 Pravidelná aktualizácia

Dáta sa neustále menia, pretože sú open source a členovia komunity ich neustále upravujú. Napísala som 3 skripty, ktoré sa o pravidelnú aktualizáciu starajú. Tieto skripty sú priložené v prílohe a ich funkcionality popísané v tomto odstavci.

3.10.1 `check_for_updates.sh`

Na pravidelnú aktualizáciu som napísala skript, ktorý spúšťa Cron. Tento skript najskôr overí, či sú nejaké aktualizácie k dispozícii tým, že porovná súbor `state.txt`, ktorý obsahuje číslo verzie dát v databáze s najaktuálnejším súborom `state.txt`, ktorý stiahne z <http://download.geofabrik.de/europe/czech-republic-updates/state.txt>. Pokiaľ sa čísla verzií líšia, spustí skript `update_db.sh`, ktorý zmenové súbory stiahne a pripojí do databázy, potom spustí skript `update_routing.sh`. Ak nie, skúsi to znovu za istý čas, ktorý závisí od toho, ako je nastavený cron.

Je vhodné aby sa skript automaticky spúšťal v nejakom časovom intervale. Ja som

zvolila interval dvakrát denne (o 10. hodine ráno a večer) a toto automaticé spúšťanie som zabezpečila tým, že skript spúšťam skript s nízkou prioritou pomocou Cron démona. Aby sa skript týmto spôsobom spúšťal, je potrebné spustiť príkaz:

```
crontab -e
```

Tento príkaz otvorí k editácii užívateľský crontab, do ktorého je potom treba pridať riadok:

```
00 10,22 * * * nice -n 15 /home/dominika/check_for_updates.sh >>
/var/log/osm/update.log 2>&1 &
```

A potom reštartovať cron démona:

```
service cron restart
```

Výstup sa pripája na koniec súbor do /var/log/osm/update.log. Pri behu je možné ho sledovať pomocou:

```
tail -f /var/log/osm/update.log
```

3.10.2 update_db.sh

Skript používa nástroj Osmosis[5] a je priložený ako príloha. Pred spustením je potrebné skript otvoriť a podľa návodu v komentároch upraviť. Je potrebné myslieť na to, že aktualizácia (databáza a routovania) môže trvať aj niekoľko hodín, preto nie je vhodné týmto procesom priradiť všetky dostupné systémové prostriedky, aby počas behu aktualizácií mohol systém obsluhovať aj požiadavky na servírovanie dlaždíc a ďalšie spustené procesy.

Ukážka z jadra skriptu.

```
echo "${0}:_Executing_wget_to_get_latest_changeset_number."
wget -O ~/.osmosisupdate/state.txt.running http://download.
geofabrik.de/europe/czech-republic-updates/state.txt

echo "${0}:_Executing_osmosis."
osmosis --read-replication-interval workingDirectory=/home/
dominika/.osmosisupdate --simplify-change --write-xml-change

# Upravte hodnotu parametru --number-processes na pocet jadier
# Vasho systemu a --cache na zhruba 75% Vasej pamate
echo "${0}:_Executing_osm2pgsql."
osm2pgsql --append -d czech -U postgres -H localhost -P 5432 --
slim --hstore --number-processes 8 --cache 6000 change.osc -
e15 -o expire.list

echo "${0}:_Rendering_expired_tiles."
cat expire.list | render_expired --map=osm --min-zoom=10

echo "${0}:_Removing_expired_tiles_list."
unlink expire.list
```

```
echo "${0}:_Updating_state.txt"
unlink ~/.osmosisupdate/state.txt
mv ~/.osmosisupdate/state.txt.running ~/.osmosisupdate/state.txt
```

3.10.3 update_routing.sh

Pre update routovacích dát je bohužiaľ potrebné stiahnuť znovu celý PBF súbor pre Českú republiku. Potom spustiť nástroj OSRM backend, ktorý pripraví routovacie dáta a reštartovať démona, ktorý routovacie dáta servíruje.

