



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

MĚŘENÍ ZTRÁTOVOSTI A RTT

PROJEKTOVÁ DOKUMENTACE

PROJECT DOCUMENTATION

AUTOR PRÁCE

AUTHOR

ANETA HELEŠICOVÁ

BRNO 2017

Obsah

1	Úvod	2
1.1	Zadání	2
1.2	Cíl práce	3
2	Teorie	4
2.1	O co se jedná?	4
2.2	Teorie implementace	4
2.3	Výpisy	4
3	Implementace	6
3.1	Jádro projektu	6
4	Typografické a jazykové zásady	8
4.1	Co to je normovaná stránka?	9
5	Závěr	11
	Literatura	12
A	Jak pracovat s touto šablonou	13
B	Ukazka rozsireni sablony	17
B.1	Definice v projekt.tex	17
B.2	Ukazky kodu	18
B.3	TODO	20

Kapitola 1

Úvod

Tato dokumentace byla vytvořena k projektu Programování síťové služby, varianty Měření ztrátovosti a RTT do předmětu Síťové aplikace a správa sítí školního roku 2017/2018. Cílem dokumentace je pečlivě popsat způsob řešení problému.

1.1 Zadání

Pro úplnost dokumentace je níže vloženo zadání z informačního systému.

Popis varianty: Vytvořte program, který monitoruje zadané síťové uzly. V případě ztráty paketů nebo překročení Round trip time (RTT) nad zvolenou hodnotu, vypíše informace na standardní výstup.

Příklad spuštění:

```
./testovac [-h] [-u] [-t <interval>] [-i <interval>] [-p <port>] [-l <port>] [-s <size>] [-r <value>] <uzel1> <uzel2> <uzel3> ...
```

-h – zobrazí nápovědu

-u – pro testování se použije UDP protokol

-s – velikost dat pro odeslání, výchozí hodnota 56B

-t <interval> – interval v sekundách za který je ztrátovost vyhodnocována, ve výchozím nastavení 300s

-i <interval> – interval v ms jak často zasílat testovací zprávy, ve výchozím nastavení 100 ms

-w <timeout> – doba jak dlouho se čeká na odpověď, pouze při neobdržení odpovědi, výchozí hodnota 2s jinak 2xhodnota vypočteného RTT

-p <port> – specifikace UDP portu

-l <port> – specifikace naslouchaného UDP portu

-r <value> – specifikace RTT hodnoty, pokud RTT překračuje danou hodnotu, reportuje se

-v – verbose mód, program vypisuje na stdout přijaté pakety, tj. chová se jako příkaz ping

<uzel> – IPv4/IPv6/hostname adresa uzlu

Upřesnění zadání: Pokud není specifikováno přepínačem -u a -p, použije se pro testování protokol ICMP echo request/reply. Při specifikaci protokolu UDP bude program zasílat UDP paket na daný port s 64B náhodných dat a bude očekávat odpověď se stejným obsahem. Pokud je specifikován port přepínačem -l, program naslouchá na UDP portu a na příchozí pakety odpovídá UDP paketem se stejným obsahem. Program musí zpracová-

vat jednotlivé uzly paralelně. Pokud není zadán přepínač -r, testuje se pouze ztrátovost paketů, RTT se v tomto případě počítá pouze pro souhrnné statistiky. Lze doporučit inspiraci nástrojem ping, který např. vkládá timestamp do obsahu ICMP paketu pro snadnější implementaci a výpočet RTT. Nicméně není to povinnost takto řešit.

Příklad výpisu v případě detekovaných chyb:

```
2017-02-01 10:35:12.23 uzel3: 0.167% packet loss, 5 packet lost
2017-02-01 10:35:12.23 uzel1: 0.067% packet loss, 2 packet lost
2017-02-01 10:35:12.23 uzel2: 0.067% (2) packets exceeded RTT threshold 5ms
```

Každou hodinu vypíše program statistiku o stavu jednotlivých uzlů: 2017-02-01

```
12:00:00.00 uzel1: 0% packet loss, rtt min/avg/max/mdev 4.845/4.882/4.912/0.063 ms
2017-02-01 12:00:00.00 uzel2: 0% packet loss, rtt min/avg/max/mdev 18.283/18.311/18.343/0.024 ms
2017-02-01 12:00:00.00 uzel3: status down
```

Pro výpis RTT, průměrných a max. hodnot zaokrouhľujte na 3 desetinná čísla. Za status down je označen prvek, který neodpovídá na testovací data (paket loss = 100%). Pro testování můžete interval pro výpočet statistiky zkrátit.

Příklad výpisu při přepínači -v:

```
2017-02-01 10:35:12.23 64 bytes from uzel3 (172.16.23.238) time=4.76 ms
2017-02-01 10:35:12.33 64 bytes from uzel3 (172.16.23.238) time=4.73 ms
2017-02-01 10:35:12.43 64 bytes from uzel3 (172.16.23.238) time=4.73 ms
```

Rozšíření: Za rozšíření nelze získat body nad rámec projektu.

