

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Aplikace pro podporu výuky NAT a NAT Traversal



2022

Jakub Mazur

Vedoucí práce:  
doc. Mgr. Jan Outrata, Ph.D.

Studijní program: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Jakub Mazur  
Název práce: Aplikace pro podporu výuky NAT a NAT Traversal  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2022  
Studijní program: Aplikovaná informatika, prezenční forma  
Vedoucí práce: doc. Mgr. Jan Outrata, Ph.D.  
Počet stran: 34  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Jakub Mazur  
Title: Application to support teaching of NAT and NAT Traversal  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2022  
Study program: Applied Computer Science, full-time form  
Supervisor: doc. Mgr. Jan Outrata, Ph.D.  
Page count: 34  
Supplements: 1 CD/DVD  
Thesis language: Czech

## **Anotace**

*Obsahem práce je přehled aktuálních metod obcházení překladu síťových adres, tzv. NAT Traversal a demonstrace vybraných metod v praktické části. Přesněji metod Port Mapping a NAT Hole Punching, které se využívají k*

## **Synopsis**

*The content of the thesis is*

**Klíčová slova:** NAT; NAT Traversal; Port Mapping; NAT Hole Punching

**Keywords:** NAT; NAT Traversal; Port Mapping; NAT Hole Punching

Děkuji panu doc. Janu Outratovi, Ph. D. za vedení, čas strávený u konzultací a objektivní rady při tvorbě práce.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>NAT</b>	<b>8</b>
2.1	Využití . . . . .	8
2.2	Překlad adres . . . . .	9
2.3	Druhy přiřazování adres . . . . .	9
2.3.1	Statický NAT . . . . .	9
2.3.2	Dynamický NAT (tzv. Maškaráda) . . . . .	10
2.3.3	Přetížený NAT (NAT overloading, Port Address Translation)	10
2.4	Druhy implementací NATu . . . . .	11
2.4.1	Full cone NAT (1:1 NAT) . . . . .	11
2.4.2	Restricted cone NAT . . . . .	12
2.4.3	Port restricted cone NAT . . . . .	12
2.4.4	Symetrický NAT . . . . .	12
<b>3</b>	<b>Metody NAT Traversal</b>	<b>13</b>
3.1	Traversal Using Relays around NAT (TURN) . . . . .	13
3.2	Reverse connection . . . . .	14
3.3	NAT hole punching . . . . .	14
3.3.1	UDP Hole Punching . . . . .	14
3.3.2	TCP Hole Punching . . . . .	16
3.4	Port Forwarding . . . . .	17
3.5	Přepínače . . . . .	18
3.6	Geometrie stránky . . . . .	20
<b>4</b>	<b>Sazba částí dokumentu</b>	<b>21</b>
4.1	Sazba úvodní strany či obsahu . . . . .	21
4.2	Závěry . . . . .	21
4.3	Matematika . . . . .	21
4.4	Sazba literatury . . . . .	22
4.4.1	Sazba bibliografie přes BIB <sub>Λ</sub> T <sub>Ε</sub> X . . . . .	22
4.4.2	Manuální sazba bibliografie . . . . .	22
4.5	Drobná makra . . . . .	22
4.6	Sazba rejstříku . . . . .	22
4.7	Sazba zdrojových kódů . . . . .	24
	<b>Závěr</b>	<b>27</b>
	<b>Conclusions</b>	<b>28</b>
<b>A</b>	<b>První příloha</b>	<b>29</b>
<b>B</b>	<b>Druhá příloha</b>	<b>29</b>

C Obsah přiloženého datového média	29
Seznam zkratek	31
Literatura	32
Rejstřík	34

## Seznam obrázků

1	Překlad adres ve statickém NATu . . . . .	9
2	Full cone NAT[3] . . . . .	11
3	Restricted cone NAT[3] . . . . .	12
4	Symetrický NAT[3] . . . . .	13
5	Obcházení NATu pomocí metody reverse connection . . . . .	14
6	Obcházení NATu pomocí metody UDP Hole Punching . . . . .	15

## Seznam tabulek

1	Seznam přepínačů . . . . .	18
2	Odstavce v tabulkách . . . . .	23

# 1 Úvod

Každé zařízení, které se chce připojit k internetu potřebuje svou IP adresu. IP adresa může nabývat pouze omezených hodnot, a proto se pomalu vyčerpávají. Tento problém nebyl zřejmý na začátku internetu, ale projevil se až později.

Řešením je překlad síťových adres (Network Address Translation, dále pouze jako NAT). NAT umožňuje pro více zařízení v privátní síti přístup k internetu pomocí jedné veřejné IP adresy. Protože jsou zařízení za NATem, tak nemají přímé připojení k Internetu a je složité navázat přímé spojení s jiným zařízením za NATem. Přímé spojení vyžadují např. VoIP služby jako je Skype nebo služby vyžadující peer-to-peer spojení. Pro navázání přímého spojení dvou zařízení na NATem je potřeba NAT obejít. K tomu slouží metody NAT Traversalu, které si představíme v průběhu práce.

Metody NAT Traversalu v dnešní době využívá mnoho služeb na internetu. Problém je v tom, že technologie NAT nemá standard a kvůli tomu jsou metody NAT Traversal mnohdy špatně zdokumentované a closed source (software s uzavřeným kódem). Proto se v této práci pokusíme problematiku NAT Traversal přiblížit a demonstrovat v praktické části jednotlivé metody v reálném síťovém prostředí.



## 2 NAT

NAT je v dnešní době důležitou vlastností směrovačů a firewallů, jelikož pomáhá zachovat omezené množství veřejných IPv4 adres a zajišťuje bezpečnost komunikace mezi privátní a veřejnou sítí.

