



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# NÁVRH A IMPLEMENTACE GEOPORTÁLŮ S POUŽITÍM OPEN-SOURCE GIS NÁSTROJŮ

GENERAL SOLUTION OF GEO-PORTALS BASED ON OPEN-SOURCE GIS TOOLS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ KUČERA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN HRUBÝ, Ph.D.

BRNO 2008

# Zadání

1. Prostudujte UMN MapServer, OpenLayers a příbuzné technologie. Zaměřte se na kritické zhodnocení jejich návrhu, implementace a provozovatelnosti.
2. Prostudujte standardy webových služeb dle specifikace Open Geospatial Consortium (OGC WPS). Prostudujte GRASS jako cílovou platformu implementace OGC WPS.
3. Navrhněte řešení geoportálů, které bude vycházet z OGC WPS a poznatků ze studijní etapy.
4. Implementujte OGC WPS specifikace v prostředí nástroje GRASS.
5. Vytvořte demoaplikaci geoportálu založenou na vašem řešení. Systém dokumentujte a testujte.

## **Licenční smlouva**

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

## **Abstrakt**

Tato bakalářská práce se zabývá návrhem a implementací serveru umožňujícího napojení GIS GRASS pomocí služby WPS na Internet. Pro řešení tohoto projektu byly použity nástroje GIS GRASS a OpenLayers. Součástí práce je seznámení se standardy OGC a užitými nástroji.

## **Klíčová slova**

GIS, Geografické informační systémy, OGC, WPS, Web Processing Service, WMS, Web Map Service, WCS, Web Coverage Service, WFS, Web Feature Service, Mapserver, GRASS, OpenLayers, GNU/Linux, C/C++, web, internet, server, JavaScript

## **Abstract**

The bachelor's thesis is concerned with design and implementation of server which provides connection between GIS GRASS and Internet through WPS service. Part of this thesis is devoted to description of OGC standards and used tools.

## **Keywords**

GIS, Geographic information system, OGC, WPS, Web Processing Service, WMS, Web Map Service, WCS, Web Coverage Service, WFS, Web Feature Service, Mapserver, GRASS, OpenLayers, GNU/Linux, C/C++, web, internet, server, JavaScript

## **Citace**

Ondřej Kučera: Návrh a implementace geoportálů s použitím Open-Source GIS nástrojů, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Návrh a implementace geoportálů s použitím Open-Source GIS nástrojů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Martina Hrubého Ph.D.

.....  
Ondřej Kučera  
12. května 2008

## Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Martinu Hrubému, Ph.D., za jeho trpělivost a cenné připomínky při řešení projektu. Dále svým rodičům a přátelům, kteří mě podpořovali při práci.

© Ondřej Kučera, 2008.

*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

<b>1 Úvod</b>	<b>3</b>
<b>2 Standardy sloužící k popisu geografických služeb</b>	<b>4</b>
2.1 OGC WPS (OpenGIS Web Processing Service)	4
2.1.1 Žádost GetCapabilities	5
2.1.2 Žádost DescribeProcess	5
2.1.3 Žádost Execute	5
2.2 OGC WMS (OpenGIS Web Map Service)	6
2.2.1 Žádost GetCapabilities	6
2.2.2 Žádost GetMap	6
2.2.3 Žádost GetFeatureInfo	6
2.3 OGC WFS (Web Feature Service)	6
2.3.1 Žádost GetCapabilities	7
2.3.2 Žádost GetFeature	7
2.3.3 Žádost GetFeatureType	7
2.3.4 Žádost GetGmlObject	7
2.4 OGC WCS (Web Coverage Service)	7
2.4.1 Žádost GetCapabilities	7
2.4.2 Žádost DescribeCoverage	7
2.4.3 Žádost DescribeCoverage	7
<b>3 Nástroje pro správu a publikaci geografických dat</b>	<b>8</b>
3.1 UMN Mapserver	8
3.1.1 Popis	8
3.1.2 Technologie Mapserveru	8
3.1.3 Rozšíření možností Mapserveru	9
3.1.4 Funkce Mapserveru	9
3.1.5 Mapfile	9
3.2 OpenLayers	10
3.3 GIS GRASS	11
3.3.1 Technologie GIS GRASS	11
3.3.2 Ukázka práce s GIS GRASS	12
<b>4 Popis řešení a jeho implementace</b>	<b>15</b>
4.1 Definice nového procesu	15
4.2 Podporované žádosti standardu OGC WPS	16
4.2.1 WPS GetCapabilities	16
4.2.2 WPS DescribeProcess	16

4.2.3	WPS Execute . . . . .	16
4.3	Příklad definice nového procesu . . . . .	17
4.4	Testování Geoserveru . . . . .	18
<b>5</b>	<b>Demonstrace řešení na jednoduché demoaplikaci</b>	<b>19</b>
5.1	Příklad ovládání procesu pomocí nástroje OpenLayers . . . . .	19
<b>6</b>	<b>Popis instalace</b>	<b>22</b>
6.1	Instalace nástrojů . . . . .	22
6.1.1	GIS GRASS . . . . .	22
6.1.2	Webový server Apache . . . . .	22
6.1.3	Knihovny . . . . .	22
6.2	Instalace Geoserveru . . . . .	23
6.3	Uvedení demoaplikace do provozu . . . . .	23
<b>7</b>	<b>Závěr</b>	<b>24</b>
<b>8</b>	<b>Seznam příloh</b>	<b>27</b>
<b>9</b>	<b>Příloha B</b>	<b>28</b>
9.1	Ukázky požadavků a XML výstupů . . . . .	28
9.1.1	WPS žádost GetCapabilities . . . . .	28
9.1.2	WPS žádost DescribeProcess . . . . .	30
9.1.3	WPS žádost Execute . . . . .	32
<b>10</b>	<b>Příloha C</b>	<b>35</b>
10.1	Ukázka konfiguračního souboru Geoserveru . . . . .	35
10.2	Ukázka konfiguračního souboru Mapfile pro Mapserver . . . . .	36
<b>11</b>	<b>Příloha D</b>	<b>40</b>
11.1	Adresářová struktura zdrojového kódu . . . . .	40

# Kapitola 1

## Úvod

V dnešní době se můžeme stále častěji setkat se sílícím zájmem o oblast geografických informačních systémů (dále GIS). Roste také skupina jejich uživatelů a narozdíl od dob minulých, kdy bylo jejich užívání spíše doménou profesionálů, se řadí také do popředí zájmu veřejnosti.

Velkou výhodou se stávají stále se rozvíjející možnosti publikování geografických dat v prostředí dnešního Internetu, jež v podstatě nezná hranic, a tak je možné služby GIS nabídnout obrovské skupině uživatelů.

Velmi oblíbené jsou zejména internetové servery poskytující mapové služby mezi jejichž zástupce patří známá služba Google Maps společnosti Google či u nás hojně využívaná služba Mapy.cz společnosti Seznam. Zmíněné služby slouží hlavně pro grafickou reprezentaci (zobrazování) map a neumožňují s nimi provádět složitější operace a výpočty. Nejedná se tedy o plnohodnotné GIS. Ty bývají standardně vybaveny robustními nástroji schopnými editovat geografická data, komponovat tematické mapové vrstvy a také provádět náročné geografické výpočty.

GIS jsou svým rozsahem prvotně určeny pro nasazení na desktopu. Často se totiž ovládají skrze grafické uživatelské rozhraní, a tak není prakticky možné jejich služby poskytovat na síti.

Cílem této bakalářské práce bylo navrhnout a implementovat řešení, které by uživatelům sítě umožnilo vzdáleně využívat všech výhod desktopového GIS, ale také zároveň poskytnout možnost jednoduchého ovládání bez nutnosti znát cílovou platformu daného GIS, od kterého je uživatel v podstatě úplně odstíнен. Veškerá komunikace probíhá na bázi výměny dat mezi servrovou částí, na které je umístěn daný GIS spolu s rozhraním umožňujícím jeho napojení na síť a klientem, kterým může být jakýkoliv počítač na síti s nainstalovaným internetovým prohlížečem.

Cíl práce spočíval právě v implementaci rozhraní na jedné straně zprostředkovávající komunikaci s klientem a na straně druhé vykonávající požadovanou operaci GIS.

Jak již napovídá název práce, bude pozornost zaměřena převážně na využití Open-Source nástrojů a platformu GNU/Linux, která těmto nástrojům poskytuje svobodné pole působnosti nezátížené uzavřenými proprietárními licencemi.

Kapitola 2 se zabývá specifikacemi otevřených standardů, které popisují způsob výměny geografických dat a služeb v prostředí počítačových sítí a možnostmi jejich implementace v rámci aplikací. Kapitola 3 stručně popisuje nejčastěji užívané nástroje pro správu a publikaci geografických dat. Kapitola 4 se podrobněji věnuje návrhu a vlastní implementaci konečného řešení. V kapitole 5 je prezentována jednoduchá aplikace demonstrující možnost propojení klienta OpenLayers spolu se službou WPS.



## Kapitola 2

# Standardy sloužící k popisu geografických služeb

Vytvářením specifikací popisující způsob výměny geografických dat a služeb se zabývá The Open Geospatial Consortium (OGC). Jedná se o mezinárodní organizaci sdružující dobrovolníky z celého světa. Mezi jeho členy můžeme najít zástupce známých světových společností, organizací a univerzit jako jsou například společnost ESRI vyvíjející známý komerční GIS ArcGIS, NASA, Google, Microsoft, Masarykova univerzita a další. Mezi nejznámější OGC standardy patří specifikace WMS (Web Map Service), WCS (Web Coverage Service), WFS (Web Feature Service) a v roce 2007 nově schválený standard WPS (Web processing service) přejmenovaný z původního GPS (Geoprocessing Service), aby se předešlo chybnému zaměňování termínu s polohovacím systémem GPS (Global Positioning System). Tyto standardy jsou zahrnuty v tzv. konceptu OWS (OGC Web Services). Služba neboli *service* je v tomto případě chápána jako provedení určité operace vyžadované klientem jež je zpracována na straně serveru a následně její výsledek odeslán zpět klientovi.