Lze implementovat rozšíření, kde RTT hodnota je specifikovaná pro každý uzel zvlášť. Syntaxe je následující: <uzel;RTT> - např. 2001:db8::1;4.5 znamená monitorování prvku s IPv6 adresou 2001:db8::1. Pokud dojde při měření ke zjištění RTT > 4.5 ms, bude hodnota zaznamenána a po uplynutí interval -t vypsána na standardní výstup nebo na syslog.

Vytvoření souboru testovac.spec umožňující vytvořit balíček .rpm

1.2 Cíl práce

Cílem projektu je tedy vytvořit program implementovaný v jazyce C nebo C++, který bude měřit ztrátovost a obousměrné zpoždění libovolného počtu na vstup zadaných uzlů. Program je vytvořený pro distribuci CentOS 7 a pro používání v příkazové řádce. Program smí používat pouze uživatelé s administrátorskými právy. O funkcionalitě si může do velké míry rozhodnout sám uživatel používáním parametrů.

Kapitola 2

Teorie

2.1 O co se jedná?

V této sekci bych se ráda pokusila vysvětlit projekt tak, aby mu rozuměl i laik. Dobrou průpravou pro tento text mi byl rozhovor s maminkou, která za mnou přišla s dotazem "A co že to vlastně děláš?". Takže, mami...

Dnes již téměř nedílnou součástí počítačů je jejich připojení do sítě a vzájemná komunikace mezi sebou. Bez internetu si už pomalu nedokážeme počítač představit. Proto se dostáváme do úzkých, když nám nějaký vzdálený počítač přestane odpovídat na naše zprávy. Pro zjištění toho, zda počítač odpovídá, běžně užíváme nástroje **ping**. Ten je ovšem vhodný pouze pro krátkodobé testování. Pokud chceme zjistit, jak nám vzdálený počítač odpovídá v průběhu hodin či dní, je vhodnější využít právě programu takového, jaký zde prezentuji. Tento program totiž sám počítá statistiky odpovědí, dokáže nám říct na kolik procent zpráv nám druhý počítač odpověděl a kupříkladu taky jak dlouho mu to průměrně trvá.

Tyto programy jsou tedy, jak je asi zřejmé, důležitou součástí testování sítí.

2.2 Teorie implementace

Testování je implementováno pomocí zasílání ICMP packetů TCP protokolem.

Program zasílá jeden ICMP packet typu ECHO REQUEST každou časovou jednotku (zadáno uživatelem). Následně čeká na ECHO REPLY k tomuto packetu. Po obdržení je vyhodnoceno obousměrné zpoždění packetu - čas od odeslání ECHO REQUEST až po příjem ECHO REPLY. Aby si program byl jist, zda se jedná o správnou odpověď, musí také testovat jednotlivé součásti přijmutého packetu. Pokud k packetu žádná odpověď nepříjde, případně přijde až příliš pozdě, je packet vyodnocen jako ztracený. Pokud si to uživatel přeje, packety se mohou vyhodnotit i jako zpožděné, pokud překročí uživatelem stanovenou hodnotu obousměrného zpoždění.

2.3 Výpisy

Výpisy jsou zcela jistě nedílnou součástí tohoto programu, kdyby se uživatel nedozvěděl co se děje, byl by mu program úplně k ničemu.

Program má několik typů výpisů. Prvním typem výpisu je výpis ztrátovosti. Ten se provádí každých 300 sekund, pokud uživatel nestanovil jinak (pomocí parametru -t) a pokud

byl za stanovenou dobu na uzlu nějaký problém (nenulová ztrátovost nebo zpožděnost). Tento výpis je v jednoduchém formátu `TIMESTAMP UZEL: ZTRÁTOVOST`. Timestamp je klasický formát času `RRRR-MM-DD HH:MM:SS.MS`, udává se čas, kdy byl výpis spuštěn. Hodnota uzel je taktéž velmi prostá, jedná se o uzel, na kterém je ztrátovost vyhodnocena, ve formátu v jakém byl uživatelem zadán na vstup. Hodnota ztrátovost je základním kamenem tohoto výpisu. Rozlišujeme dvě varianty - statistika ztrátovosti a statistika zpožděných packetů. Jedná-li se o statistiku ztrátovosti, je klasicky vyhodnocena jako procentuální poměr ztracených packetů a přesným číselným vyjádřením počtu ztracených packetů. Pokud je ztrátovost 100%, uzel hlásí status down. Statistika zpožděnosti je ve formátu procentuální poměr zpožděných packetů, (počet zpožděných packetů) překročil uživatelem stanovenou maximální hodnotu obousměrného zpoždění *uživatelem stanovená hodnota ms*.

2017-11-20 12:52:35.20 seznam.cz: 0.165% packet loss, 2 packet lost

2017-11-20 12:52:35.20 seznam.cz: 0.330% (4) packets exceeded RTT threshold 35 ms

2017-11-20 12:52:35.20 seznam.cz: 0.330% (4) packets exceeded RTT threshold 35 ms
0.165% packet loss, 2 packet lost

Druhým typem výpisu jsou hodinové statistiky. Tento čas nemůže uživatel ovlivnit, každou hodinu se vypíše statistika ztrátovosti ve formě `TIMESTAMP UZEL: ZTRÁTOVOST RTT-STATISTIKA`. Hodnoty timestamp a uzel jsou stejně jako při předešlém výstupu. Hodnota ztrátovost je taktéž podobná, vyhodnocuje se jako procentuální ztrátovost packetů. Rtt statistika je novou hodnotou, která obsahuje několik typů údajů o obousměrném zpoždění, které by mohly uživatele zajímat. Přesněji se jedná o minimální hodnotu, průměrnou hodnotu, maximální hodnotu a mean odchylku obousměrného zpoždění všech packetů, které byly za poslední hodinu vyhodnoceny.