### 2.1 Využití

Když byla vytvořena IPv4 adresa a zavedena jako standard v roce 1982, nikdo si neuvědomoval, jak velký internet jednou bude. Přestože jsou k dispozici přes 4 miliardy ( $2^{32}$ ) IPv4 adres, tak v poslední době je velké téma jejich nedostatek a náhrada. Aby se nedostatku zabránilo, byly vytvořeny soukromé IP adresy a jejich překlad (NAT). Existují 2 druhy IPv4 adres:

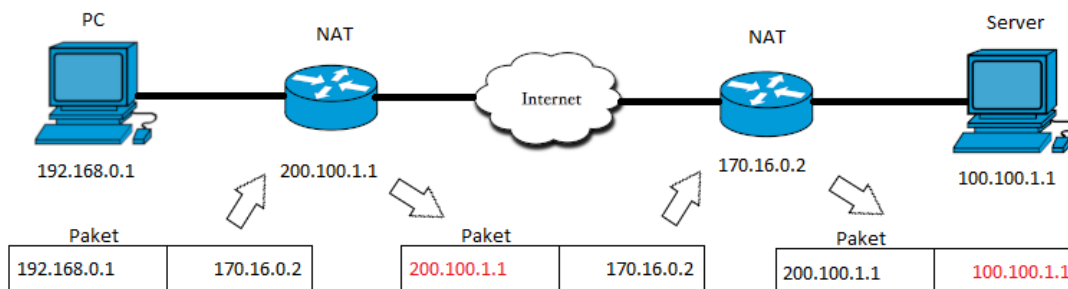
1. **Veřejné** - veřejně zaregistrované na internetu, jsou potřeba k připojení k internetu
2. **Privátní** - nejsou veřejně zaregistrované, jsou využívány pouze interně (např. v domácnosti nebo ve firmě), nelze se pomocí ní připojit k internetu

Privátní IP adresu přiřazuje jednotlivým zařízením směrovač. Většina domácností nemá pouze jedno zařízení, které potřebuje přístup k internetu, ale několik. Tato zařízení potřebují veřejnou IP adresu, aby se mohla připojit k internetu. Samozřejmě je možnost požádat poskytovatele internetového zařízení, aby přiřadil každému zařízení svou vlastní veřejnou IP adresu. To by bylo zbytečné, drahé a hned by se všechny IPv4 adresy vyčerpaly.

Místo toho necháme směrovač, aby přiřadil jednotlivému zařízení v domácnosti privátní IP adresu. Pokud budou zařízení potřebovat přístup k internetu, jejich privátní IP adresy budou pomocí NAT přeloženy na jednu veřejnou IP adresu. To zajistí, že se nevyčerpají všechny IPv4 adresy a zařízení z veřejné sítě nemůže zahájit spojení se zařízením v privátní síti. V budoucnu bychom NAT ani privátní IP adresy neměly potřebovat, protože IPv4 adresy by měly být nahrazeny novou generací IPv6 adres.

## 2.2 Překlad adres

Mechanismus NAT je popsán v dokumentu RFC 1631. Mechanismus překladu NAT převádí IP adresy v paketu během jeho průchodu směrovačem. Na obrázku 1 je znázorněn překlad adres u statického NATu.



Obrázek 1: Překlad adres ve statickém NATu

Host připojený do privátní sítě odesílá pakety na server. V tomto případě musí směrovač s NAT převést zdrojovou privátní IP adresu 192.168.0.1 na veřejnou IP adresu 200.100.1.1. Host sice odeslal paket se svou zdrojovou adresou, ta je ale privátní a je potřeba ji převést na registrovanou veřejnou IP adresu. Server přijme paket a myslí si, že komunikuje s hostem 200.100.1.1. Odešle zpět paket s cílovou adresou 200.100.1.1. Tu pak směrovač s NAT převede zpátky na vnitřní IP adresu 192.168.0.1 a odešle hostovi.

Na předchozím obrázku 1 lze popsat 4 typy adres v NAT [1]:

1. **Vnitřní lokální** - privátní IP adresa, která se nachází se uvnitř lokální sítě (192.168.0.1)
2. **Vnitřní globální** - adresa, pod kterou jsou zařízení v lokální síti vidět z vnější sítě (200.100.1.1)
3. **Vnější lokální** - privátní IP adresy zařízení ve vnější síti, jak jsou vidět ve vnitřní síti (170.16.0.2)
4. **Vnější globální** - veřejná IP adresa přiřazena koncovému zařízení ve vnější síti (100.100.1.1)

## 2.3 Druhy přiřazování adres

### 2.3.1 Statický NAT

Na směrovači, kde probíhá statický NAT se překládá jedna IP adresa vždy na stejnou IP adresu. Každá vnitřní lokální adresa je vždy pevně namapovaná na stejnou vnější globální IP adresu. Statický NAT mapuje adresy 1:1, a tedy nedochází k úsporám veřejných IPv4 adres. Statický NAT je výhodný, pokud chceme dát nějaký server veřejně k dispozici na Internetu, jelikož tento server bude mít

stále stejnou veřejnou IP adresu. Protože je směrovací tabulka statická, při každé změně je potřeba ji ručně upravit.

### 2.3.2 Dynamický NAT (tzv. Maškaráda)

Většina směrovačů s NAT mapuje více zařízení v lokální síti na jednu veřejnou IP adresu. K mapování využívá dynamický NAT množinu vnitřních globálních a vnitřních lokálních adres a mapování dvojí probíhá dynamicky podle potřeby. [1] Výhodou je zpřístupnění vnější sítě mnoha zařízením přes několik globálních adres. Princip dynamického NATu:

- Zařízení **A** ve vnitřní síti se chce spojit se serverem ve vnější síti
- Směrovač s NAT dostane paket.
- Směrovač pak nahradí IP adresu v paketu podle směrovací tabulky z vnitřní lokální adresy na vnitřní globální adresu. Pokud v tabulce neexistuje záznam, přidělí mu dostupnou vnitřní globální adresu. Paket pošle na cílovou adresu.
- Jakmile dostane směrovač odpověď, zkontroluje cílovou adresu. Podívá se do směrovací tabulky a zjistí, kterému zařízení cílová adresa patří.
- Tato vnitřní globální adresa v paketu se vymění za vnitřní lokální a paket se pošle do vnitřní sítě cílovému zařízení **A**.
- Tento proces se opakuje, dokud zařízení **A** komunikuje se serverem. Po určité době se záznam ve směrovací tabulce smaže.
- Pokud by se nyní chtělo připojit k serveru zařízení **B**, tak dostane stejnou vnitřní globální adresu, jako zařízení **A**.