### 2.1 OGC WPS (OpenGIS Web Processing Service)

Specifikace tohoto standardu je poměrně nedávnou záležitostí. Její poslední definice pro verzi 1.0.0 je popsána v dokumentu [8].

Standard WPS popisuje jakým způsobem je možné poskytovat geoprostorové operace neboli procesy a také způsob komunikace mezi klientskou aplikací a serverem. Celý protokol je postaven na jazyku XML a pro komunikaci je využito protokolu HTTP a jeho dotazovacích metody GET a POST. Pro komunikaci stačí teoreticky jakýkoliv internetový prohlížeč, avšak vypisování všech potřebných paramterů do URL je značně nepohodlné a nepřehledné. Proto se nejčastěji využívá komplexnějších klientů, se kterými uživatel pracuje výhradně prostřednictvím grafického uživatelského rozhraní usnadňujícího formulaci požadavku. Mezi takovéto aplikace patří nástroj OpenLayers viz. 3.2.

Jedinou volně dostupnou implementací služby WPS je aplikace PyWPS napsaná v programovacím jazyce Python viz. [14]

Podobně jako většina OGC standardů spadajících do konceptu OWS umožňuje WPS tři typy žádostí (**request**) směřujícím od klienta k serveru: **GetCapabilities**, **DescribeProcess** a **Execute**.

### 2.1.1 Žádost GetCapabilities

Pomocí URL je serveru předán parametr `request=GetCapabilities`. Dále musí být zadán parametr `service`, který jednoznačně specifikuje požadovanou službu, v tomto případě `service=WPS`. Žádost `GetCapabilities` využívají totiž i další standardy a takto je zabráněno jejich záměně.

Pokud server rozezná, že se jedná o tento požadavek, odešle klientovi XML dokument obsahující kontaktní informace o provozovateli služby a seznam nabízených procesů (operací) spolu s jejich stručným popisem. V příloze č. 9.1.1 je uveden příklad formulace URL požadavku a ukázka výsledného XML dokumentu.

### 2.1.2 Žádost DescribeProcess

Po zjištění základních informací o službě následuje detailnější popis jednoho uživatelem zvoleného procesu. Jeho název je předán prostřednictvím URL nebo vstupního XML souboru parametrem `Identifier`. Klientovi je zaslán XML dokument obsahující definici vstupních a výstupních dat, na jejichž základě může klient formulovat požadavek na provedení určité operace a zároveň také ví, jaký výsledek může očekávat. Ukázka požadavku a výsledného XML souboru je uvedena viz. 9.1.2

Vstupní a výstupní data mohou být následujících typů:

- `LiteralValue` je hodnota určitého datového typu (`string`, `int`, `float`, ...). Slouží pro přímé zadávání vstupních hodnot.
- `ComplexValue` je obsah datového souboru přímo začleněný do stromu XML dokumentu obsahující vektorová data uložená ve formátu GML <sup>1</sup> nebo rastrová data uložená ve formátu Base64 <sup>2</sup>
- `ComplexValueReference` je odkaz na soubor umístěný buď na lokálním nebo vzdáleném serveru nebo jinou webovou službu která požadovaný soubor nabízí.
- `BoundingBoxValue` je obdéníkový výřez určitého území, které je ohraničeno pomocí čtyř souřadnic. Ty se zadávají jako dvojice souřadnic dvou protilehlých rohů.

### 2.1.3 Žádost Execute

Všechny potřebné informace ke spuštění jsou známy z výsledků předchozích požadavků a nyní následuje spuštění požadovaného procesu. Vstupní data jsou předána buď pomocí URL nebo vstupního XML souboru. Nastavením parametrů `status` a `storeExecuteResponse` může uživatel ovlivnit provádění procesu. Pokud parametr `status=true` bude uživateli bezprostředně vrácen XML dokument informující o tom, zda li proces byl a nebo nebyl přijat a také odkaz na místo, kde si bude možné přečíst, jak proces dopadl. Nastavením parametru `storeExecuteResponse=true` je povoleno uložení výsledku odpovědi přímo na serveru, pokud tuto možnost podporuje. Využití těchto parametrů je doporučeno zejména u procesů trvajících dlouho dobu, během které není výhodné pro server ani klient udržovat neustálé spojení.

Příklad formulace žádosti a výsledného XML dokumentu po provedení operace `Execute`.9.1.3

---

<sup>1</sup>GML (Geography Markup Language) je XML gramatika definována OGC sloužící k vyjádření geografických tvarů [17]

<sup>2</sup>Base64 je datový formát zobrazující binární data pomocí tisknutelných znaků ASCII [16]

## 2.2 OGC WMS (OpenGIS Web Map Service)

WMS jeden z nejznámějších a nejužívanějších standardů OGC. Tato služba slouží pro zpřístupnění tématicky zaměřených mapových dat (rastrů). Stejně jako služba WPS využívá pro svoji komunikaci HTTP protokol. Na straně WMS serveru jsou uložena georeferencovaná data nejčastěji se užívá formátů SHP (ESRI Shapefile) GeoTIFF či dat uložených v databázi, která jsou na žádost klienta primárně převedena na rastry různých formátů (PNG, JPEG, ...) a zaslána jako výsledek.

Stejně jako WPS umožňuje i WMS tři typy základních požadavků: `GetCapabilities`, `GetMap` a `GetFeatureInfo`.

### 2.2.1 Žádost `GetCapabilities`

Požadavek `GetCapabilities` je pro většinu standardů OGC společný a slouží pro získání informací a možností, které služba nabízí. Klient na jeho základně může formulovat přesnější požadavek pro získání detailnějších dat v tomto případě získání tématických mapových vrstev. Pomocí URL musí být předány serveru nejméně dva parametry `request=GetCapabilities` a `service=WMS`. Jako výsledek klient opět obdrží XML dokument popisující možnosti služby.

### 2.2.2 Žádost `GetMap`

Žádost server zaregistruje skrze parametr URL `request=GetCapabilities`. Spolu s ním je nutné zadat další povinné parametry - `version`, definující používanou verzi služby WMS<sup>1</sup>, `layers`, obsahující list výsledných mapových vrstev oddělených čárkou, `styles`, čárkami oddělený list stylů jakými budou výsledné mapové vrstvy graficky zformátovány, `CRS`, název požadovaného souřadnicového referenčního systému, `bbox`, obdélníkem vymezený výsek prostoru, `width`, šířka výstupní mapy v pixelech, `height`, výška v pixelech a nakonec `format`, formát požadované mapy (PNG, TIFF, JPEG ...). Detailní soupis všech parametrů je popsán v dokumentu [9].

### 2.2.3 Žádost `GetFeatureInfo`

Narozdíl od předchozích žádostí nemusí být dle specifikace serverem nutně podporována. Je uvedena parametrem `request=GetFeatureInfo` a klientovi je navrácen XML dokument obsahující souhrn atributů daného prvku na mapě o určitých souřadnicích. Požadavek musí tedy obsahovat další povinné parametry viz. [9].

## 2.3 OGC WFS (Web Feature Service)

Tato služba je určena především pro práci s vektorovými daty. Ta jsou uložena ve formátu GML viz. 2.1.2. Pro práci je na straně klienta vyžadován nástroj, který umožní formulace dotazu a také jeho výsledek interpretovat. S běžným internetovým prohlížečem by si uživatel v tomto případě nevystačil. Nevýhodou této služby je velikost výsledných dat, ta mohou narůst podle velikosti území do desítek až stovek megabajtů. Tato služba nadále umožňuje manipulaci s daty uloženými na serveru.

Základní podporované požadavky:

---

<sup>1</sup>Pozn.: Jeden server může poskytovat zároveň více verzí jejichž specifikace se od sebe odlišují.

### 2.3.1 Žádost GetCapabilities

Výsledkem je XML dokument popisující možnosti služby, seznam vrstev, popis atributů. Funkce žádostí tohoto typu je pro zmíněné standardy takřka společná.

### 2.3.2 Žádost GetFeature

Server vrátí klientovi XML dokument obsahující požadované geoobjekty.

### 2.3.3 Žádost GetFeatureType

Pro správnou interpretaci výsledného XML dokumentu klientem je nutné mu poskytnout také schéma výsledného dokumentu, které je mu na žádost zasláno.

### 2.3.4 Žádost GetGmlObject

Nemusí být povinně implementována každým serverem. Vrátí objekt, na který se odkazuje zadaný XLink<sup>2</sup> [18]

## 2.4 OGC WCS (Web Coverage Service)

Tato služba se využívá pro poskytování geografických dat tzv. „coverages“, což jsou digitální geografické informace reprezentující povrch planet, nejčastěji planety Země. Slouží k přenosu satelitních dat a dále umožňuje pracovat nejen v dvourozměrném a trojrozměrném prostoru, ale přidává i rozměr čtvrtý a tím je čas. Je tak možné podrobně sledovat pochody odehrávající se např. na Zemském povrchu a dále je analyzovat. Příkladem analýzy může být sledování plochy Amazonského pralesa dokazující jeho neustálé mýcení.

Podporované žádosti na WCS server:

### 2.4.1 Žádost GetCapabilities

Význam této žádosti je podobný jako u ostatních výše uvedených služeb.