2017-11-20 12:52:35.20 seznam.cz: 0.165% packet loss, rtt min/avg/max/mdev

4.523/12.682/20.402/2.503 ms

Posledním typem výpisu je výpis, který si může uživatel zapnout argumentem `-v`. Jedná se o takzvaný verbose výstup, s jehož používáním je program velmi podobný nástroji `ping`. Výpis probíhá při každém přijatém packetu a je ve formátu `TIMESTAMP VELIKOST UZEL (IP) RTT`. Timestamp a uzel jsou opět známé hodnoty. Hodnota velikost udává velikost packetu, který přišel na vstup. (Pozor, nerovná se případnému zadanému `-s`, neboť `-s` udává jen náhodná data, která se za packet přidávají, zde se jedná o velikost celého packetu.) . Hodnota IP je důležitá zejména, je-li uzel zadán pomocí `hostname`. Nabývá totiž hodnoty IPv4/IPv6 adresy, na kterou packet zasíláme. Poslední hodnota Rtt nám ve formátu `time=obousměrné zpoždění konkrétního packetu ms`.

2017-11-20 12:52:35.20 75 bytes from seznam.cz (77.75.79.39) time=35.245 ms

Kapitola 3

Implementace

V této kapitole se budu zabývat řešením implementace.

3.1 Jádru projektu

Projekt jsem implementovala v jazyce C++, standart C++11. Vyvíjen a testován je pro operační systém CentOS 7, k testování jsem používala virtuální stroj, jehož předpis jsme dostali k dispozici.

Běh programu začíná zpracováním argumentů, které se uluží do struktury. Pokud byl zadán argument -h, je vypsán Help a program se ukončí. Jinak se pro každý zadaný uzel vytvoří dvě `std::thread` vlákna, jedno pro zasílání packetů, druhé pro jejich přijímání. Dvě vlákna jsou důležitá, aby program mohl dodržovat rozestupy pro zasílání zpráv bez toho, aby musel čekat na odpověď, což by v případě, kdy by musel čekat déle, než dobu, za kterou má poslat další packet, způsobovalo nevyžádané chování.

Ve vlákne k zasílání packetů pak proběhne získání dat o adrese pomocí `gethostbyname` v případě IPv4 a `getaddrinfo` v případě IPv6 adresy. Dále dojde k vytvoření RAW socketu pro ICMP pomocí funkce `socket`. Tento socket se pak nastaví funkcí `setsockopt`. Následuje vytvoření ICMP hlavičky a datové části packetu. Nastavení ICMP hlavičky probíhá standartním způsobem. Velikost datové části se určí pomocí argumentu -s, případně je defaultně nastaveno 56B. Na začátek datové části se uloží informace o aktuálním čase, za tato data se připojí náhodná data o správné velikosti. Datová část s hlavičkou se poté spojí, vypočítá se kontrolní součet, který se uloží do hlavičky a packet je zaslán funkcí `sendto`. Poté se thread na počet sekund stanovený parametrem -i, případně defaultně na 100 ms uspí funkcí `usleep`. Tento thread má ještě jednu funkci a to zajišťování výpisů ztrátovosti a hodinových výpisů. Zda je správný čas zjišťuje pomocí srovnávání dvou `timeval` struktur nastavovaných funkcí `gettiimeofday`. V případě, že je čas některý výpis vypsát, přejde do funkce buď `tOutput` nebo `hourOutput`. Výpisy jsou popsány v předešlé kapitole.

Druhé vlákno slouží k přijímání packetů. Začne vytvořením socketu a jeho nastavením stejně jako thread pro zasílání. Dále v nekonečné smyčce naslouchá pomocí funkce `recvfrom`. Přijde-li nějaký packet, zkontroluje především jeho typ (ICMP ECHOREPLY), id a neporušenost dat. Pokud packet vše správně splňuje, označí jej za přijatý. Pokud byl zadán parametr -r a packet svým rtt překročí stanovenou hodnotu, je označen za zpožděný. Pokud na packet vlákno čeká déle než hodnotu -w nebo dvojnásobek předešlého rtt, je označen za ztracený. Vlákno tedy ještě počítá rtt každého packetu, tuto hodnotu pak ukládá pro

pozdější zpracování. Dále má na starosti výpis verbose, pokud byl uživatelem vyžádaný pomocí parametru -v (popsán taktéž v předešlé kapitole).

Program aktuálně funguje pouze pro IPv4 adresy a hostname, která jsou do IPv4 přeložitelná. Při zadání parametru -u je vypsána chybová hláška a program dále pokračuje stejně, jako by parametr zadán nebyl - pomocí ICMP.