Díky dynamickému NATu dosáhneme částečné úspory IPv4 adres.

### 2.3.3 Přetížený NAT (NAT overloading, Port Address Translation)

Přetížený NAT mapuje více vnitřních lokálních adres na jednu vnitřní globální adresu na různých portech. To umožňuje více zařízením využití jedné veřejné IP adresy. Přetížený NAT rozšiřuje směrovací tabulku o **vnitřní lokální port** a **vnitřní globální port**. Princip přetíženého NATu:

- Zařízení **A** ve vnitřní síti se chce spojit se serverem ve vnější síti
- Směrovač s NAT dostane paket.
- Směrovač s NAT nahradí IP adresu a port v paketu podle směrovací tabulky z vnitřní lokální na vnitřní globální. Paket potom odešle na cílovou adresu.

- Jakmile dostane směrovač odpověď, zkontroluje cílovou adresu a port. Podívá se do směrovací tabulky a zjistí, kterému zařízení cílová adresa s portem patří.
- Vnitřní globální adresa a port v paketu se vymění za vnitřní lokální a paket se pošle do vnitřní sítě cílovému zařízení **A**.
- Tento proces se opakuje, dokud zařízení **A** komunikuje se serverem. Po určité době se záznam v směrovací tabulce smaže.
- Pokud by se kdykoliv během předchozího procesu chtělo připojit k serveru zařízení **B**, tak dostane stejnou vnitřní globální adresu, jako zařízení **A**, ale jiný vnitřní globální port.

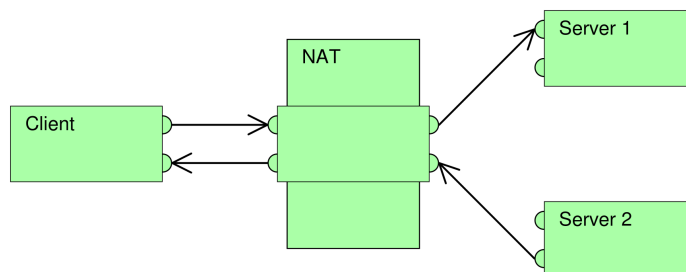
Přetížený NAT přináší výše zmiňované velké úspory IPv4 adres.

## 2.4 Druhy implementací NATu

Překlad síťových adres a portů lze implementovat několika způsoby. Některé aplikace, které používají informace o IP adrese, mohou potřebovat určit vnitřní globální IP adresu. Někdy může být potřeba určit typ mapování, pro navázání přímého spojení mezi dvěma klienty za NATem. Pro tento účel vznikl protokol Simple Traversal of UDP over NATs (STUN), který rozděluje jednotlivé NAT implementace. [2]

### 2.4.1 Full cone NAT (1:1 NAT)

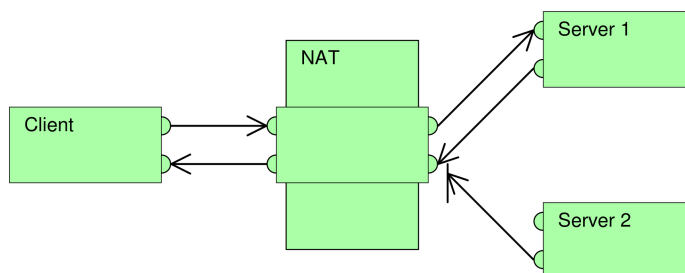
Full cone NAT je implementace, kde veškerá komunikace ze stejné vnitřní lokální IP adresy a portu jsou namapovány na právě jednu stejnou vnitřní globální IP adresu a port. Jakékoliv zařízení ve vnější síti může kontaktovat zařízení ve vnitřní síti odesláním paketu na mapovanou vnitřní globální IP adresu a port, jak je znázorněno na obrázku 2. **Server 2** zná vnitřní globální IP adresu a port, tudíž mu nic nebrání v navázání spojení s **klientem**.



Obrázek 2: Full cone NAT[3]

### 2.4.2 Restricted cone NAT

Restricted cone NAT je implementace, kde veškerá komunikace ze stejné vnitřní lokální IP adresy a portu jsou namapovány na právě jednu stejnou vnitřní globální IP adresu a port. Na obrázku 3 je rozdíl od Full-cone NATu na první pohled jasný. **Klienta** ve vnitřní síti mohou kontaktovat pouze taková vnější zařízení, která někdy v minulosti byla kontaktována vnitřní globální IP adresou zařízení. Tedy v případě obrázku 3 může **klienta** kontaktovat pouze **server 1** a příchozí pakety ze **serveru 2** jsou blokovány směrovačem s NAT.



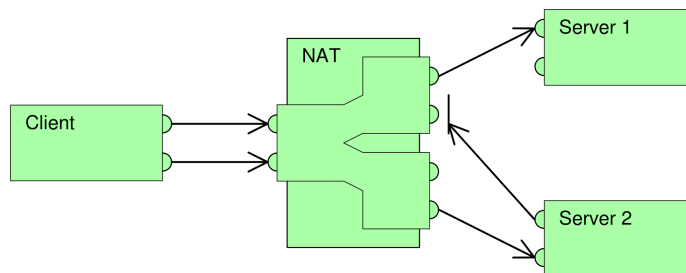
Obrázek 3: Restricted cone NAT[3]

### 2.4.3 Port restricted cone NAT

Port restricted cone NAT je podobný jako Restricted cone NAT, ale do restrikce se nyní přidá a vnitřní globální port. Vnější zařízení může odeslat paket na vnitřní globální IP adresu a port pouze, pokud by tou samou adresou a portem kontaktován v minulosti.