### 2.4.2 Žádost DescribeCoverage

Vrátí XML dokument popisující vlastnosti rozložení uvedeného v dotazu.[18]

### 2.4.3 Žádost DescribeCoverage

Vrátí samotné rozložení - skupinu hodnot v určitém prostoru (zobrazit ji lze např. pomocí rastrového formátu GeoTIFF). [18]

---

<sup>2</sup>XLink: odkaz používaný v XML dokumentech

## Kapitola 3

# Nástroje pro správu a publikaci geografických dat

### 3.1 UMN Mapserver

#### 3.1.1 Popis

Jedná se o jednu z nejpoužívanějších a nejznámějších Open-Source aplikací pro publikaci kartografických dat v prostředí dnešního Internetu. Mapserver se zrodil ze spolupráce Univerzity v Minnesotě, Minnesotského odboru pro přírodní zdroje a NASA. Za konkurenci Mapserveru můžeme považovat produkt ArcIMS společnosti ESRI, avšak v tomto případě se jedná o zcela komerční software. Slouží hlavně pro poskytování map, s daty není umožněno provádět výpočty či jiné operace. Nejedná se tedy o plnohodnotný GIS. Zástupcem GIS v pravém slova smyslu je například Open-Source nástroj GRASS či komerční produkt ArcGIS společnosti ESRI.

#### 3.1.2 Technologie Mapserveru

Mezi výhody Mapserveru kromě toho, že je k dispozici zcela zdarma, jistě nesporně patří i to, že se jedná o aplikaci, kterou je možno provozovat na většině hlavních platform. Podporovány jsou platformy GNU/Linux, Windows, Mac OSX, Solaris, FreeBSD a další.

Mapserver je relativně velkou aplikací skládající se z mnoha částí, postupně jsou do-programovávány další. Stále čím dál tím častěji je usilováno o zahrnutí podpory standardů pro publikaci geografických dat, zejména standardů definovaných OGC [2](#). V poslední stabilní verzi t.j. verze 5.0.2 jsou Mapserverem implementovány tyto OGC standardy: WMS ([2.2](#)), WFS ([2.3](#)), WCS ([2.4](#)), GML ([2.1.2](#)) a WMC (Web Map Context [\[10\]](#)). Kompletní výčet podporovaných standardů je uveden na domovských stránkách produktu [\[15\]](#).

Mapserver pro svoji funkčnost vyžaduje web server, nejčastěji bývá spojován se známým serverem Apache. Jedná se o CGI aplikaci, která na základně konfiguračního souboru (tzv. Mapfile) vygeneruje grafický či jiný požadovaný výstup. Grafických formátů je podporována velká řada např. PNG, JPEG, GIF, TIFF/GeoTIFF, EPPL7 a další ([\[15\]](#)), je přitom využíváno Open-Source knihovny GDAL. Předávání údajů a požadavků od uživatele (klienta) probíhá skrze parametry, zasílanými serveru metodami HTTP protokolu GET nebo POST. Chceme-li však využít pokročilejších vlastností a implementovat tak komfortnější ovládání, nabízí se možnost využití některého z „populárních“ programovacích jazyků jako je PHP, Python, Perl, Ruby, Java nebo C#.

### 3.1.3 Rozšíření možností Mapserveru

V dnešní době již k Mapserveru neodmyslitelně patří i skupinka dalších programů, knihoven a nástrojů, bez nichž by byl Mapserver ochuzen o spoustu funkcí.

- **PROJ.4**  
knihovna sloužící pro práci s kartografickými zobrazeními umožňující jejich vzájemný převod. [13]
- **GDAL**  
Tuto knihovnu využívá Mapserver pro čtení a zápis rastrových geografických dat. Formátů je podporována opravdu velká řada např. (PNG, GeoTIFF, GIF, JPEG, BMP, nativní rastrové formáty GIS GRASS, ... [1]).
- **OGR**  
Další knihovna, která je přímo zahrnuta do knihovny GDAL a slouží pro práci s geografickými vektorovými formáty. Mezi nejznámější podporované formáty patří ESRI Shapefiles, PostGIS, MySQL, Oracle Spatial a další viz. [11]
- **PostGIS**  
V současnosti dochází čím dál tím častěji k přesunu dat do databázových systémů, které mají bezesporně spoustu výhod. Mezi nejčastější z nich patří zejména rychlost vyhledávání. PostGIS je rozšíření nad objektově-relačním databázovým systémem PostgreSQL, které umožňuje databázi pracovat přímo s prostorově orientovanými objekty (geobjekty). Přidává tak do databáze nové datové typy a funkce. Mimo jiné je Mapserveru díky knihovně OGR umožněno pracovat i s dalšími databázovými systémy.

### 3.1.4 Funkce Mapserveru

S Mapserverem je možné pracovat v několika módech. Základním je **BROWSE**, jehož výsledkem je na základě vytvořené šablony dynamicky generovaná webová stránka s požadovanou mapou. Mód **MAP** se využívá tehdy pokud chce uživatel jako odpověď na svůj požadavek získat pouze obrázek dané mapy. Pro vygenerování obrázků legendy a měřítka se využívá módů **LEGEND** a **SCALEBAR**. Kompletní seznam těchto módů Mapserveru je uveden v dokumentaci [15].

### 3.1.5 Mapfile

Mapfile je konfigurační soubor na základě kterého Mapserver generuje požadovanou mapu. Každý Mapfile je složen z několika částí. Hlavní je část **MAP**, která je v hierarchii konfiguračního souboru kořenovým prvkem.

Stručný souhrn ostatních částí a prvků konfiguračního souboru:

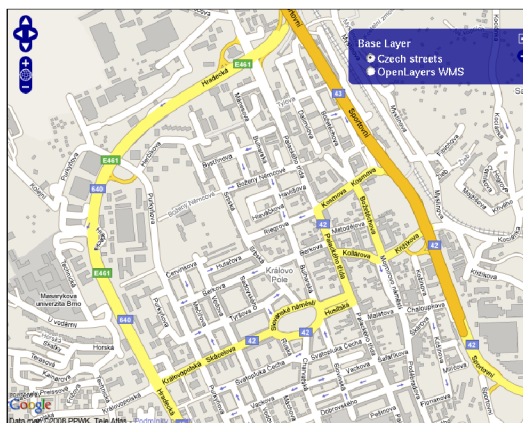
- **NAME** název mapy
- **SIZE** velikost v pixelech
- **UNITS** jednotky vzdálenosti na mapě
- **EXTENT** hraniční souřadnice
- **PROJECTION** definice geografické zobrazení

- IMAGECOLOR barva pozadí
- IMAGETYPE typ obrázku
- OUTPUTFORMAT výběr požadovaného grafického ovladače
- LEGEND část pro nastavení stylu legendy
- SCALEBAR část popisující vzhled a nastavení měřítka mapy
- REFERENCE část popisující referenční mapu<sup>1</sup>
- WEB velmi důležitá část, zde se nastavují metadata pro zpřístupnění mapy přes WMS službu 2.2 a také případné propojení se šablonou pro generování webové stránky
- LAYER pomocí této části jsou definovány jednotlivé mapové vrstvy a každý Mapfile obsahuje takovýchto částí více v závislosti na složitosti požadované mapy

Příklad jednoduchého konfiguračního souboru je uveden v kapitole 10.2

## 3.2 OpenLayers

OpenLayers je knihovna napsaná v jazyce JavaScript umožňující jednoduše do obsahu webových stránek začleňovat mapy. S mapami je možné také interaktivně pracovat. Mimo klasických operací zoom a posunu mapy jsou poslední vydanou verzí 2.6. implementovány další rozšířené vlastnosti jako například možnost animace mapy, pokročilá práce s vektory, reprojekce a další . . . Ovládání OpenLayers probíhá skrze jednoduché okno a je velmi intuitivní (Obrázek 3.2). Ve většině případech si uživatel vystačí pouze s myší a knihovna sama provede formulaci mnohdy velmi náročných požadavků, spojí se s danou službou, zpracuje její odpověď a zobrazí v okně prohlížeče výsledek. Největší výhodou knihovny OpenLayers



Obrázek 3.1: Ukázka rozhraní klienta OpenLayers

je to, že podporuje velké množství formátů pro sdílení geografických dat. Pomocí OpenLayers můžeme na svoji stránku vložit mapy z Mapserveru, Google Maps, Yahoo Maps,

<sup>1</sup>Referenční mapa je zmenšená verze původní mapy sloužící k zobrazení náhledu nebo oblasti současné pozice na mapě.

KaMap a podporovány jsou také standardy OGC WMS, WFS, WMC, KML, a GML [12]. Neustále probíhá práce na začleňování dalších standardů. Ve vývojové větvi je dostupná také implementace standardu WPS, avšak do oficiálního vydání zahrnuta zatím nebyla.

### 3.3 GIS GRASS

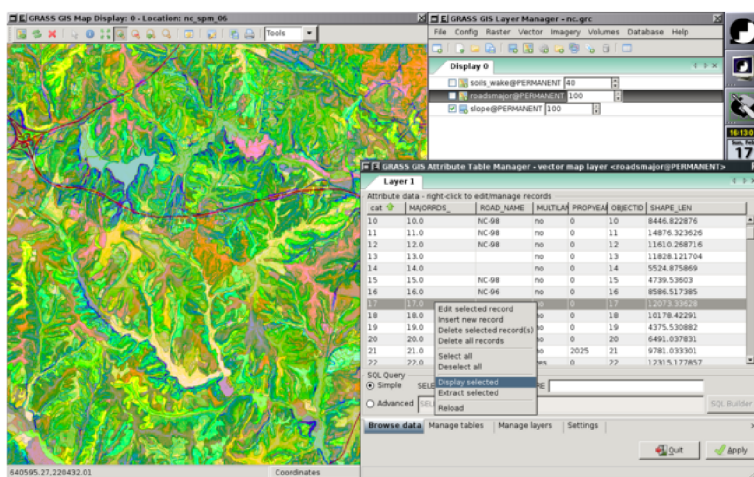
GRASS (Geographic Resources Analysis Support System) je plnohodnotným geografickým informačním systémem, uvolněným pod svobodnou licencí GNU GPL. Patří mezi profesionální nástroje určené pro správu geoprostorových dat, který nemá na poli Open-Source konkurenci, tu je možné hledat pouze u komerčních řešení např. u nástroje ArcGIS společnosti ESRI. Pracuje jak s rastrovými tak vektorovými daty. Umožňuje zpracovávat letecké a družicové snímky, prostorové modelování, pracovat s 3D daty, umožňuje vizualizaci mnoha typů geoprostorových dat, vytvářet a upravovat tématické mapy, provádět výpočty, vytvářet databáze informací o geobjektech. Tímto výčet dovedností GRASSu zdaleka nekončí.