Kapitola 4

Typografické a jazykové zásady

Při tisku odborného textu typu *technická zpráva* (anglicky *technical report*), ke kterému patří například i text kvalifikačních prací, se často volí formát A4 a často se tiskne pouze po jedné straně papíru. V takovém případě volte levý okraj všech stránek o něco větší než pravý – v tomto místě budou papíry svázány a technologie vazby si tento požadavek vynucuje. Při vazbě s pevným hřbetem by se levý okraj měl dělat o něco širší pro tlusté svazky, protože se stránky budou hůře rozevírat a levý okraj se tak bude oku méně odhalovat.

Horní a spodní okraj volte stejně veliký, případně potištěnou část posuňte mírně nahoru (horní okraj menší než dolní). Počítejte s tím, že při vazbě budou okraje mírně oříznuty.

Pro sazbu na stránku formátu A4 je vhodné používat pro základní text písmo stupně (velikosti) 11 bodů. Volte šířku sazby 15 až 16 centimetrů a výšku 22 až 23 centimetrů (včetně případných hlaviček a patiček). Proklad mezi řádky se volí 120 procent stupně použitého základního písma, což je optimální hodnota pro rychlost čtení souvislého textu. V případě použití systému LaTeX ponecháme implicitní nastavení. Při psaní kvalifikační práce se řiďte příslušnými závaznými požadavky.

Stupeň písma u nadpisů různé úrovně volíme podle standardních typografických pravidel. Pro všechny uvedené druhy nadpisů se obvykle používá polotučné nebo tučné písmo (jednotně buď všude polotučné nebo všude tučné). Proklad se volí tak, aby se následující text běžných odstavců sázel pokud možno na *pevný rejstřík*, to znamená jakoby na linky s předem definovanou a pevnou roztečí.

Uspořádání jednotlivých částí textu musí být přehledné a logické. Je třeba odlišit názvy kapitol a podkapitol – píšeme je malými písmeny kromě velkých začátečních písmen. U jednotlivých odstavců textu odsazujeme první řádek odstavce asi o jeden až dva čtverčíky (vždy o stejnou, předem zvolenou hodnotu), tedy přibližně o dvě šířky velkého písmene M základního textu. Poslední řádek předchozího odstavce a první řádek následujícího odstavce se v takovém případě neoddělují svislou mezerou. Proklad mezi těmito řádky je stejný jako proklad mezi řádky uvnitř odstavce.

Při vkládání obrázků volte jejich rozměry tak, aby nepřesáhly oblast, do které se tiskne text (tj. okraje textu ze všech stran). Pro velké obrázky vyčleňte samostatnou stránku. Obrázky nebo tabulky o rozměrech větších než A4 umístěte do písemné zprávy formou skládanky vřité do přílohy nebo vložené do záložek na zadní desce.

Obrázky i tabulky musí být pořadově očíslovány. Číslování se volí buď průběžné v rámci celého textu, nebo – což bývá praktičtější – průběžné v rámci kapitoly. V druhém případě se číslo tabulky nebo obrázku skládá z čísla kapitoly a čísla obrázku/tabulky v rámci kapitoly – čísla jsou oddělena tečkou. Čísla podkapitol nemají na číslování obrázků a tabulek žádný vliv.

Tabulky a obrázky používají své vlastní, nezávislé číselné řady. Z toho vyplývá, že v odkazech uvnitř textu musíme kromě čísla udát i informaci o tom, zda se jedná o obrázek či tabulku (například “... viz tabulka 2.7 ...”). Dodržování této zásady je ostatně velmi přirozené.

Pro odkazy na stránky, na čísla kapitol a podkapitol, na čísla obrázků a tabulek a v dalších podobných příkladech využíváme speciálních prostředků DTP programu, které zajistí vygenerování správného čísla i v případě, že se text posune díky změnám samotného textu nebo díky úpravě parametrů sazby. Příkladem takového prostředku v systému LaTeX je odkaz na číslo odpovídající umístění značky v textu, například návěští (`\ref{navesti}`) – podle umístění návěští se bude jednat o číslo kapitoly, podkapitoly, obrázku, tabulky nebo podobného číslovaného prvku), na stránku, která obsahuje danou značku (`\pageref{navesti}`), nebo na literární odkaz (`\cite{identifikator}`).

Rovnice, na které se budeme v textu odvolávat, opatříme pořadovými čísly při pravém okraji příslušného řádku. Tato pořadová čísla se píší v kulatých závorkách. Číslování rovnic může být průběžné v textu nebo v jednotlivých kapitolách.

Jste-li na pochybách při sazbě matematického textu, snažte se dodržet způsob sazby definovaný systémem LaTeX. Obsahuje-li vaše práce velké množství matematických formulí, doporučujeme dát přednost použití systému LaTeX.

Mezeru neděláme tam, kde se spojují číslice s písmeny v jedno slovo nebo v jeden znak – například *25krát*.

Členicí (interpunkční) znaménka tečka, čárka, středník, dvojtečka, otazník a vykřičník, jakož i uzavírací závorky a uvozovky se přimykají k předcházejícímu slovu bez mezery. Mezera se dělá až za nimi. To se ovšem netýká desetinné čárky (nebo desetinné tečky). Otevírací závorka a přední uvozovky se přimykají k následujícímu slovu a mezera se vynechává před nimi – (takto) a “takto”.