### 2.4.4 Symetrický NAT

Symetrický NAT je implementace, kde jsou všechny pakety ze stejné vnitřní lokální IP adresy a portu odeslané na konkrétní cílovou IP adresu a port. Každá z těchto kombinací je mapována na vnitřní globální IP adresu a port. Paket může být poslán ze stejné vnitřní lokální IP adresy a portu, ale pokud se změní cílová IP adresa, tak je vytvořeno nové mapování a je mu přidělena nová vnitřní globální IP adresa a port. Tento princip je znázorněn na obrázku 4.



Obrázek 4: Symetrický NAT[3]

Mnoho implementací NATu používá kombinace těchto druhů, proto je lepší se v praxi řídit chováním jednotlivých NATů. Tato chování se snaží definovat pomocí základní terminologie dokument RFC 4787. [4]

### 3 Metody NAT Traversal

NAT sice přináší spoustu výhod, ale má i svá negativa. Problém nastává, když se dva klienti za rozdílnými NATy snaží komunikovat. Řešením je použití některé z metod NAT Traversal (metody obcházení NATu), které jsou popsány v této sekci.

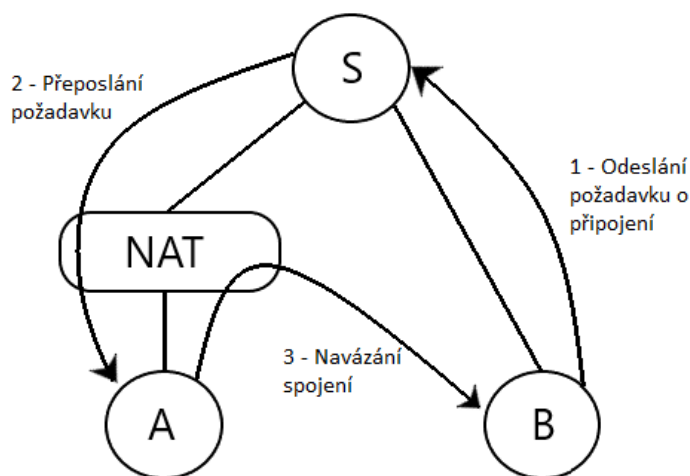
#### 3.1 Traversal Using Relays around NAT (TURN)

TURN poskytuje řešení pro dva klienty za NATem, kteří chtějí navzájem komunikovat a posílat média. Klíčem řešení je server, který slouží jako most mezi dvěma klienty. Oba klienti posílají pakety na server a ten je navzájem klientům přeposílá. Protokol TURN se nejvíce využívá k navázání peer-to-peer připojení mezi dvěma klienty, kteří jsou za symetrickým NATem. Přestože TURN poskytuje spolehlivé připojení, není výhodné ho používat jako výchozí metodu NAT Traversal kvůli náročnému udržování TURN serveru, přes který běží veškerá komunikace klientů. Neexistuje efektivnější řešení se stejnou spolehlivostí jakou poskytuje TURN, proto se využívá v moment, kdy klienti vyžadují maximální spolehlivost připojení. Protokoly využívané protokolem TURN pro přenos dat:

- UDP - preferovaný protokol pro přenos dat v reálném čase
- TCP - v základu by měl být vypnutý. Využívá se pouze, pokud je UDP zakázáno kvůli firemní politice nebo pokud není UDP dosažitelné z klienta na server.

## 3.2 Reverse connection

Reverse connection je metoda, která se využívá, pokud jsou oba klienti připojeni k serveru **S** a jeden klient za NATem, jak je ukázáno na obrázku 5. Klient **A** při pokusu o přímé spojení s klientem **B** nemá problém, protože **B** není za NATem. Pokud by se chtěl klient **B** připojit ke klientovi **A**, tak odešle přes server **S** požadavek na klienta **A**, aby zahájil přímé spojení s klientem **B**. Tato metoda je velmi limitovaná, ale využívá myšlenku přeposlání požadavku přes server, kterou využívají hole punching metody.



Obrázek 5: Obcházení NATu pomocí metody reverse connection

## 3.3 NAT hole punching

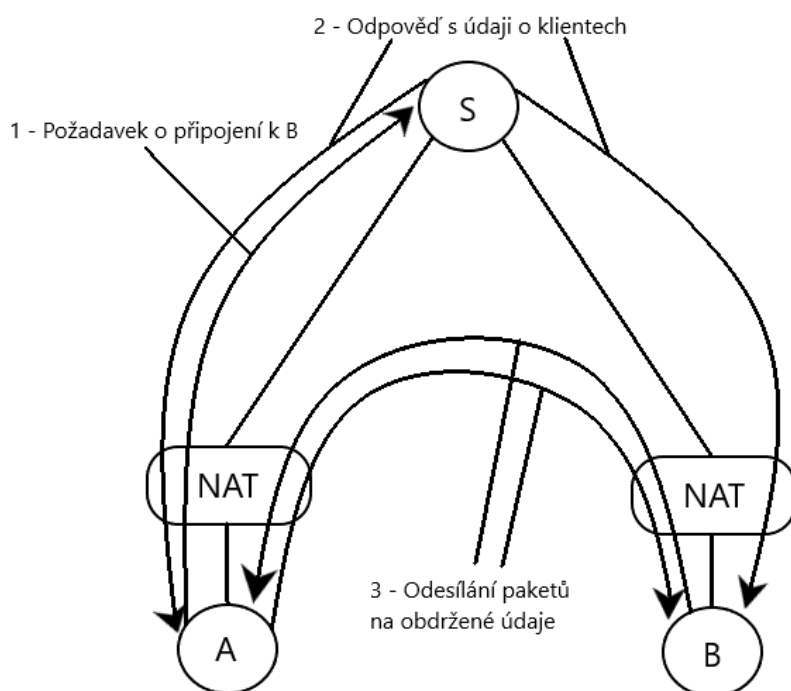
Hole punching je obecně často používaná metoda. Využívá toho, jak NAT zpracovává protokoly UDP a TCP pro navázání přímého spojení mezi dvěma klienty za pomoci serveru.