GRASS byl původně vyvíjen americkou armádou U.S. Army Corps of Engineer/CERL (Construction Engineering Research Lab) pro vojenské účely, náklady dosahovaly k několika milionům US dolarů. Koncem 80-tých let uvolnila armáda celý nástroj veřejnosti a to včetně všech zdrojových kódů. V roce 1995 ustoupil CERL z projektu a od roku 1997 převzal další vývoj GRASS Development Team na Baylor University v Texas, Universita Hannover v Německu a rovněž i další odborníci z celého světa. [2].

GRASS je implementován v jazyce ANSI-C a je plně vyhovující normě POSIX. Díky tomu ho můžeme provozovat na nemalé řadě architektur a operačních systémů (GNU/Linux, Solaris, Mac OS X, FreeBSD, iPAQ/Linux, MS-Windows . . .) [3]

#### 3.3.1 Technologie GIS GRASS

GRASS se skládá z velkého množství samostatných programů a modulů, obsahuje jich kolem 400. Většinu je možné ovládat přímo z příkazové řádky nebo využít grafického uživatelského rozhraní (dále GUI) napsaného pomocí Tcl/Tk. V současné době se pracuje na jeho renovaci založené na toolkitu wxPython 3.3.1.



Obrázek 3.2: Prototyp GUI pro GRASS využívající wxPython [6]



Pro práci s rastry se využívá knihovny GDAL 3.1.3, s vektory knihovny OGR 3.1.3 a s kartografickými projekcemi knihovny PROJ.4 3.1.3.

Data jsou uložena ve třech úrovních [7]:

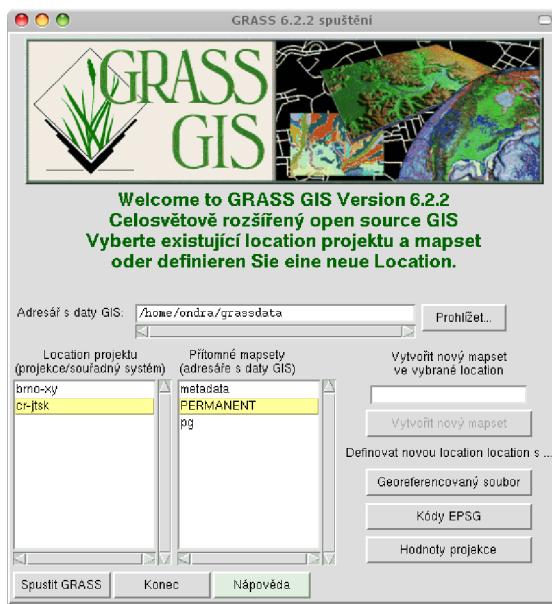
- DATABASE adresář obsahující všechna data
- LOCATION jméno lokace (uživatelé vytvořený nebo importovaný projekt definující souřadnicový systém, mapové zobrazení, referenční elipsoid . . .)
- MAPSET složka mapových listů, které může uživatel vytvářet a upravovat. Součástí každého mapsetu je tzv. mapset PERMANENT, který obsahuje originální mapové listy a je ho možné používat jenom pro čtení.

Uvedené úrovně jdou v adresářové struktuře hierarchicky za sebou DATABASE, LOCATION a jako poslední MAPSET.

Klasické uložení v adresářích je možné nahradit databází. Spolu s GIS GRASS je možné využívat databáze MySQL, PostgreSQL s rozšířením PostGIS 3.1.3, SQLite a ODBC ovladače.

### 3.3.2 Ukázka práce s GIS GRASS

Po spuštění je uživateli nabídnuto okno, kde si může vybrat požadovaný adresář s GIS daty (DATABASE), lokaci projektu (LOCATION), využít již existujícího MAPSETu či založit nový viz. obrázek 3.3.2. GRASS je dobré spouštět z příkazové řádky, abychom mohli využít všech příkazů a mít tak „absolutní“ kontrolu nad prováděnými úkony (pozn. popis příkladu je zaměřen na operační systém GNU/Linux).



Obrázek 3.3: Úvodní okno GIS GRASS

Pro vizualizaci prováděných operací s vektorovými a rastrovými daty se využívá nástroje zvaného *Monitor*. V jedné instanci GRASSu jich můžeme mít spuštěno maximálně sedm.

Pro vizualizaci 3D dat slouží nástroj *NVIZ*, který dokonce umožňuje simulovat průlet nad vybraným územím.

Spuštění monitoru se provádí příkazem

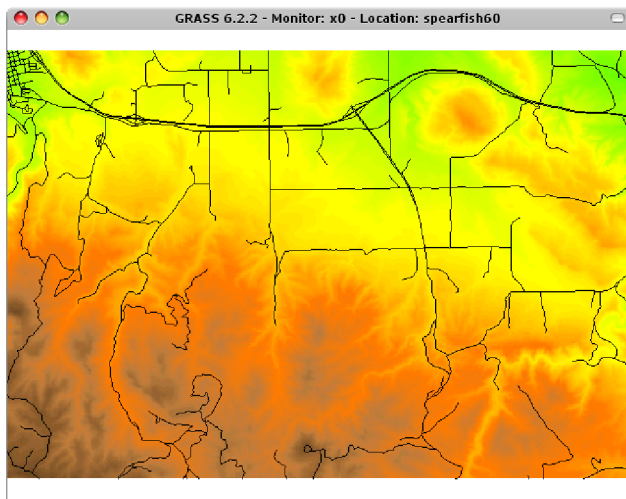
```
d.mon start=x0
```

Písmeno *d* vyznačuje zkrácený prefix pro slovo *display*. Většina programů sloužící pro vizualizaci grafických dat začíná tímto prefixem viz. 3.3.2 například pro vykreslení vektorové vrstvy se používá příkaz *d.vect* a rastrové vrstvy příkaz *d.rast*.

Následující příklad demonstruje jakým způsobem se provádí vykreslení do *Monitoru*. Vrstvy použité v příkladu jsou součástí lokace *Spearfish*, která je poskytnuta zdarma ke stažení na stránkách produktu. [3]

1. `d.mon start=x0` – spuštění monitoru *x0*
2. `d.rast elevation.10m` – vykreslení rastrové vrstvy *elevation.10m*
3. `d.rast d.vect roads` – vykreslení vektorové vrstvy *roads*
4. `d.mon stop=x0` – ukončení práce s monitorem

Výsledek ukazuje následující obrázek.



Obrázek 3.4: Výstup z monitoru

Jak už bylo zmíněno výše, využívá se v názvu většiny příkazů GIS GRASS tzv. prefixů (předpon). Ty symbolicky označují data a nástroje, se kterými pracují. [7]

<b>prefix</b>	<b>význam</b>	<b>popis skupiny příkazů</b>
g.	general	obecné příkazy
d.	display	příkazy pro práci s grafickými výstupy
r.	raster	příkazy pro práci s rastrovými daty
v.	vector	příkazy pro práci s vektorovými daty
db.	database	příkazy pro práci s databázemi
...		

Tabulka 3.1: Užívané prefixy v názvech programů GIS GRASS

## Kapitola 4

# Popis řešení a jeho implementace

Jak již bylo zmíněno v úvodu je cílem této bakalářské práce návrh a implementace řešení, které umožňuje zpřístupnit funkcionalitu desktopového GIS v síťovém prostředí (internetu). Konkrétně jde o implementaci rozhraní (jednoduchého serveru) schopného na jedné straně provádět operace GIS GRASS 3.3 a na straně druhé komunikovat s klientem pomocí standardu OGC WPS 2.1.

Výsledné řešení je implementováno jako CGI aplikace pracující pod webovým serverem Apache a nese název **Geoserver**, zkráceně **geoserv**. Aplikace je napsána v programovacím jazyce C/C++ a kromě standardních knihoven využívá také Open-Source knihovny GNU Cgicc (GNU GPL), libconfig (GNU LGPL) a xmlParser (BSD). Při návrhu byl brán ohled také na případná budoucí (rozšíření např. přidání podpory pro nový standard nebo GIS).

Klient je od koncového GIS GRASS v podstatě naprosto odstíněn a nepotřebuje znát jeho příkazy. Veškerá komunikace se serverem *geoserv* probíhá na základě specifikace standardu WPS viz. 2.1, která využívá formulace požadavku prostřednictvím URL nebo XML dokumentu, který je také primárně vrácen jako odpověď klientovi.

Na straně *geoserveru* jsou umístěny správcem definované procesy (operace), které může klient požadovat. V rámci těchto procesů je definováno jak bude požadavek na server proveden v GIS GRASS.

### 4.1 Definice nového procesu

S ohledem na to, aby bylo vytváření nových procesů co nejjednodušší, ale zároveň také umožňovalo využívat všech vymožeností vyššího programovacího jazyka, probíhá jeho definice rovněž v jazyce C/C++. Každý nový proces musí být definován jako potomek třídy **Process** umístěné v adresáři **wps** zdrojového archivu viz. 11.1. Tato třída obsahuje předdefinované metody usnadňující vytváření nových procesů a zároveň poskytuje programátorovi v podstatě volnou ruku, takže může využít i metody vlastní.

Přidávání nových procesů lze provádět přímo za běhu serveru. Jejich načítání probíhá dynamicky. V konfiguračním souboru *geoserveru* (ukázka viz. 10.1) musí být nastavena cesta, kde se mají požadované procesy vyhledávat a také názvy procesů, které mají být klientům zpřístupněny. Při požadavku na server je pak dynamicky načtena třída požadovaného procesu a provedena příslušná operace. Příklad definice nového procesu je uveden viz. 4.3.