Pro spojovací a rozdělovací čárku a pomlčku nepoužíváme stejný znak. Pro pomlčku je vyhrazen jiný znak (delší). V systému TeX (LaTeX) se spojovací čárka zapisuje jako jeden znak “pomlčka” (například “Brno–město”), pro sázení textu ve smyslu intervalu nebo dvojic, soupeřů a podobně se ve zdrojovém textu používá dvojice znaků “pomlčka” (například “zápas Sparta – Slavie”; “cena 23–25 korun”), pro výrazné oddělení části věty, pro výrazné oddělení vložené věty, pro vyjádření nevyslovené myšlenky a v dalších situacích (viz Pravidla českého pravopisu) se používá nejdelší typ pomlčky, která se ve zdrojovém textu zapisuje jako trojice znaků “pomlčka” (například “Další pojem — jakkoliv se může zdát nevýznamný — bude neformálně definován v následujícím odstavci.”). Při sazbě matematického mínus se při sazbě používá rovněž odlišný znak. V systému TeX je ve zdrojovém textu zapsán jako normální mínus (tj. znak “pomlčka”). Sazba v matematickém prostředí, kdy se vzoreček uzavírá mezi dolary, zajistí vygenerování správného výstupu.

Lomítko se píše bez mezer. Například školní rok 2008/2009.

Pravidla pro psaní zkratk jsou uvedena v Pravidlech českého pravopisu [1]. I z jiných důvodů je vhodné, abyste tuto knihu měli po ruce.

4.1 Co to je normovaná stránka?

Pojem *normovaná stránka* se vztahuje k posuzování objemu práce, nikoliv k počtu vytištěných listů. Z historického hlediska jde o počet stránek rukopisu, který se psal psacím strojem na speciální předtištěné formuláře při dodržení průměrné délky řádku 60 znaků a při 30 řádcích na stránku rukopisu. Vzhledem k zápisu korekturních značek se používalo řádkování 2 (ob jeden řádek). Tyto údaje (počet znaků na řádek, počet řádků a proklad

mezi nimi) se nijak nevztahují ke konečnému vytištěnému výsledku. Používají se pouze pro posouzení rozsahu. Jednou normovanou stránkou se tedy rozumí $60 \cdot 30 = 1800$ znaků. Obrázky zařazené do textu se započítávají do rozsahu písemné práce odhadem jako množství textu, které by ve výsledném dokumentu potisklo stejně velkou plochu.

Orientační rozsah práce v normostranách lze v programu Microsoft Word zjistit pomocí funkce *Počet slov* v menu *Nástroje*, když hodnotu *Znaky (včetně mezer)* vydělíte konstantou 1800. Do rozsahu práce se započítává pouze text uvedený v jádru práce. Části jako abstrakt, klíčová slova, prohlášení, obsah, literatura nebo přílohy se do rozsahu práce nepočítají. Je proto nutné nejdříve označit jádro práce a teprve pak si nechat spočítat počet znaků. Přibližný rozsah obrázků odhadnete ručně. Podobně lze postupovat i při použití OpenOffice. Při použití systému LaTeX pro sazbu je situace trochu složitější. Pro hrubý odhad počtu normostran lze využít součet velikostí zdrojových souborů práce podělený konstantou cca 2000 (normálně bychom dělili konstantou 1800, jenže ve zdrojových souborech jsou i vyznačovací příkazy, které se do rozsahu nepočítají). Pro přesnější odhad lze pak vyextrahovat holý text z PDF (např. metodou cut-and-paste nebo *Save as Text...*) a jeho velikost podělit konstantou 1800.

Kapitola 5

Závěr

Závěrečná kapitola obsahuje zhodnocení dosažených výsledků se zvlášť vyznačeným vlastním přínosem studenta. Povinně se zde objeví i zhodnocení z pohledu dalšího vývoje projektu, student uvede náměty vycházející ze zkušeností s řešeným projektem a uvede rovněž návaznosti na právě dokončené projekty.

Literatura

- [1] Hlavsa, Z.; aj.: *Pravidla českého pravopisu*. Academia, 2005, ISBN 80-200-1327-X.

Příloha A

Jak pracovat s touto šablonou

V této kapitole je uveden popis jednotlivých částí šablony, po kterém následuje stručný návod, jak s touto šablonou pracovat.

Jedná se o přechodnou verzi šablony. Nová verze bude zveřejněna do konce roku 2016 a bude navíc obsahovat nové pokyny ke správnému využití šablony, závazné pokyny k vypracování bakalářských a diplomových prací (rekapitulace pokynů, které jsou dostupné na webu) a nezávazná doporučení od vybraných vedoucích. Jediné soubory, které se v nové verzi změní, budou `projekt-01-kapitoly-chapters.tex` a `projekt-30-prilohy-appendices.tex`, jejichž obsah každý student vymaže a nahradí vlastním. Šablonu lze tedy bez problémů využít i v současné verzi.

Popis částí šablony

Po rozbalení šablony naleznete následující soubory a adresáře:

bib-styles Styly literatury (viz níže).

obrazky-figures Adresář pro Vaše obrázky. Nyní obsahuje `placeholder.pdf` (tzv. TODO obrázek, který lze použít jako pomůcku při tvorbě technické zprávy), který se s prací neodevzdává. Název adresáře je vhodné zkrátit, aby byl jen ve zvoleném jazyce.

template-fig Obrázky šablony (znak VUT).

fitthesis.cls Šablona (definice vzhledu).