### 3.3.1 UDP Hole Punching

Tato metoda umožňuje klientům vytvořit přímé UDP spojení. Využívá se nejčastěji u akčních online her a VoIP produktů (např. Skype) kvůli snížené latenci, kterou UDP Hole Punching nabízí. UDP hole punching se spoléhá na to, že oba klienti za NATem, kteří chtějí navázat spojení jsou připojeni ke společnému serveru. Jakmile se nový klient připojí k serveru, jsou jeho vnitřní globální IP adresa a port a vnitřní lokální IP adresa a port zaznamenány na serveru (pokud se klient

nenachází za NATem, tak jsou tyto údaje totožné). Princip metody, který je na obrázku 6, funguje následovně:

1. Klient A chce navázat UDP spojení s klientem B, ale ten je za NATem. Proto pošle žádost o připojení na server S.
2. Server odpoví zprávou, která obsahuje vnitřní globální a vnitřní lokální údaje o klientovi B. Zároveň server pošle klientovi B údaje o klientovi A, aby mohli oba navázat UDP spojení.
3. Po obdržení zprávy ze serveru, začne klient A odesílat UDP pakety na vnitřní globální a vnitřní lokální údaje zároveň a čeká, ze kterých mu přijde odpověď od klienta B. Jakmile obdrží klient A odpověď, naváže pomocí těchto údajů spojení s klientem B.
4. Klient B také začne odesílat UDP pakety na obdržené údaje a navazuje spojení po obdržení odpovědi od klienta A.



Obrázek 6: Obcházení NATu pomocí metody UDP Hole Punching



### 3.3.2 TCP Hole Punching

TCP Hole Punching je metoda podobná UDP Hole Punchingu. Jelikož je protokol složitější a těžší na pochopení než UDP, tak není moc používaný směrovači s NATem. Pokud ho ovšem směrovač podporuje, je TCP Hole Punching stejně rychlý jako UDP Hole Punching a spolehlivější, protože u protokol TCP umožňuje určit dobu trvání jednotlivých TCP spojení. Hlavním problémem pro aplikace využívající TCP Hole Punching není složitost protokolu, ale to, že standardní socket API neumožňuje současně TCP socketu odesílat a přijímat data. Takže po připojení socketu na konkrétní port, pokusy o připojení druhého socketu na ten samý port selžou. Aby TCP Hole Punching fungoval, musí klient použít jeden místní port TCP, jak na poslouchání příchozích TCP spojení, tak na navazování TCP spojení. Většina operačních systémů v dnešní době podporují speciální nastavení TCP socketu, pojmenované `SO_REUSEADDR`, které umožňuje aplikacím navázat několik socketů na jeden port. [5]

Předpokládejme stejnou situaci jako u UDP Hole Punchingu. Tedy 2 klienti za NATem, kteří mají navázané TCP spojení se serverem. Potom by TCP Hole Punching vypadal následovně:

1. Klient A chce navázat TCP spojení s klientem B, ale ten je za NATem. Proto pošle žádost o připojení na server.
2. Server odpoví zprávou, která obsahuje vnitřní globální a vnitřní lokální údaje o klientovi B. Zároveň server pošle klientovi B údaje o klientovi A, aby mohli oba navázat TCP spojení.
3. Oba klienti využijí ten samý vnitřní port, který odeslali na server a začnou na něm poslouchat pro příchozí pakety. Zároveň se s tímto portem budou pokoušet navázat spojení s druhým klientem pomocí obdržených údajů ze serveru.
4. Klienti čekají, jestli se jim podaří navázat TCP spojení. Pokud se nějaký z odeslaných pokusů o TCP spojení selže, klient ho znovu odešle. Tento proces se opakuje, dokud se nepodaří navázat TCP spojení.
5. Po úspěšném navázání TCP spojení, klienti ověří, zda se připojili ke správnému klientovi. Pokud se jim správnost ověřit nepodaří, klient navázané spojení uzavře a čeká na další. Po spojení a ověření se všechny ostatní pokusy o spojení zruší.

Na rozdíl od UDP Hole Punchingu, kterému k navázání spojení stačí pouze jeden socket, TCP Hole Punching ke spojení potřebuje jeden socket na každou činnost (poslouchání, připojení k serveru, připojení ke klientovi).

### 3.4 Port Forwarding

Port Forwarding umožňuje přesměrovat komunikaci z jedné IP adresy a portu na druhou IP adresu a port, což způsobí obejetí směrovače s NATem. Vnější zařízení, která se budou chtít připojit ke klientovi ve vnitřní síti musí pro navázání komunikace znát přesměrovaný port. Port Forwarding může sloužit i jako příchozích paketů díky vytvoření pravidel přesměrování. Na rozdíl od Hole Punchingu, Port Forwarding redukuje provoz v síti vytvořený odeslanými zprávami na server.

Protokol UPnP (Universal Plug and Play) poskytuje funkci pro automatické přesměrování portů do vnitřních lokálních portů. Aplikace mohou využít protokol UPnP k rezervaci portu a přesměrovat příchozí pakety na socket, na kterém poslouchá. Využití protokolu UPnP může být riziko zabezpečení, protože UPnP nepodporuje žádné ověření příchozích paketů.