## 4.2 Podporované žádosti standardu OGC WPS

### 4.2.1 WPS GetCapabilities

Při této žádosti *geoserver* nejprve provede načtení kontaktních informací o poskytovateli služby z konfiguračního souboru (10.1). Dále jsou postupně dynamicky načítány třídy procesů obsahující detailnější informace o každém z nich a nakonec je provedena formulace výstupního XML dokumentu, který je zaslán klientovi.

Příklad formulace žádosti klienta na server je uvedena viz. 9.1.1.

### 4.2.2 WPS DescribeProcess

Při zaregistrování toho požadavku se načte třída konkrétního procesu, jehož jméno je serveru předáno v rámci URL viz. 9.1.2. *Geoserver* provede vytvoření XML dokumentu popisujícího veškeré informace o daném procesu, zahrnující typy jeho vstupů a výstupů. Následně je XML dokument zaslán klientovi. Na základě informací získaných z tohoto požadavku již klient může provést formulaci obsahující konkrétní vstupní data a zaslat je ke zpracování žádostí typu *Execute*.

### 4.2.3 WPS Execute

K propojení s GIS GRASS dochází až v momentě tohoto požadavku. *Geoserveru* jsou ze strany klienta předána všechna data potřebná k provedení procesu (operace) a následně je také provedena kontrola jejich datových typů. Špatně zadaná hodnota datového typu vstupních dat vede k výjimce, jejíž popis je zahrnut ve standardu WPS. V tomto případě je klientovi vrácen XML dokument blíže specifikující výjimku, která nastala 9.1.3. Po úspěšné kontrole vstupních dat je vytvořen dočasný MAPSET v lokaci (LOCATION) definované procesem, případně provedeno vytvoření zcela nové lokace. V tomto bodě jsou již všechna vstupní data nastavena a zkontrolována, následuje tedy vykonání vlastní operace. Ta je definována virtuální metodou `execute()` přímo v rámci třídy obsahující vlastní proces 4.3. Metoda `execute()` má přístup ke všem předaným vstupním datům, které lze jednoduše získat metodou `getInput(string Identifier)`, kde argument `Identifier` je jednoznačný název vstupní hodnoty. Příkazy GIS GRASS je možné spouštět pomocí metody `Cmd(string cmd, string* resultStr)` definované ve třídě `Process`, kde argument `cmd` předpokládá zadání příkazu k provedení a skrze argument `resultStr` může být vrácen řetězec obsahující hlášení, které příkaz vytisknul na standardní či chybový výstup. Při úspěšném provedení celé operace je na základě definice výstupů daného procesu vytvořen XML dokument obsahující výsledek a zaslán klientovi.

Správnou kombinací vstupních parametrů (viz. 2.1.3) pro žádost *Execute* lze dosáhnout asynchronního chování procesu tzn., že klientovi je bezprostředně po obdržení požadavku zaslán XML dokument s odkazem, kde může očekávat výsledek. *Geoserver* vytvoří nový proces a také spustí vlákno, které testuje stav prováděného procesu a aktualizuje soubor obsahující odpověď.

Při zaznamenání jakékoliv chyby během provádění operace je proces ukončen a klientovi zaslán XML dokument s detailnějším popisem výjimky 9.1.3.

Příklad formulace požadavků a jejich možných výsledků je uveden viz. 9.1.3.

### 4.3 Příklad definice nového procesu

```
#include "Process.h"

class CzeLayers : public Process
{
public:
    CzeLayers() : Process()
    {
        Identifier = "czelayers";
        Title = "Export vektorové vrstvy z lokace CR-JTSK do GML";
        processVersion = "0.1";
        Abstract = "CR-JTSK";
        statusSupported = false;
        storeSupported = true;

        grassLocation = "/var/www/wps/cr-jtsk";

        // nastavení informací o procesu
        addMetadata("Export do GML");

        // přidání možnosti vstupu přímé hodnoty
        addLiteralInput("vect1", "Vektorová vrstva", "string");
        addLiteralInput("type", "Typ prvku", "string");
        addLiteralInput("outputName", "Název exportu", "string");

        // definice typu výsledku
        addComplexReferenceOutput("czechOutOGR", "OGR výstup", "text/xml");
    }
    virtual int execute() {
        try {
            string outputFilename = getInput("outputName")+".gml";
            setStatus("Executing v.out.ogr",50);
            Cmd("v.out.ogr input="+getInput("vect1")
                +" type="+getInput("type")
                +" dsn="+outputFilename
                +" olayer="+getInput("outputName")
                +" format=GML");
            setOutput("czechOutOGR", outputFilename);

        } catch(CmdFailed &e) {
            return(-1);
        }

        return 0;
    }
    virtual ~CzeLayers(){};
};
```

```

/* tyto řádky musí obsahovat každý nově definovaný proces
 * aby bylo možné jeho dynamické spouštění
 */
extern "C" Process* create() {
    return new CzeLayers;
}
extern "C" void destroy(Process* p) {
    delete p;
}

```

## 4.4 Testování Geoserveru

Funkčnost řešení byla otestována na operačních systémech GNU/Linux (distribuce Gentoo a Debian) a FreeBSD. Testy byly provedeny na jejich lokálním serveru Apache.

V rámci testování nástroje OpenLayers bylo odzkoušeno více internetových prohlížečů a provedeno srovnání úspěšnosti WPS žádostí mezi *Geoserverem* a implementací PyWPS viz. tabulka 4.1.

Prohlížeč		Geoserver			PyWPS		
název	verze	GC	DP	E	GC	DP	E
Firefox	2.0.13	ok	ok	ok	ok	ok	ok
SeaMonkey	1.1.9	ok	ok	ok	ok	ok	ok
Epiphany	2.20.3	ok	ok	ok	ok	ok	ok
Opera	9.27	ok	ok	x	ok	ok	x
Safari	3.1.1	ok	ok	x	ok	ok	x
IE	7.0	x	x	x	x	x	x

Tabulka 4.1: Test podpory nástroje OpenLayers v internetových prohlížečích

Význam uvedených zkratk v tabulce 4.1 : GC znamená žádost GetCapabilities, DP žádost DescribeProcess a E žádost Execute. Z tabulky je patrné, že prohlížeče založené na jádru *Gecko* nemají se zpracováním WPS požadavku prostřednictvím knihovny OpenLayers problém, avšak modul WPS je zatím ve fázi vývoje a do stabilní verze ještě zatím nebyl začleněn. Test zátěže *Geoserveru* byl vyzkoušen na malé domácí síti skládající se z pěti osobních počítačů. Při poskytování ukázkových procesů viz. kapitola 5, nebylo zjištěno žádných zásadních problémů. S větším počtem požadavků mírně rostla doba jejich zpracování. Test zátěže na síti Internet nemohl být proveden, protože nejsem vlastníkem ani provozovatelem žádného internetového serveru.

## Kapitola 5

# Demonstrace řešení na jednoduché demoaplikaci

Pro účel demonstrace a ověření funkčnosti implementace standardu WPS byla vytvořena jednoduchá webová stránka obsahující okno JavaScriptové knihovny OpenLayers (viz. 3.2), do které byla z vývojové větve projektu přidána podpora pro standard WPS. Na serverové části byly vytvořeny 2 ukázkové procesy:

### 1. Nalezení nejkratší cesty

Proces demonstruje použití příkazu `v.net.path` GIS GRASS. Na vstupu očekává parametry typu `float` – počáteční a koncové souřadnice bodů, mezi kterými se má provést nalezení cesty a paramter typu `string` obsahující jméno výsledné vrstvy, která bude obsahovat nalezenou cestu.

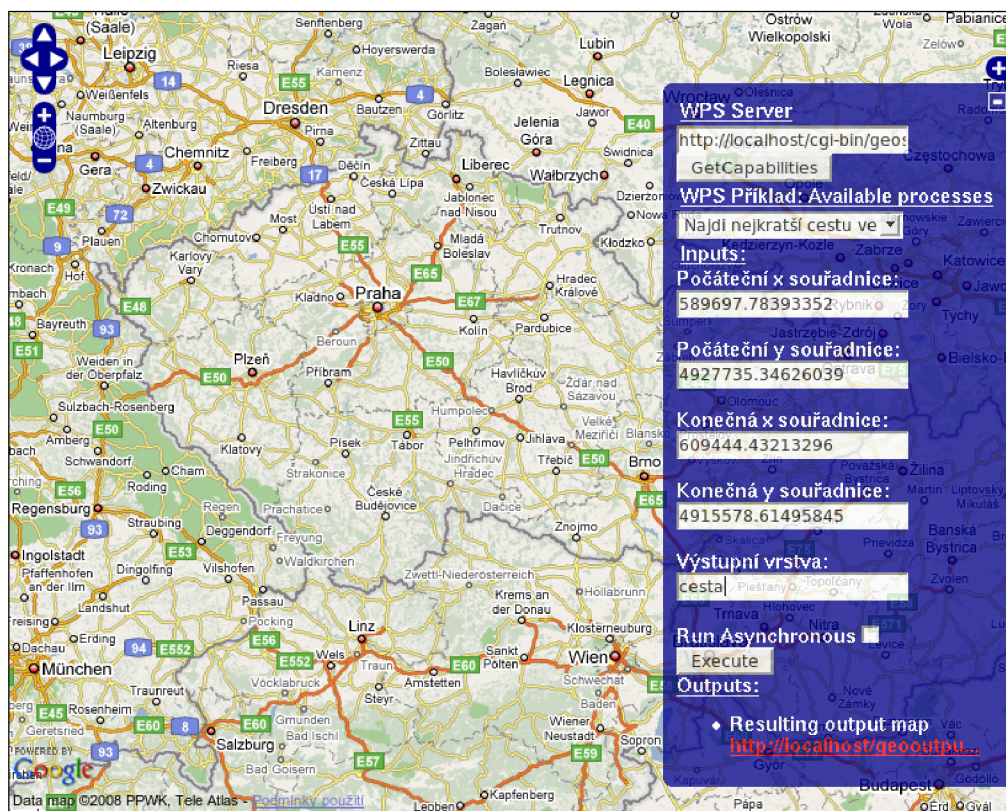
### 2. Export vektorové vrstvy do formátu typu GML

Proces pro své spuštění vyžaduje 3 vstupní parametry typu `string` (název vektorové vrstvy určené k exportu, typ vrstvy a jméno výsledného GML souboru).