Ukážka z jadra skriptu.

```
wget -O /mnt/resource/mapy/czech-republic-latest.osm.pbf http://
download.geofabrik.de/europe/czech-republic-latest.osm.pbf

# Nizky pocet pouzitych jadier, aby system priebezne zvladal
# renderovanie, servirovanie a routovanie
/home/dominika/osrm-backend/build/osrm-extract --threads 2 --
profile /home/dominika/osrm-backend/profiles/foot.lua /mnt/
resource/mapy/czech-republic-latest.osm.pbf
/home/dominika/osrm-backend/build/osrm-prepare --threads 2 --
profile /home/dominika/osrm-backend/profiles/foot.lua /mnt/
resource/mapy/czech-republic-latest.osrm

echo "${0}:_Stopping_routing_daemon."
/etc/init.d/osrm-backend stop

echo "${0}:_Copying_updated_data."
mv /mnt/resource/mapy/czech-republic-latest.osrm* /home/dominika/
osrm-data/

echo "${0}:_Starting_routing_daemon."
/etc/init.d/osrm-backend start
```


Kapitola 4

Záver

V tejto bakalárskej práci som študovala postup tvorby vlastných máp z podkladu OpenStreetMap. V kapitole 1 som uviedla poznatky o projekte OpenStreetMap a zoznámila čitateľov s niektorými pojmami špecifickými pre tvorbu papierových a digitálnych máp.

V ďalšej fáze som potom študovala proces tvorby máp z podkladu OpenStreetMap. Tento proces je zložitý, skladá sa z viacerých krokov a je značne rozvetvený. Nástroje, ktoré dokážu exekúovať jednotlivé kroky sú zväčša tak, ako samotné dáta open source. To je na jednej strane výhoda, pretože ktokoľvek, kto má záujem sa môže dozvedieť ako fungujú a implementovať svoje vlastné, prípadne existujúce implementácie upraviť. Na strane druhej, znamená to, že ich vývoj a dokumentácia je v rukách autorov a tým pádom môže byť zastaralá, alebo vývoj už ukončený.

Preštudovala som teda funkcionality dostupných nástrojov a v kapitole 2 nástroje s rovnakou, podobnou, alebo prekrývajúcou sa funkcionalitou porovnávala. Z porovnania som potom vybrala jedného kandidáta, ktorý najlepšie spĺňal moje požiadavky a to aby bol stále aktívne vyvíjaný, alebo aspoň udržiavaný, aby podľa tretieho bodu zadania umožňoval rozumným spôsobom mapy aktualizovať a bol i čo najjednoduchší na použitie. Vybrať vhodného kandidáta nebolo vždy jednoduché.

V kapitole 3 som potom s vybranými nástrojmi pracovala na svojom serveri a ako vyplýva zo štvrtého bodu zadania, snažila som sa tento postup dôkladne zdokumentovať. Na niektoré fázy postupu som našla zrozumiteľné a funkčné návody a tak sa na ne odkazujem a prípadne ich dovysvetľujem. O častiach, ktoré boli menej popísané a jasné som sa rozpísala viac, aj s vysvetlením, prečo je to, čo odporúčam potrebné robiť a aký je očakávaný výsledok. Navyše som počas tohto postupu zistila, že neexistuje žiadny nástroj, ktorý by automatizoval pravidelnú aktualizáciu máp a tak som napísala 3 jednoduché skripty, ktoré tento problém riešia.

Myslím si, že zadanie som splnila, preštudovala som proces tvorby vlastných máp z podkladu OpenStreetMap, analyzovala jeho jednotlivé kroky a určila ich výhodné varianty. Navrhla a implementovala som funkčný postup, ktorý umožňuje časté aktualizácie, podporuje vyhľadávanie trás na mape (routovanie), zobrazuje vrstevnice a turistické chodníky. Celý postup som dôkladne zdokumentovala.