Makefile Makefile pro překlad, počítání normostran, sbalení apod. (viz níže).

projekt-01-kapitoly-chapters.tex Soubor pro Váš text (obsah nahradte).

projekt-20-literatura-bibliography.bib Seznam literatury (viz níže).

projekt-30-prilohy-appendices.tex Soubor pro přílohy (obsah nahradte).

projekt.tex Hlavní soubor práce – definice formálních částí.

Výchozí styl literatury (`czechiso`) je od Ing. Martínka, přičemž anglická verze (`englishiso`) je jeho překladem s drobnými modifikacemi. Oproti normě jsou v něm určité odlišnosti, ale na FIT je dlouhodobě akceptován. Alternativně můžete využít styl od Ing. Radima Loskota

nebo od Ing. Radka Pyšného¹. Alternativní styly obsahují určitá vylepšení, ale zatím nebyly řádně otestovány větším množstvím uživatelů. Lze je považovat za beta verze pro zájemce, kteří svoji práci chtějí mít dokonalou do detailů a neváhají si nastudovat detaily správného formátování citací, aby si mohli ověřit, že je vysázený výsledek v pořádku.

Makefile kromě překladu do PDF nabízí i další funkce:

- přejmenování souborů (viz níže),
- počítání normostran,
- spuštění vlny pro doplnění nezlomitelných mezer,
- sbalení výsledku pro odeslání vedoucímu ke kontrole (zkontrolujte, zda sbalí všechny Vámi přidané soubory, a případně doplňte).

Nezapomeňte, že vlna neřeší všechny nezlomitelné mezery. Vždy je třeba manuální kontrola, zda na konci řádku nezůstalo něco nevhodného – viz Internetová jazyková příručka².

Pozor na číslování stránek! Pokud má obsah 2 strany a na 2. jsou jen „Přílohy“ a „Seznam příloh“ (ale žádná příloha tam není), z nějakého důvodu se posune číslování stránek o 1 (obsah „nesedí“). Stejný efekt má, když je na 2. či 3. stránce obsahu jen „Literatura“ a je možné, že tohoto problému lze dosáhnout i jinak. Řešení je několik (od úpravy obsahu, přes nastavení počítadla až po sofistikovanější metody). **Před odevzdáním proto vždy přezkontrolujte číslování stran!**

Doporučený postup práce se šablonou

1. **Zkontrolujte, zda máte aktuální verzi šablony.** Máte-li šablonu z předchozího roku, na stránkách fakulty již může být novější verze šablony s aktualizovanými informacemi, opravenými chybami apod.
2. **Zvolte si jazyk,** ve kterém budete psát svoji technickou zprávu (česky, slovensky nebo anglicky) a svoji volbu konzultujte s vedoucím práce (nebyla-li dohodnuta předem). Pokud Vámi zvoleným jazykem technické zprávy není čeština, nastavte příslušný parametr šablony v souboru projekt.tex (např.: `documentclass[english]{fitthesis}`) a přeložte prohlášení a poděkování do angličtiny či slovenštiny.
3. **Přejmenujte soubory.** Po rozbalení je v šabloně soubor projekt.tex. Pokud jej přeložíte, vznikne PDF s technickou zprávou pojmenované projekt.pdf. Když vedoucímu více studentů pošle projekt.pdf ke kontrole, musí je pracně přejmenovávat. Proto je vždy vhodné tento soubor přejmenovat tak, aby obsahoval Váš login a (případně zkrácené) téma práce. Vyhněte se však použití mezer, diakritiky a speciálních znaků. Vhodný název tedy může být např.: „xlogin00-Cisteni-a-extrakce-textu.tex“. K přejmenování můžete využít i přiložený Makefile:

```
make rename NAME=xlogin00-Cisteni-a-extrakce-textu
```

¹BP Ing. Radka Pyšného <http://www.fit.vutbr.cz/study/DP/BP.php?id=7848>

²Internetová jazyková příručka <http://prirucka.ujc.cas.cz/?id=880>

4. Vyplňte požadované položky v souboru, který byl původně pojmenován `projekt.tex`, tedy typ, rok (odevzdání), název práce, svoje jméno, ústav (dle zadání), tituly a jméno vedoucího, abstrakt, klíčová slova a další formální náležitosti.
5. Nahraďte obsah souborů s kapitolami práce, literaturou a přílohami obsahem svojí technické zprávy. Jednotlivé přílohy či kapitoly práce může být výhodné uložit do samostatných souborů – rozhodnete-li se pro toto řešení, je doporučeno zachovat konvenci pro názvy souborů, přičemž za číslem bude následovat název kapitoly.
6. Nepotřebujete-li přílohy, zakomentujte příslušnou část v `projekt.tex` a příslušný soubor vyprázdníte či smažete. Nesnažte se prosím vymyslet nějakou neúčelnou přílohu jen proto, aby daný soubor bylo čím naplnit. Vhodnou přílohou může být obsah přiloženého paměťového média.
7. Nascanované zadání uložte do souboru `zadani.pdf` a povolte jeho vložení do práce parametrem šablony v `projekt.tex` (`\documentclass[zadani]{fitthesis}`).
8. Nechcete-li odkazy tisknout barevně (tedy červený obsah – bez konzultace s vedoucím nedoporučuji), budete pro tisk vytvářet druhé PDF s tím, že nastavíte parametr šablony pro tisk: (`\documentclass[zadani,print]{fitthesis}`). Barevné logo se nesmí tisknout černobíle!
9. Vzor desek, do kterých bude práce vyvázána, si vygenerujte v informačním systému fakulty u zadání. Pro disertační práci lze zapnout parametrem v šabloně (více naleznete v souboru `fitthesis.cls`).
10. Nezapomeňte, že zdrojové soubory i (obě verze) PDF musíte odevzdat na CD či jiném médiu přiloženém k technické zprávě.