### 3.5 Přepínače

Styl kidiplom je z hlediska uživatele zastoupen ekvivalentně nazvanou třídou, kterou je třeba volat na začátku dokumentu:

```
1 \documentclass[
2   master=true,
3   font=sans,
4   printversion=false,
5   joinlists=true,
6   glossaries=true,
7   figures=true,
8   tables=true,
9   sourcecodes=true,
10  theorems=true,
11  bibencoding=utf8,
12  language=czech,
13  encoding=utf8,
14  program=infoi,
15  index=true,
16  biblatex=true
17 ]{kidiplom}
```

Zdrojový kód 1: Volání třídy **kidiplom**

Následuje přehled přepínačů, je vždy uvedeno jméno přepínač, včetně výchozí hodnoty. Přepínače uvádí tabulka [1](#).

Tabulka 1: Seznam přepínačů

Přepínač	Výchozí hodnota	Popis
<b>master</b>	false	Povolí nebo zakáže režim diplomové práce. Výchozí režim je tedy bakalářská práce.

<b>program</b>	<b>infpvs</b>	Specifikuje studijní program/obor
	<b>ainfvs</b>	(specializaci):
	<b>infoi</b>	Informatika (Obecná informatika) – bakalářský i navazující magisterský,
	<b>infpvs</b>	Informatika (Programování a vývoj software) – bakalářský,
	<b>itp</b>	Informační technologie – bakalářský, prezenční forma,
	<b>itk</b>	Informační technologie – bakalářský, kombinovaná forma,
	<b>infui</b>	Informatika (Umělá inteligence) – navazující magisterský,
	<b>ainfvs</b>	Aplikovaná informatika (Vývoj software) – navazující magisterský,
	<b>ainfpst</b>	Aplikovaná informatika (Počítačové systémy a technologie) – navazující magisterský,
	<b>infv</b>	Informatika pro vzdělávání – bakalářský,
	<b>uinf</b>	Učitelství informatiky pro střední školy – navazující magisterský,
	<b>binf</b>	Bioinformatika – bakalářský i navazující magisterský,
	<b>inf</b>	Informatika (bez specializací) – bakalářský i navazující magisterský,
	<b>ainfp</b>	Aplikovaná informatika (bez specializací) – bakalářský, prezenční forma,
	<b>ainfk</b>	Aplikovaná informatika (bez specializací) – bakalářský, kombinovaná forma,
	<b>ainf</b>	Aplikovaná informatika (bez specializací) – navazující magisterský.

<b>font</b>	<code>serif</code>	Zapne či vypne podporu pěkného bezpatkového fontu. Možné hodnoty jsou:  <b>sans</b> Bezpatkové písmo (písmo Iwona).  <b>serif</b> Patkové písmo (písmo Computer Modern).
<b>encoding</b>	<code>utf8</code>	Kódování souboru dokumentu, doporučuje se ponechat výchozí hodnotu.
<b>bibencoding</b>	<code>utf8</code>	Kódování souboru bibliografie. Tato volba má smysl pouze, pokud je použita bibliografie skrze balíček <code>BIBLATEX</code> .
<b>language</b>	<code>czech</code>	Jazyk práce.
<b>printversion</b>	<code>false</code>	Je-li zapnuto, pak budou odkazy vysázeny optimalizovaně pro knižní sazbu. Tuto volbu je nutno použít pro tisk práce.
<b>joinlists</b>	<code>true</code>	Je-li zapnuto, pak seznamy obrázků, tabulek, vět a zdrojových kódů sázené za obsahem nebudou rozděleny na samostatné stránky.
<b>figures</b>	<code>true</code>	Je-li zapnuto, pak v seznamech položek bude zahrnut seznam obrázků.
<b>tables</b>	<code>true</code>	Je-li zapnuto, pak v seznamech položek bude zahrnut seznam tabulek.
<b>theorems</b>	<code>false</code>	Je-li zapnuto, pak v seznamech bude zahrnut seznam teorémů.
<b>sourcecodes</b>	<code>false</code>	Je-li zapnuto, pak v seznamech bude zahrnut seznam zdrojových kódů.
<b>glossaries</b>	<code>false</code>	Je-li zapnuto, pak na konci dokumentu bude vysázen seznam zkratk.
<b>index</b>	<code>false</code>	Zapíná podporu sazby rejstříku.
<b>biblatex</b>	<code>true</code>	Zapne sazbu bibliografie přes balík <code>BIBLATEX</code> .

### 3.6 Geometrie stránky

Tento styl používá list velikosti A4. Pro sazbu prací je třeba použít jednostrannou sazbu. Levý okraj je rozšířen s ohledem na vazbu výsledné knižní podoby práce.

## 4 Sazba částí dokumentu

### 4.1 Sazba úvodní strany či obsahu

Vysázení všech podstatných částí úvodu práce obstará makro `\maketitle`. Pro správné vysázení všech částí a meta-informací je potřeba použít makra `\title`, `\author` a další. Jejich přehled lze najít ve zdrojovém souboru tohoto dokumentu. V případě použití **pdf** výstupu se generuje i dodatečná hlavička souboru s meta-informacemi jako je autor dokumentu, název práce či dalšími.

### 4.2 Závěry

Závěr práce by se měl poskytnout jak v původním jazyce práce, tak v jazyce anglickém. Pro sazbu závěru jsou k dispozici příslušná makra. Berte na vědomí, že v anglickém závěru se aktivuje plně anglická sazba se všemi konvencemi. Tedy je třeba používat anglické uvozovky a další správné typografické prvky.

```
1 % Tiskne český závěr práce.
2 \begin{kiconclusions}
3 Závěr práce v \uv{českém} jazyce.
4 \end{kiconclusions}
5
6 % Tiskne anglický závěr práce.
7 \begin{kiconclusions}[english]
8 Thesis conclusions written in \uv{English}.
9 \end{kiconclusions}
```

Zdrojový kód 2: Sazba závěrů

### 4.3 Matematika

Pro sazbu matematiky je k dispozici sada standardních maker.

$$\langle f \rangle, [g], [h], \lceil i \rceil$$

$$\left\{ \frac{x^2}{y^3} \right\}$$
$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$
$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

## 4.4 Sazba literatury

Pro sazbu literatury má uživatel dvě možnosti. Může použít služeb balíků `BIBLATEX`, který je pro `kistyles` zapnutý, či lze použít manuální sazbu bibliografie.