Literatura

- [1] SRTM [online]. http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/Eurasia/, 2000 [cit. 2015-04-26].
- [2] OpenStreetMap [online]. <https://www.openstreetmap.org/copyright>, 2004 [cit. 2015-04-26].
- [3] OpenStreetMap [online]. <https://www.openstreetmap.org>, 2004 [cit. 2015-04-26].
- [4] Elements - OpenStreetMap Wiki [online]. <http://wiki.openstreetmap.org/wiki/Elements>, 2014-10-07 [cit. 2015-04-26].
- [5] Osmosis - OpenStreetMap Wiki [online]. <http://wiki.openstreetmap.org/wiki/Osmosis>, 2015-02-24 [cit. 2015-05-01].
- [6] PBF Format - OpenStreetMap Wiki [online]. http://wiki.openstreetmap.org/wiki/PBF_Format, 2015-04-10 [cit. 2015-04-26].
- [7] Open Data Commons Open Database License (ODbL) — Open Data Commons [online]. <http://opendatacommons.org/licenses/odbl>, [cit. 2015-04-26].
- [8] Open Source Routing Machine [online]. <http://project-osrm.org>, [cit. 2015-04-26].
- [9] osm2pgrouting - Import OSM data into pgRouting Database – Open Source Routing Library [online]. <http://pgrouting.org/docs/tools/osm2pgrouting.html>, [cit. 2015-04-26].
- [10] PostgreSQL: Documentation: 9.4: The Password File [online]. <http://www.postgresql.org/docs/9.4/static/libpq-pgpass.html>, [cit. 2015-05-01].
- [11] openstreetmap/osm2pgsql [online]. <https://github.com/openstreetmap/osm2pgsql>, [cit. 2015-05-03].
- [12] API v0.6 OpenStreetMap Wiki [online]. http://wiki.openstreetmap.org/wiki/API_v0.6, [cit. 2015-05-06].
- [13] Shapefiles - OpenStreetMap Wiki [online]. <http://wiki.openstreetmap.org/wiki/Shapefiles>, [cit. 2015-05-06].
- [14] Tile disk usage - OpenStreetMap Wiki [online]. http://wiki.openstreetmap.org/wiki/Tile_disk_usage, [cit. 2015-05-06].

- [15] HOWTO : Performance tuning for PostgreSQL on Ubuntu/Debian — Samiux's Blog [online]. <https://samiux.wordpress.com/2009/07/26/howto-performance-tuning-for-postgresql-on-ubuntudebian>, [cit. 2015-05-10].
- [16] Osm2pgsql/benchmarks - OpenStreetMap Wiki [online]. http://wiki.openstreetmap.org/wiki/Osm2pgsql/benchmarks#Best_Results, [cit. 2015-05-10].
- [17] PostgreSQL: Documentation: 9.4: Resource Consumption [online]. <http://www.postgresql.org/docs/9.4/static/runtime-config-resource.html>, [cit. 2015-05-10].
- [18] Pricing - Virtual Machines (VMs) — Microsoft Azure [online]. <http://azure.microsoft.com/en-us/pricing/details/virtual-machines/#Linux>, [cit. 2015-05-10].
- [19] Welcome to MapServer – MapServer 7.0.0-beta1 documentation [online]. <http://mapserver.org>, [cit. 2015-05-10].
- [20] openstreetmap/mod_tile [online]. https://github.com/openstreetmap/mod_tile, [cit. 2015-05-11].
- [21] Boyle, M.: Department of Defense World Geodetic System 1984-It's definition and relationship with local geodetic systems. Technická zpráva, DMA Technical Report 83502.2., Washington, DC, 1987.
- [22] Consortium, O. G.; aj.: OpenGIS Web Map Service version 1.3. 0. 2006.

Příloha A

Obsah CD

Shell skripty:

- `check_for_updates.sh`
- `update_db.sh`
- `update_routing.sh`

Unixový démon:

- `osrm-backend`

Webová stránka:

- `index.html`