Pokyny pro oboustranný tisk

- Zapíná se parametrem šablony: `\documentclass[twoside]{fitthesis}`
- Po vytištění oboustranného listu zkontrolujte, zda je při prosvícení sazební obrazec na obou stranách na stejné pozici. Méně kvalitní tiskárny s duplexní jednotkou mají často posun o 1–3 mm. Toto může být u některých tiskáren řešitelné tak, že vytisknete nejprve liché stránky, pak je dáte do stejného zásobníku a vytisknete sudé.
- Za titulním listem, obsahem, literaturou, úvodním listem příloh, seznamem příloh a případnými dalšími seznamy je třeba nechat volnou stránku, aby následující část začínala na liché stránce (`\cleardoublepage`).
- Konečný výsledek je nutné pečlivě přezkontrolovat.

Užitečné nástroje

Následující seznam není výčtem všech využitelných nástrojů. Máte-li vyzkoušený osvědčený nástroj, neváhejte jej využít. Pokud však nevíte, který nástroj si zvolit, můžete zvážit některý z následujících:

MikTeX L^AT_EX pro Windows – distribuce s jednoduchou instalací a vynikající automatizací stahování balíčků.

TeXstudio Přenositelné opensource GUI pro \LaTeX . Ctrl+klik umožňuje přepínat mezi zdrojovým textem a PDF. Má integrovanou kontrolu pravopisu, zvýraznění syntaxe apod. Pro jeho využití je nejprve potřeba nainstalovat MikTeX.

JabRef Pěkný a jednoduchý program v Javě pro správu souborů s bibliografií (literaturou). Není potřeba se nic učit – poskytuje jednoduché okno a formulář pro editaci položek.

InkScape Přenositelný opensource editor vektorové grafiky (SVG i PDF). Vynikající nástroj pro tvorbu obrázků do odborného textu. Jeho ovládnutí je obtížnější, ale výsledky stojí za to.

GIT Vynikající pro týmovou spolupráci na projektech, ale může výrazně pomoci i jednomu autorovi. Umožňuje jednoduché verzování, zálohování a přenášení mezi více počítači.

Overleaf Online nástroj pro \LaTeX . Přímě zobrazuje náhled a umožňuje jednoduchou spolupráci (vedoucí může průběžně sledovat psaní práce), vyhledávání ve zdrojovém textu kliknutím do PDF, kontrolu pravopisu apod. Zdarma jej však lze využít pouze s určitými omezeními (někomu stačí na disertaci, jiný na ně může narazit i při psaní bakalářské práce) a pro dlouhé texty je pomalejší.

Užitečné balíčky pro \LaTeX

Studenti při sazbě textu často řeší stejné problémy. Některé z nich lze vyřešit následujícími balíčky pro \LaTeX :

- `amsmath` – rozšířené možnosti sazby rovnic,
- `float`, `afterpage`, `placeins` – úprava umístění obrázků,
- `fancyvrb`, `alltt` – úpravy vlastností prostředí Verbatim,
- `makecell` – rozšíření možností tabulek,
- `pdflscape`, `rotating` – natočení stránky o 90 stupňů (pro obrázek či tabulku),
- `hyphenat` – úpravy dělení slov,
- `picture`, `epic`, `eepic` – přímé kreslení obrázků.

Některé balíčky jsou využity přímo v šabloně (v dolní části souboru `fitthesis.cls`). Nahlednutí do jejich dokumentace může být rovněž užitečné.

Sloupec tabulky zarovnaný vlevo s pevnou šířkou je v šabloně definovaný „L“ (používá se jako „p“).

Příloha B

Ukazka rozsireni sablony

B.1 Definice v projekt.tex

Kazde nastaveni sablony se hodi pro neco jineho. Minamlne pouzijete tyto:

- `\documentclass[]{\fitthesis}`
- `\documentclass[zadani,print]{\fitthesis}`

Experimentujte a vyzkousejte i zbyte moznosti napr seznamem obrazku, tabulek, atd...

```
\documentclass[]{\fitthesis} % bez zadání - pro začátek práce, aby nebyl
                             problém s překladem
%\documentclass[english]{\fitthesis} % without assignment - for the work
                             start to avoid compilation problem
%\documentclass[zadani]{\fitthesis}
% odevzdani do wisu - odkazy jsou barevné
%\documentclass[english,zadani]{\fitthesis} % for submission to the
                             IS FIT - links are color
%\documentclass[zadani,print]{\fitthesis} % pro tisk - odkazy jsou černé
%\documentclass[english,zadani,print]{\fitthesis} % for the print - links
                             are black
%\documentclass[listFigures,listTables,listAppendices]{\fitthesis}
% s listem obrazku, tabulek a priloh
```