### 4.4.1 Sazba bibliografie přes `BibLATEX`

Při použití tohoto balíku se data o použité literatuře ukládají do dedikovaného textového souboru, ukázku najdete i v tomto stylu pod jménem `bibliografie.bib`.

Formát daného souboru je nad rámec této dokumentace a je na každém uživateli, aby si jej nastudoval. Bibliografie se tiskne makrem `\printbibliography`. Taktéž v preambuli dokumentu je třeba definovat, který soubor data bibliografie obsahuje, tedy například `\bibliography{bibliografie.bib}`.

Dokument, který využívá `BIBLATEX` je následně nutné přeložit jak pomocí překladače zvoleného ovladače, tak pomocí aplikace `biber`. Více informací poskytnou soubor `Makefile` z distribuce tohoto stylu.

Výhodou tohoto přístupu je, že bibliografie se vysází automaticky a (obvykle) není třeba manuální úprava formátování.

### 4.4.2 Manuální sazba bibliografie

Manuální sazba obnáší vysázení prostředí `thebibliography` ručně. To je nad rámec tohoto dokumentu. Ukázku tohoto přístupu lze samozřejmě nalézt ve zdrojovém souboru tohoto dokumentu nebo také [zde](#).

Pro aktivaci manuální sazby bibliografie je třeba volat třídu `kidiplom` s parametrem `biblatex=false`. Mějte, prosím, na paměti, že v tomto módu jsou makra `\bibliography` a `\printbibliography` nedostupná.

## 4.5 Drobná makra

Základní styl definuje hned několik maker pro usnadnění práce. Například makro `\buno` vysází řetězec „bez újmy na obecnosti“. Je k dispozici i verze s prvním velkým písmenem, `\Buno`.

Je rovněž možno přidávat položky do seznamu zkratk. K tomu slouží makro `\newacronym`, které lze použít například jednoduše jako `\newacronym{UPOL}{UPOL}{\kitextunivcz}`. Na danou zkratku se pak lze odkazovat jednoduše, `\gls{UPOL}`.

Sazba uvozovek respektuje nastavení částí dokumentu, a proto se doporučuje používat makro `\uv`. V anglické závěru práce toto platí taky, viz tato PDF ukázka.

Styl podporuje sazbu odstavců v tabulkách, více obsahuje tabulka [2](#).

K dispozici jsou také makra pro sazbu `C#` (`\csharp`) či `C++` (`\cpp`).

## 4.6 Sazba rejstříku

Sazba rejstříku sestává z několika kroků:

Tabulka 2: Odstavce v tabulkách

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.	Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.	Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.
---	--	--



1. Je třeba přes volbu `index=true` rejstříkování povolit.
2. Použitím makra `\index` rejstříkovat vybrané pojmy.
3. Kompilovat s použitím utility `makeindex`. Pro specifika tohoto kroku si stačí prohlédnout soubor `Makefile`.

Makro `\index` je redefinováno tak, že sází klikací odkaz na výraz v rejstříku. Je doporučeno jej použít ihned za výrazem<sup>{1}</sup>.

**Omezení redefinovaného makra `\index`:** klikací odkaz nefunguje, pokud použijete konstrukci `\index{výraz|makro}` (resp. `\index{výraz|(makro)}`), např. `\index{výraz|textit}`.

Rejstřík lze vysázet pomocí makra `\printindex`.

## 4.7 Sazba zdrojových kódů

Styl nabízí dva způsoby sazby zdrojových kódů:

1. Sazbu řádkových kódů, například **`background-color: white;`**. K tomu slouží makro formátu `\kiinlinecode{jazyk}{separátor}{kód}`. Za separátor je vhodné volit jakýkoliv znak, který se nevyskytuje v samotném sázeném zdrojovém kódu. Za jazyk je nutno dosadit jeden z těchto: C, TeX, PHP, HTML, Lisp, SQL, TeX, Python, Java, TutorialD, text, csharp, cpp, JavaScript, CSS.
2. Sazbu zdrojových kódů do separátních prostředí. Takto vytištěný kód se objeví v seznamu zdrojových kódů. Ukázka například zdrojový kód 3. Ukázku sazby naleznete ve zdrojovém kódu tohoto dokumentu.

### Definice 1 (Název definice)

Abcd. [Univerzita Palackého v Olomouci \(UPOL\)](#)

### Důkaz (Název důkazu)

Abcd. □

### POZNÁMKA 2 (PUMPOVACÍ VĚTA)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.

### PŘÍKLAD 3 (PUMPOVACÍ VĚTA)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.

### Lemma 4 (Název definice)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd. Abcd.

### Důsledek 5 (Název důkazu)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd.

### Věta 6 (Pumpovací věta)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.  
Abcd. Abcd. Abcd. Abcd. Abcd.

```
1 int main("cs acsa") // komentar
2 int main("cs acsa") // komentar
3 int main("cs acsa") // komentar
4 int main("cs acsa") // komentar
5 int main("cs acsa") // komentar
```

Zdrojový kód 3: C++

```
1 new object() // komentar
```

Zdrojový kód 4: JS

```
1 public static int main("cs acsa") // komentar
```

Zdrojový kód 5: C#

```
1 SELECT * FROM table_1; /* komentar */
```

Zdrojový kód 6: SQL

```
1 table_1 AND table_2;
```

Zdrojový kód 7: TutorialD

## **Závěr**

Závěr práce v „českém“ jazyce.