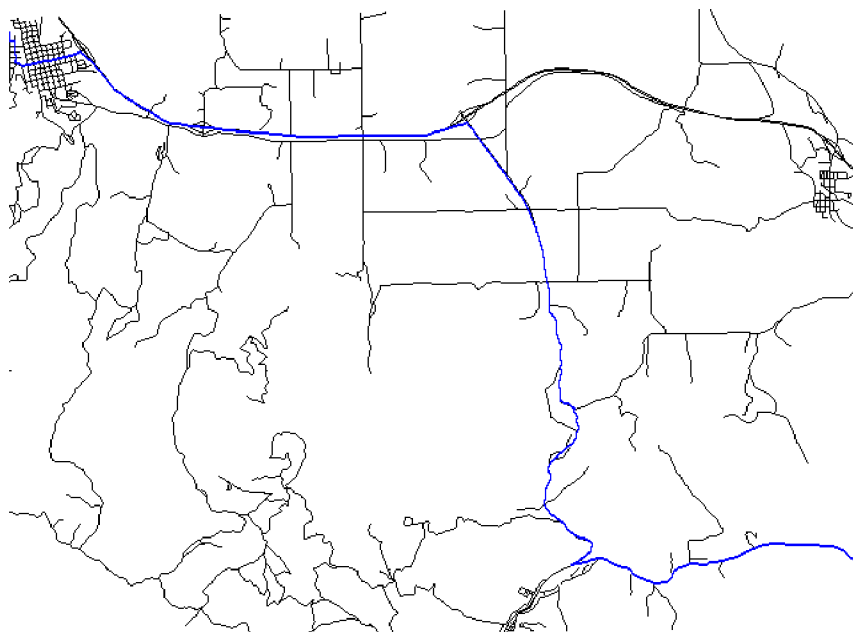
## 5.1 Příklad ovládání procesu pomocí nástroje OpenLayers

Výběr služby WPS provedeme kliknutím na druhý křížek v oknu OpenLayers. Následuje kliknutí na tlačítko `GetCapabilities`, kterým získáme ze serveru seznam podporovaných procesů. Vybereme například proces nazvaný jako „*Najdi nejkratší cestu ve Spearfish lokaci*“. Do textových políček zadáme požadované údaje (`x` a `y` souřadnice počátečního a koncového bodu a název výstupní vrstvy). Dále stiskneme tlačítko `Execute`, které vytvoří formulaci požadavku na základě námi zadaných hodnot a pošle ji *geoserveru* ke zpracování. Pokud proces proběhl úspěšně, bude výsledek vypadat podobně jako na obrázku 5.1. Pod tlačítkem `Execute` je nám nabídnut ke stažení soubor, který obsahuje výsledek hledané cesty viz. obrázek 5.2. Nastala-li v provádění operace chyba, zobrazí klient OpenLayers jeho popis viz. obrázek 5.3.

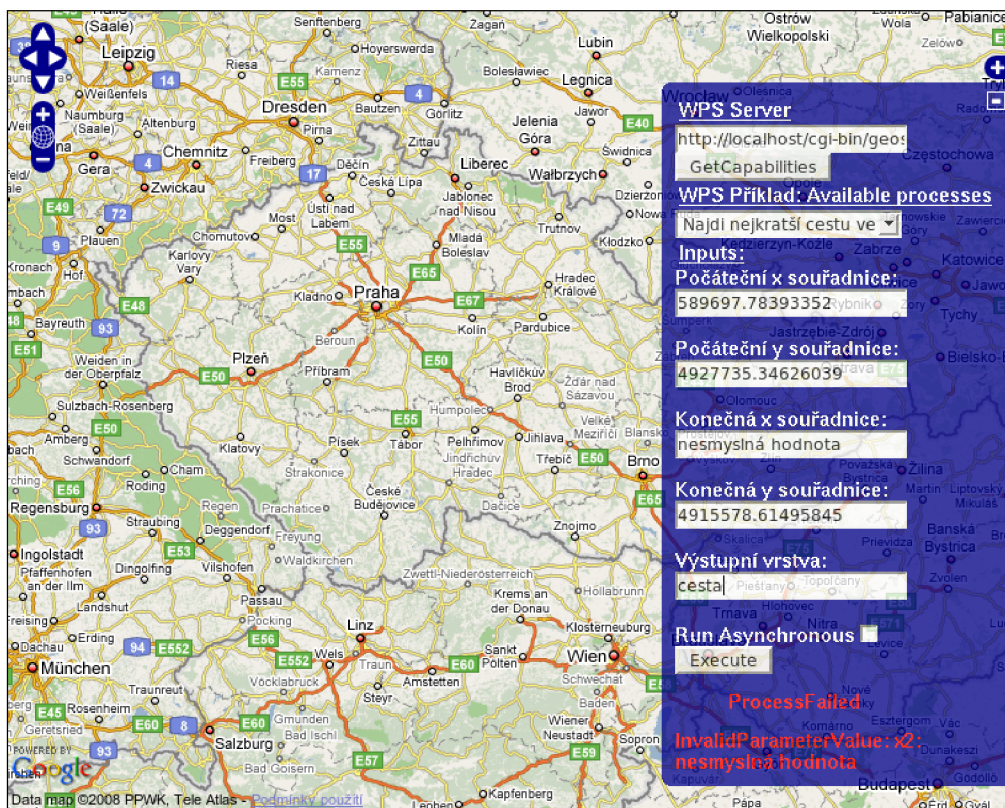




Obrázek 5.1: Výsledek procesu v podání knihovny OpenLayers



Obrázek 5.2: Úspěšné nalezení nejkratší cesty



Obrázek 5.3: Provedení operace se nezdařilo (chybný vstupní parametr)

# Kapitola 6

## Popis instalace

Instalace *Geoserveru* byla úspěšně vyzkoušena na operačních systémech Gentoo Linux, Debian a FreeBSD. Následující popis instalace předpokládá platformu x86 s nainstalovaným operačním systémem Gentoo Linux. Instalace na ostatních linuxových distribucích bude však obdobná, jen se budou lišit nástroje pro instalaci balíčků a případně i názvy balíčků.

V Gentoo probíhá instalace přímo kompilováním ze zdrojových kódů. V popisu uvádím nastavení některých dodatečných USE flags <sup>1</sup>, které umožní využívat co nejvíce zajímavých vlastností instalovaných nástrojů.

### 6.1 Instalace nástrojů

Aby bylo možné na našem počítači provozovat *Geoserver*, je nutná instalace dalších nástrojů a knihoven, kterých *Geoserver* využívá.

#### 6.1.1 GIS GRASS

Pro podporu grafického uživatelského rozhraní je nutné provést následující aktivaci USE flags.

```
echo "sci-geosciences/grass tcl tk motif" >> /etc/portage/package.use
```

Dalším příkazem se provede kompilace zdrojového kódu a jeho instalace.

```
emerge -av grass
```

#### 6.1.2 Webový server Apache

V tomto případě není pro webový server nutné aktivovat dalších USE flags.

```
emerge -av apache
```

#### 6.1.3 Knihovny

Pro překlad *Geoserveru* je nutné mít v systému přítomny knihovny GNU Cgicc a libconfig. Jejich instalace je provedena následujícím příkazem.

```
emerge -av cgicc libconfig
```

---

<sup>1</sup>USE flag: obdoba atributů, které se před kompilací předávají skriptu `configure`

## 6.2 Instalace Geoserveru

Všechny potřebné nástroje jsou již nainstalovány a tak můžeme přistoupit k vlastní instalaci *Geoserveru*. Nejprve je nutné v souboru `Makefile` nastavit cestu k adresáři `cgi-bin`, kde jsou umístěny programy, jejichž spuštění server Apache umožňuje. To se provádí nastavením proměnné `CGI_BIN_PATH` na cestu ke zmíněnému adresáři. V Gentoo může být nastavení následující `CGI_BIN_PATH = /var/www/localhost/cgi-bin`.

Příkazem `make` provedeme překlad zdrojových kódů a příkazem `make install` instalaci *Geoserveru* včetně ukázkových procesů. Nakonec provedeme modifikaci konfiguračního souboru `settings.conf` (viz. 10.1), který by měl být po instalaci umístěn v adresáři `/etc/geoserv/`.

## 6.3 Uvedení demoaplikace do provozu

Demoaplikace včetně nástroje OpenLayers je umístěna na přiloženém CD v archivu `demo.tar.gz`. Její instalaci provedeme rozbalením tohoto archivu do kořenového adresáře serveru Apache. V Gentoo je cesta k tomuto adresáři `/var/www/localhost/htdocs`.

K vyzkoušení ukázkových procesů potřebujeme také data pro GIS GRASS. Konkrétně se jedná o lokace Spearfish viz. [4] a CR-JTSK viz. [5], které jsou také umístěny na přiloženém CD v adresáři `geodata`. Archivy lokací doporučuji rozbalit do adresáře `/var/www/wps/`.

# Kapitola 7

## Závěr

Cílem této bakalářské práce bylo vytvoření jednoduchého serveru umožňujícího napojení desktopového GIS GRASS prostřednictvím služby WPS na intranet/internet. Požadavkem bylo, aby klient se serverem komunikoval pomocí standardu OGC WPS. Tento požadavek byl v rámci řešeného projektu splněn. Konečné řešení je nastaveno tak, aby bylo možné přidávat podporu pro další služby či standardy. Při procházení internetu nebylo dosud objeveno jiné otevřené řešení, které by tento problém implementovalo v jazyce C/C++. Současný stav aplikace je vhodný spíše pro účely testování. Pro nasazení do skutečného provozu by bylo nutné aplikaci ještě dále upravovat.