B.2 Ukazky kodu

Ukazka kodu pro Microsoft Network monitor v jazyce NPL [Listing B.1](#).

```
1  //[RegisterAfter(TCPPayload.HTTP, YMSG, 5050)]
2  //[RegisterAfter(PayloadHeader.LLC, YMSG, YMSG)]
3  Protocol YMSG = FormatString("Type = %s (%d), status = %s (%d)",
4                             YMSGTypes(Type),Type, YMSGHEADStatus(Status),Status)
5  {
6      AsciiString(4) YMSGconst;
7      UINT16 Version;
8      UINT16 VendorID;
9      [Post.Properties.Length = Length]
10     UINT16 Length;
11     UINT16 Type = FormatString("%s (%#04x)", YMSGTypes(Type), Type);
12     UINT32 Status = FormatString("%s (%#04x)", YMSGHEADStatus(this), this);
13     UINT32 Session;
14
15     /* [BuildConversationWithParent(Session)] StartPayload
16        [YMSGPayload = Blob(FrameData, FrameOffset, Length),
17        DataFieldFrameLength = frameOffset + Length,
18        PayloadStart (
19        NetworkDirection, //direction
20        0, // id
21        0, // sequence token
22        0, // next sequence token
23        Length, // total payload length
24        !Property.TCPContinuation, // is first
25        (TCP.Flags.Push || TCP.Flags.Fin || TCP.Flags.Urgent),//is last
26        RssmblyIndStartBit + RssmblyIndEndBit // Properties... )]*/
27
28     [HeaderOffset = FrameOffset]
29     TVs tvs;
30     status = "%s (%d)", YMSGTypes(Type),Type,YMSGHEADStatus(Status),Status)]
31     switch{
32         case (FrameLength - FrameOffset - 1 > 0) && (HeaderOffset <=
33             FrameOffset):
34             [MultiYMSG = "YES"]
35             StringTerm(0, "YMSG", 1, 0, 0 ) blank0 = FormatString("Optional
36                 stuffing between multiple YMSG messages", ymsg);
37             YMSG ymsg;
38         case (FrameLength - FrameOffset > 0) && (HeaderOffset <= FrameOffset):
39             BLOB(FrameLength - FrameOffset) blank1 = FormatString("Optional
40                 terminator of YMSG message", this);
41     }
42 }
```

Listing B.1: Example of not complete NPL description of YMSG IM protocol. Commented parts are not supported by NPlangCompiler.

Ukazka XML vstupu ..., Listing B.2.

```
<?xml version="1.0" encoding="utf-8"?>
<log>
  <user guid="192.168.2.101">
    <protocol val="ICQ">
      <event validity="valid">
        <src version="IPv4" port="49874">192.168.2.101</src>
        <dst version="IPv4" port="5190">205.188.10.251</dst>
        <conversationPart type="statusChange" timeStamp="2013-08-08T02:06:19.518767"
          frameNumber="86" flowDirection="up" statusType="online" />
        <conversationPart type="contactList" timeStamp="2013-08-08T02:06:39.318927"
          frameNumber="113" flowDirection="down">
          <group id="0" name="">
            <contact id="3" firstName="277264821"
              nick="Contact1" />
          </group>
          <group id="1" name="rename_group" />
          <group id="3" name="friends_group">
            <contact id="9327" firstName="647175775"
              nick="Contact2" />
            <contact id="2" firstName="284569266"
              nick="Contact3" />
            <contact id="29425" firstName="312345170"
              nick="Contact4" />
          </group>
        </conversationPart>
        <conversationPart type="message"
          timeStamp="2013-08-08T02:11:18.357794" frameNumber="791" flowDirection="up">
          <![CDATA[<HTML><BODY dir="ltr"><FONT size="2">test</FONT></BODY></HTML>]]>
          <receiver>310451170</receiver>
        </conversationPart>
        <conversationPart type="contactListChange"
          timeStamp="2013-08-08T02:32:11.914733" frameNumber="1528" flowDirection="up"
          contactListChangeAction="add">
          <contact id="31432670" />
        </conversationPart>
        <conversationPart type="authorizationReply"
          timeStamp="2013-08-08T02:32:55.885157"
          frameNumber="1543" flowDirection="down" sender="310451170"
          authorizationStatus="permit" />
      </protocol>
    </user>
  </log>
```

Listing B.2: Example of IMSleuths's xml log used as input in ContextBrowser.

```
#include<stdio.h>
#include<iostream>
int main(void)
{
  printf("Hello World\n");
  // comment
  return 0;
}
```

Listing B.3: Basic C code.

B.3 TODO

Kdyz pouzijete `TODO` . Pro verzi do WISu nebo tisk zmente nastaveni v `fitthesis.cls`

```
\usepackage[colorinlistoftodos,prependcaption,textsize=tiny]{todonotes}
```

Neco bych
mel dopl-
nit...