## Conclusions

Thesis conclusions in “English”.

## A První příloha

Text první přílohy

## B Druhá příloha

Text druhé přílohy

## C Obsah přiloženého datového média

Na samotném konci textu práce je uveden stručný popis obsahu přiloženého datového média (CD/DVD, flash disk apod.), tj. jeho závazné adresářové struktury, důležitých souborů apod.

### **bin/**

Instalátor `INSTALATOR` programu, popř. program `PROGRAM`, spustitelné přímo z média. / Kompletní adresářová struktura webové aplikace `WEBOVKA` (v ZIP archivu) pro zkopírování na webový server. Adresář obsahuje i všechny runtime knihovny a další soubory potřebné pro bezproblémový běh instalátoru a programu z média / pro bezproblémový provoz webové aplikace na webovém serveru.

### **doc/**

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

### **src/**

Kompletní zdrojové texty programu `PROGRAM` / webové aplikace `WEBOVKA` se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu / adresářové struktury pro zkopírování na webový server.

### **readme.txt**

Instrukce pro instalaci a spuštění programu `PROGRAM`, včetně všech požadavků pro jeho bezproblémový provoz. / Instrukce pro nasazení webové aplikace `WEBOVKA` na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Navíc médium obsahuje:

### **data/**

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

**install/**

Instalátory aplikací, runtime knihoven a jiných souborů potřebných pro provoz programu PROGRAM / webové aplikace WEBOVKA, které nejsou standardní součástí operačního systému určeného pro běh programu / provoz webové aplikace.

**literature/**

Vybrané položky bibliografie, příp. jiná užitečná literatura vztahující se k práci.

U veškerých cizích převzatých materiálů obsažených na médiu jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na médiu, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.





## Literatura

- [1] Odom, Wendell; Healy, Rus; Mehta, Naren. *Směrování a přepínání sítí: autorizovaný výukový průvodce: Samostudium* [online]. First. Brno): Computer Press, 2009 [cit. 2022-7-20]. ISBN 978-802-5125-205.
- [2] Rosenberg, J.; Weinberger, J.; Huitema, C.; Mahy, R. STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs): RFC 3489. *Network Working Group* [online]. 2003, [cit. 2022-7-20]. Dostupný z: <https://datatracker.ietf.org/doc/html/rfc3489>.
- [3] Wikipedia. *Network address translation* [online]. 2022 [cit. 2022-7-21]. Dostupný z: [https://en.wikipedia.org/wiki/Network\\_address\\_translation](https://en.wikipedia.org/wiki/Network_address_translation).
- [4] Audet, F.; Jennings, C. Network Address Translation (NAT) Behavioral Requirements for Unicast UDP: RFC 4787. *Network Working Group* [online]. 2007, [cit. 2022-7-22]. Dostupný z: <https://datatracker.ietf.org/doc/html/rfc4787>.
- [5] Ford, Bryan; Srisuresh, Pyda; Kegel, Dan. *Peer-to-Peer Communication Across Network Address Translators* [online]. [cit. 2022-7-23]. Dostupný z: <https://bford.info/pub/net/p2pnat/>.
- [6] Greenberg, David. Camel drivers and gatecrashers: quality control in the digital research library. In Hawkins, B.L; Battin, P (ed.). *The mirage of continuity: reconfiguring academic information resources for the 21st century*. Washington (D.C.): Council on Library and Information Resources; Association of American Universities, 1998, s. 105–116.
- [7] LYNCH, C. Where do we go from here?: the next decade for digital libraries. *DLib Magazine* [online]. 2005, vol. 11, no. 7/8 [cit. 2005-8-15]. Dostupný z: <http://www.dlib.org/dlib/july05/lynch/07lynch.html>. ISSN 1082-9873.
- [8] Národní knihovna. A big paper. *The journal of big papers*. 1991, vol. 12, no. 3. ISSN 2232-332X.
- [9] Děta, Hugh; Rychlík, Tomáš. *A big paper: Podtitul* [online]. Druhé vyd. Praha: Academia, 1991 [cit. 2011-1-12]. 550 s. Pokusná edice. Dostupný z: <http://pokus.cz>. ISBN 978-44-55-X.
- [10] Freely, I.P. A small paper: Podtitulek. *The journal of small papers*. 1997, roč. 1, č. 3, s. 2–5. to appear.
- [11] Jass, Hugh. A big paper. *The journal of big papers*. 1991, roč. 23.
- [12] Černý, Hugh. Titulek. *The journal of big papers*. 1991, roč. 12, č. 2, s. 22–44. Dostupný také z: <http://dx.doi.org/10.112.22/jkn>.
- [13] Kollmannová, Ludmila; Bubeniková, Libuše; Kopecká, Alena. *Angličtina pro samouky*. 5. vyd. Praha: Státní pedagogické nakladatelství, 1977. 525 s. Učebnice pro samouky. ISBN 80-04-25663-5.

- [14] Kollmannová, Ludmila; Bubeniková, Libuše; Kopecká, Alena. *Angličtina pro samouky*. 5. vyd. Novotná, Pepina. Praha: Státní pedagogické nakladatelství, 1977. 2. Podkapitola, s. 22–29. ISBN 80-04-25663-5.
- [15] *TUGBoat*. 1980-. Dostupný také z: <http://tugboat.tug.org/>. ISSN 1222-3333.
- [16] Knuth, Donald. Journeys of TeX. *TUGBoat*. 2003, vol. 17, no. 3, s. 12–22. Dostupný také z: <http://tugboat.tug.org/kkk.pdf>. ISSN 1222-3333.
- [17] Geniální, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007. ISBN 978-222-626-222-2.
- [18] Vlaštovka, Josef. Velmi zajímavý článek. In Geniální, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007, s. 22–45. ISBN 978-222-626-222-2.

# Rejstřík

výraz, [24](#)