Projekt by se měl ubírat směrem přidávání podpory pro další standardy OGC, zejména implementaci standardu WMS (Web Map Service).

# Literatura

- [1] GDAL: GDAL Raster Formats. [online], cit [2008-05-03].  
URL [http://www.gdal.org/formats\\_list.html](http://www.gdal.org/formats_list.html)
- [2] GISAK: GRASS. WWW stránky linuxové live CD distribuce GISák [online], cit [2008-04-15].  
URL <http://geo2.fsv.cvut.cz/wms/index.php>
- [3] GRASS: GRASS Features. [online], GRASS Development Team, cit [2008-04-15].  
URL <http://grass.itc.it/>
- [4] GRASS: GRASS locations (data sets). [online], GRASS Development Team, cit [2008-04-15].  
URL <http://grass.itc.it/download/data.php>
- [5] GRASSwikiCZ: FreeGeodataCZ. [online], cit [2008-04-15].  
URL <http://grass.fsv.cvut.cz/wiki/index.php/FreeGeodataCZ>
- [6] GRASSwikiCZ: GRASS. [online], cit [2008-04-10].  
URL <http://grass.fsv.cvut.cz/wiki/index.php/GRASSwikiCZ>
- [7] Hrubý, M.: Texty k přednáškám pro předmět Geografické Informační Systémy. [online], cit [2008-05-03].  
URL <http://perchta.fit.vutbr.cz:8000/vyuka-gis>
- [8] OGC: OpenGIS Web Map Server Implementation Specification. [online], OGC 06-042, Open Geospatial Consortium Inc., cit [2008-05-03].  
URL <http://www.opengeospatial.org/standards/wps>
- [9] OGC: OpenGIS Web Processing Service. [online], OGC 05-007r7, Open Geospatial Consortium Inc., cit [2008-05-03].  
URL <http://www.opengeospatial.org/standards/wms>
- [10] OGC: Web Map Context Documents. [online], OGC 05-005, Open Geospatial Consortium Inc., cit [2008-05-03].  
URL <http://www.opengeospatial.org/standards/wmc>
- [11] OGR: OGR Vector Formats. [online], cit [2008-05-03].  
URL [http://www.gdal.org/ogr/ogr\\_formats.html](http://www.gdal.org/ogr/ogr_formats.html)
- [12] OpenLayers: OpenLayers 2.6 Release Notes. [online], cit [2008-05-03].  
URL <http://trac.openlayers.org/wiki/Release/2.6/Notes>

- [13] PROJ.4: Cartographic Projections Library. [online], cit [2008-05-03].  
URL <http://proj.maptools.org/>
- [14] PyWPS: Implementation of OGC Web Processing Service standard. [online], PyWPS development team, cit [2008-05-03].  
URL <http://pywps.wald.intevation.org/>
- [15] UMN: UMN Mapserver. [online], University of Minnesota, cit [2008-05-03].  
URL <http://mapserver.gis.umn.edu/>
- [16] Wikipedie: Base64. [online], cit [2008-05-03].  
URL <http://cs.wikipedia.org/wiki/Base64>
- [17] Wikipedie: Geography Markup Language. [online], cit [2008-05-03].  
URL [http://en.wikipedia.org/wiki/Geography\\_Markup\\_Language](http://en.wikipedia.org/wiki/Geography_Markup_Language)
- [18] ČVUT: WWW stránky. [online], cit [2008-04-14].  
URL <http://geo2.fsv.cvut.cz/wms/index.php>

## Kapitola 8

# Seznam příloh

- A CD se zdrojovými kódy
- B Ukázky požadavků a XML výstupů
- C Ukázky konfiguračních souborů Geoserveru a Mapserveru
- D Adresářová struktura zdrojového kódu



## Kapitola 9

# Příloha B

### 9.1 Ukázky požadavků a XML výstupů

#### 9.1.1 WPS žádost GetCapabilities

##### Ukázka formulace URL požadavku

```
http://localhost/cgi-bin/geoserv.cgi?service=WPS&request=GetCapabilities
```

##### Výsledný XML dokument, který obdrží klient od serveru

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-US"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  ../wpsGetCapabilities_response.xsd">
  <ows:ServiceIdentification>
    <ows:Title>WPS Příklad</ows:Title>
    <ows:Abstract>Příklad použití WPS spolu s nástrojem GRASS</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>GIS</ows:Keyword><ows:Keyword>WPS</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>0.5.0</ows:ServiceTypeVersion>
    <!-- Element Fees is not set -->
    <ows:AccessConstraints>Unset</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>
      VUT v Brně, Fakulta informačních technologií
    </ows:ProviderName>
    <ows:ServiceContact>
      <ows:IndividualName>Ondřej Kučera</ows:IndividualName>
      <ows:PositionName>student</ows:PositionName>
    </ows:ServiceContact>
  </ows:ServiceProvider>
</wps:Capabilities>
```

```

    <ows:ContactInfo>
      <ows:Address>
        <ows:DeliveryPoint>Božetěchova 2</ows:DeliveryPoint>
        <ows:City>Brno</ows:City>
        <ows:PostalCode>612 66</ows:PostalCode>
        <ows:Country>Česká republika</ows:Country>
        <ows:ElectronicMailAddress>
          xkucer51@stud.fit.vutbr.cz
        </ows:ElectronicMailAddress>
      </ows:Address>
    </ows:ContactInfo>
    <ows:Role>WPS administrátor</ows:Role>
  </ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationMetadata>
  <ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost/cgi-bin/geoserv.cgi?"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeProcess">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost/cgi-bin/geoserv.cgi?"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="Execute">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost/cgi-bin/geoserv.cgi?"/>
        <ows:Post xlink:href="http://localhost/cgi-bin/geoserv.cgi?"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
</ows:OperationMetadata>
<wps:ProcessOfferings>
  <wps:Process wps:processVersion="0.1">
    <ows:Identifier>spearpath</ows:Identifier>
    <ows:Title>Najdi nejkratší cestu ve Spearfish lokaci</ows:Title>
    <ows:Abstract>SpearPath</ows:Abstract>
  </wps:Process>
  <wps:Process wps:processVersion="0.1">
    <ows:Identifier>czelayers</ows:Identifier>
    <ows:Title>Export vektorové vrstvy z lokace CR-JTSK do GML</ows:Title>
    <ows:Abstract>CR-JTSK</ows:Abstract>
  </wps:Process>

```

```

        <ows:Metadata xlink:title="Export do GML"/>
    </wps:Process>
</wps:ProcessOfferings>
<wps:Languages>
    <wps:Default>
        <ows:Language>en-US</ows:Language>
    </wps:Default>
    <wps:Supported>
        <ows:Language>en-US</ows:Language>
        <ows:Language>cs-CZ</ows:Language>
    </wps:Supported>
</wps:Languages>
</wps:Capabilities>

```

### 9.1.2 WPS žádost DescribeProcess

#### Ukázka formulace URL požadavku

```

http://localhost/cgi-bin/geoserv.cgi?
service=WPS&request=DescribeProcess&identifier=czelayers

```

#### Výsledný XML dokument, který obdrží klient od serveru

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions service="WPS" version="1.0.0" xml:lang="cs-CZ"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
../wpsGetCapabilities_response.xsd">
  <ProcessDescription wps:processVersion="0.1"
storeSupported="true" statusSupported="false">
    <ows:Identifier>czelayers</ows:Identifier>
    <ows:Title>Export vektorové vrstvy z lokace CR-JTSK do GML</ows:Title>
    <ows:Abstract>CR-JTSK</ows:Abstract>
    <ows:Metadata xlink:title="Export do GML"/>
    <wps:Profile>OGC:WPS:GML</wps:Profile>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>vect1</ows:Identifier>
        <ows:Title>Vektorová vrstva</ows:Title>
        <ows:Abstract/>
        <LiteralData>
          <ows:DataType
ows:reference="http://www.w3.org/TR/xmlschema-2/#string">
            string
          </ows:DataType>
        <UOMs>

```

```

        <Default>
            <ows:UOM>m</ows:UOM>
        </Default>
        <Supported>
            <ows:UOM>m</ows:UOM>
        </Supported>
    </UOMs>
    <ows:AnyValue/>
</LiteralData>
</Input>
<Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>type</ows:Identifier>
    <ows:Title>Typ prvku</ows:Title>
    <ows:Abstract/>
    <LiteralData>
        <ows:DataType
            ows:reference="http://www.w3.org/TR/xmlschema-2/#string">
            string
        </ows:DataType>
        <UOMs>
            <Default>
                <ows:UOM>m</ows:UOM>
            </Default>
            <Supported>
                <ows:UOM>m</ows:UOM>
            </Supported>
        </UOMs>
        <ows:AnyValue/>
    </LiteralData>
</Input>
<Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>outputName</ows:Identifier>
    <ows:Title>Název exportu</ows:Title>
    <ows:Abstract/>
    <LiteralData>
        <ows:DataType
            ows:reference="http://www.w3.org/TR/xmlschema-2/#string">
            string
        </ows:DataType>
        <UOMs>
            <Default>
                <ows:UOM>m</ows:UOM>
            </Default>
            <Supported>
                <ows:UOM>m</ows:UOM>
            </Supported>
        </UOMs>
        <ows:AnyValue/>
    </LiteralData>
</Input>

```

```

        </LiteralData>
    </Input>
</DataInputs>
<ProcessOutputs>
    <Output>
        <ows:Identifier>czechOutOGR</ows:Identifier>
        <ows:Title>OGR výstup</ows:Title>
        <ows:Abstract/>
        <ComplexOutput>
            <Default>
                <Format>
                    <MimeType>text/xml</MimeType>
                </Format>
            </Default>
            <Supported>
                <Format>
                    <MimeType>text/xml</MimeType>
                </Format>
            </Supported>
        </ComplexOutput>
    </Output>
</ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>

```

### 9.1.3 WPS žádost Execute

#### Ukázka formulace URL požadavku

```

http://localhost/cgi-bin/geoserv.cgi?service=WPS&request=Execute
&identifier=czelayers
&datainputs=vect1,reky,type,line,outputName,testogr
&status=true&storeExecuteResponse=true

```

#### Ukázka formulace XML požadavku

```

<wps>DataInputs>
    <wps:Input>
        <ows:Identifier>x1</ows:Identifier>
        <wps:LiteralValue>589697.78393352</wps:LiteralValue>
    </wps:Input>
    <wps:Input>
        <ows:Identifier>y1</ows:Identifier>
        <wps:LiteralValue>4927735.34626039</wps:LiteralValue>
    </wps:Input>
    <wps:Input>
        <ows:Identifier>x2</ows:Identifier>
        <wps:LiteralValue>609444.43213296</wps:LiteralValue>
    </wps:Input>

```

```

    <wps:Input>
      <ows:Identifier>y2</ows:Identifier>
      <wps:LiteralValue>4915578.61495845</wps:LiteralValue>
    </wps:Input>
    <wps:Input>
      <ows:Identifier >outputLayer</ows:Identifier>
      <wps:LiteralValue>cesta</wps:LiteralValue>
    </wps:Input>
  </wps>DataInputs>

```

### Výsledný XML dokument, který obdrží klient od serveru

```

<?xml version="1.0" encoding="UTF-8"?>
<ows:ExecuteResponse service="WPS"
  version="1.0.0" xml:lang="en-US"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  ../wpsExecute_response.xsd">
  <wps:Process wps:processVersion="0.1">
    <ows:Identifier>czelayers</ows:Identifier>
    <ows:Title>Export vektorové vrstvy z lokace CR-JTSK do GML</ows:Title>
    <ows:Abstract>CR-JTSK</ows:Abstract>
    <ows:Metadata xlink:title="Export do GML"/>
    <wps:Profile>OGC:WPS:GML</wps:Profile>
  </wps:Process>
  <wps:Status creationTime="2008-05-11T18-25-26Z">
    <wps:ProcessSucceeded/>
  </wps:Status>
  <ProcessOutputs>
    <Output>
      <ows:Identifier>czechOutOGR</ows:Identifier>
      <ows:Title>OGR výstup</ows:Title>
      <ows:Abstract/>
      <ComplexValueReference format="text/xml"
        ows:reference="http://localhost/geooutputs/2008-5-11-18-25-24-testogr.g
      </Output>
    </ProcessOutputs>
  </ows:ExecuteResponse>

```

### XML výstup vyjímek, které mohou nastat při vykonávání procesu

```

<?xml version="1.0" encoding="UTF-8"?>
<ExceptionReport version="1.0.0" xmlns="http://www.opengis.net/ows"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Exception exceptionCode="MissingParameterValue" locator="x1"/>
</ExceptionReport>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<ows:ExecuteResponse service="WPS" version="1.0.0"
  xml:lang="en-US"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  ../wpsExecute_response.xsd">
  <wps:Process wps:processVersion="0.1">
    <ows:Identifier>spearpath</ows:Identifier>
    <ows:Title>Najdi najkratši cestu ve Spearfish lokaci</ows:Title>
    <ows:Abstract>SpearPath</ows:Abstract></wps:Process>
    <wps:Status creationTime="2008-05-11T18-41-04Z">
      <wps:ProcessFailed>
        <ExceptionReport version="1.0.0"
          xmlns="http://www.opengis.net/ows"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <Exception exceptionCode="NoApplicableCode">
            <ows:ExceptionText>Render vector roads layer</ows:ExceptionText>
          </Exception>
        </ExceptionReport>
      </wps:ProcessFailed>
    </wps:Status>
  </ows:ExecuteResponse>

```

# Kapitola 10

## Příloha C

### 10.1 Ukázka konfiguračního souboru Geoserveru

```
# WPS configuration

WPS:
{

    version = "1.0.0";

    Keywords = ["GIS","WPS"];

    ServiceIdentification:
    {
        Title = "WPS Příklad";
        Abstract = "Příklad použití WPS spolu s nástrojem GRASS";
        ServiceType = "WPS";
        ServiceTypeVersion = "0.5.0";
        AccessConstraints = "Unset";
    };

    ServiceProvider:
    {
        ProviderName = "VUT v Brně, Fakulta informačních technologií";
        IndividualName = "Ondřej Kučera";
        PositionName = "student";
        Role = "WPS administrátor";
        DeliveryPoint = "Božetěchova 2";
        City = "Brno";
        PostalCode = "612 66";
        Country = "Česká republika";
        ElectronicMailAddress = "xkucer51@stud.fit.vutbr.cz";
    };

    OperationsMetadata:
    {
```



```

        ServerAddress = "http://localhost/cgi-bin/geoserv.cgi";
    };

    # Languages - first language is default
    Languages = ["en-US","cs-CZ"];

    Processes = ["spearpath","czelayers"];
};

#Server configuration

ServerConfiguration:
{
    OutputDir = "/var/www/localhost/htdocs/geooutputs";
    OutputUrl = "http://localhost/geooutputs";

    TmpPath = "/tmp";

    ProcessesPath = "/var/www/localhost/cgi-bin/processes";
};

#GRASS configuration

GRASS:
{
    PATH = "/usr/grass-6.2.2/bin/:/usr/grass-6.2.2/scripts/:/usr/bin:/bin/";
    GRASS_VERSION = "6.2.2";
    GRASS_GUI = "text";
    GISBASE = "/usr/grass-6.2.2";
    LD_LIBRARY_PATH = "/usr/grass-6.2.2/lib";
    GRASS_PERL = "/usr/bin/perl";
};

```

## 10.2 Ukázka konfiguračního souboru Mapfile pro Mapserver

```

MAP
    NAME czech
    # Velikost výsledné mapy v pixelech
    SIZE 600 400
    # Jednotky vzdálenosti
    UNITS meters

    # Hraniční oblast
    EXTENT -904222.837576 -1229939.629021 -431433.332168 -932539.778845

    # Geografické zobrazení
    PROJECTION
        'proj=longlat'

```

```

    'ellps=WGS84'
    'datum=WGS84'
    'no_defs'
END

# Barva pozadí
IMAGECOLOR 192 192 192
IMAGEQUALITY 95
IMAGETYPE png

# Výběr grafického ovladače
OUTPUTFORMAT
    NAME png
    DRIVER 'GD/PNG'
    MIMETYPE 'image/png'
    #IMAGEMODE PC256
    EXTENSION 'png'
END

# Nastavení legendy
LEGEND
    IMAGECOLOR 255 255 255
    STATUS ON
    KEYSIZE 18 12
    LABEL
        TYPE BITMAP
        SIZE MEDIUM
        COLOR 0 0 89
    END
END

WEB
# Nastavení cesty kam může Mapserver ukládat výsledné soubory (mapy)
IMAGEPATH '/tmp/'

IMAGEURL '/tmp/'

# Nastavení služby WMS
METADATA
    'wms_title'          'czech'
    'wms_onlineresource' 'http://localhost/cgi-bin/mapserv?map=czech.map&'
    'wms_srs'            'EPSG:4326'
END

END

# Vektorová vrstva hranic ČR

```

```

LAYER
  NAME 'cr'
  TYPE POLYGON
  DATA 'data/cr.shp'
  METADATA
    'wms_title' 'cr'
  END
  STATUS DEFAULT
  TRANSPARENCY 100
  PROJECTION
    'proj=longlat'
    'ellps=WGS84'
    'datum=WGS84'
    'no_defs'
  END
  CLASS
    NAME 'cr'
    STYLE
      SYMBOL 0
      SIZE 6
      OUTLINECOLOR 0 0 0
      COLOR 22 50 170
    END
  END
END

# Vektorová vrstva okresů ČR
LAYER
  NAME 'okresy_pseudo'
  TYPE POLYGON
  DATA '/var/www/localhost/htdocs/mapserver/data/okresy_pseudo.shp'
  METADATA
    'wms_title' 'okresy_pseudo'
  END
  STATUS DEFAULT
  TRANSPARENCY 100
  PROJECTION
    'proj=longlat'
    'ellps=WGS84'
    'datum=WGS84'
    'no_defs'
  END
  CLASS
    NAME 'okresy_pseudo'
    STYLE
      SYMBOL 0
      SIZE 6
      OUTLINECOLOR 0 0 0

```

COLOR 0 85 255

END

END

END

END

# Kapitola 11

## Příloha D

### 11.1 Adresářová struktura zdrojového kódu

```
geoserv
|-- ogc
|   '-- OgcConstants.h
|-- wps
|   |-- CzeLayers.cpp
|   |-- DescribeProcess.cpp
|   |-- DescribeProcess.h
|   |-- Execute.cpp
|   |-- Execute.h
|   |-- GetCapabilities.cpp
|   |-- GetCapabilities.h
|   |-- Process.cpp
|   |-- Process.h
|   |-- Spearpath.cpp
|   |-- Spearpath.h
|   |-- Wps.cpp
|   |-- Wps.h
|   |-- exceptions.cpp
|   |-- exceptions.h
|   |-- processLoader.cpp
|   '-- processLoader.h
|-- xml
|   |-- xmlParser.cpp
|   '-- xmlParser.h
|-- Configuration.cpp
|-- Configuration.h
|-- Geoserver.cpp
|-- Geoserver.h
|-- Grass.cpp
|-- Grass.h
|-- Makefile
|-- Makefile.debian
|-- constants.h
```

```
|-- convert.h  
|-- errors.cpp  
|-- errors.h  
|-- geoserv.cpp  
|-- settings.conf  
'-- settings.conf.default
```

3 directories, 35 files