

# Zpravodaj Československého sdružení uživatelů TeXu

---

Barbara Beeton

Ladění LaTeXových souborů

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 31 (2021), No. 1-4, 63–75

Persistent URL: <http://dml.cz/dmlcz/150296>

## Terms of use:

© Československé sdružení uživatelů TeXu, 2021

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:  
*The Czech Digital Mathematics Library* <http://dml.cz>

Každý uživatel L<sup>A</sup>T<sub>E</sub>Xu se alespoň jednou během své práce setkal s problémem, kdy z nějakého důvodu L<sup>A</sup>T<sub>E</sub>X dokument nepřeloží. Jak se vypořádat s jednoduchými problémy v L<sup>A</sup>T<sub>E</sub>Xu, je poměrně dobře známo, ale jsou i situace, na které běžné metody nestačí.

V tomto článku si představíme postupy řešení několika typů problémů, které se objevily během dlouholeté praxe člena technické podpory Americké matematické společnosti, který vyřizuje dotazy jak od autorů, tak i od samotných editorů. Podíváme se na obvyklé i méně časté problémy, se kterými se při své práci setkává. A se kterými se může setkat každý uživatel L<sup>A</sup>T<sub>E</sub>Xu.

**Klíčová slova:** L<sup>A</sup>T<sub>E</sub>X, ladění, chyby, log soubor

## 1. Úvod

V roce 2016 Americká matematická společnost publikovala řádově 60 000 stran knih a časopisů. Většina z nich byla dodána v L<sup>A</sup>T<sub>E</sub>Xových souborech, které vytvořili přímo příslušní autoři. Aby mohl být dokument přijat k tisku, musí mít vědecký přínos a musí být posouzen recenzenty. Autor může článek dodat ve formě elektronické nebo papírové, dokonce může být napsán i ručně. Není důležitý vzhled dokumentu, ale pouze jeho obsah. Knihy jsou smluvně zajišťovány zaměstnanci, z nichž všichni jsou profesionální matematici obeznámení s L<sup>A</sup>T<sub>E</sub>Xem, ale v žádném případě to nejsou odborníci na T<sub>E</sub>X. S veškerými příchozími dokumenty se pak redakce musí nějak vypořádat.

Jestliže dodaný dokument není napsán v L<sup>A</sup>T<sub>E</sub>Xu, přepíše jej do L<sup>A</sup>T<sub>E</sub>Xu nějaká kompetentní osoba. Proto dále v článku předpokládáme, že dokument již je v L<sup>A</sup>T<sub>E</sub>Xu.

Kvalita dodaných zdrojových souborů se napříč dokumenty velmi liší. To poskytuje velké množství příležitostí k testování a vylepšování ladicích dovedností členů redakce.

---

Z anglického originálu [1] přeložili Lucie Šustková a Jan Šustek.

Redakční práce probíhají na síti počítačů s Linuxem. Použité balíky maker spadají do tří skupin:

- balíčky patřící do T<sub>E</sub>Xlive, které se aktualizují maximálně jednou ročně,
- lokální upravené verze balíčků a fontů (někdy také nové verze balíčků, které budou v následujícím vydání T<sub>E</sub>Xlive),
- vlastní lokální makra AMS.

Všechny soubory jsou archivovány v systému Subversion. Archivy publikovaných knih a článků sahají desítky let do minulosti. U každé publikace jsou balíčkem snapshot zaznamenány aktuální verze L<sup>A</sup>T<sub>E</sub>Xu a všech použitých souborů – v případě potřeby je možné reprodukovat původní soubory a definice. Tento postup práce poskytuje stabilitu při zpracování dotisků, při dalších vydáních knih a při převodu existujících publikací do jiných formátů, jako jsou například elektronické knihy.

Výše uvedený postup však může být funkční a spolehlivý až v okamžiku, kdy jsou všechny soubory bez chyb a dokument je připravený k vytištění. Ale před tímto šťastným okamžikem se může přihodit spousta věcí. O nich pojednává tento článek, který navazuje na přednášku autorky na konferenci TUG@BachoT<sub>E</sub>X 2017.

V případě libovolné chyby v dokumentu platí hlavní obecná zásada: dostat se rychle a spolehlivě zpět do stavu, kdy ještě chyba nenastala.

## 2. Příprava dokumentů – plánujte dopředu

Existují určité zásady, které, pokud se důsledně dodržují, mohou z dlouhodobého hlediska dost usnadnit práci. První z nich je vybrat si vhodné nástroje a naučit se je používat.

Nejdůležitějším nástrojem je kvalitní textový editor nebo vývojové prostředí. Autorka tohoto článku používá emacs. Jsou však jiné možnosti. Některé jsou vhodné pro práci jediného uživatele na jednom počítači, jiné pro společnou práci více autorů online, a spousta dalších někde mezi tím. Seznam možných nástrojů lze najít v odpovědi na otázku položenou na T<sub>E</sub>X Stack Exchange [2] (dále jen „`tex.sx`“).<sup>1</sup>

Autorka tohoto článku dává přednost zpracování souborů z příkazového řádku. To umožňuje interaktivně opravit jednoduché chyby, jako jsou překlepy v názvech řídicích sekvencí, a tím se vyhnout hromadě navazujících chybových hlášení způsobených jedinou triviální chybou. Samozřejmě je v takovém případě důležité před dalším překladem stejným způsobem opravit zdrojový soubor.

Mezi funkce nejužitečnější pro ladění patří tyto:

- dobré možnosti vyhledávání,
- párování závorek a `\begin–\end`,

---

<sup>1</sup>L<sup>A</sup>T<sub>E</sub>X Editors/IDEs, <http://tex.stackexchange.com/q/339>

- více oken viditelných současně,
- možnost přejít na konkrétní číslo řádku,
- určení počtu nalezených řetězců.

Dále je třeba dát si pozor na to, jak jsou adresáře a soubory umístěny a pojmenovány. Je vhodné se vyhnout mezerám v názvech souborů – ne všechny operační systémy dokážou takové soubory zpracovat. Podobně některé operační systémy rozlišují velká a malá písmena – aby se předešlo problémům, je doporučeno používat pro názvy souborů pouze malá písmena, případně číslice a spojovníky. Je dobré se vyhnout tečkám a znakům, které mají v  $\text{\TeX}$ u zvláštní význam (například podtržítko).

Vyplatí se udržovat soubory v přijatelné velikosti. Pro velký dokument, například knihu nebo diplomovou práci, umístěte každou kapitolu do samostatného souboru a mějte jeden řídicí soubor, který bude jednotlivé soubory načítat makrem `\include`. S využitím  $\text{\LaTeX}$ ového makra `\includeonly` vám to umožní pracovat v daný čas pouze na jedné kapitole. Pokud máte velké tabulky nebo obrázky, opět se vyplatí umístit je do samostatných souborů, protože pak můžeme celou tabulku nebo obrázek skrýt jediným procentem. (Další výhodou je jednodušší přesun na jiné místo v dokumentu v případě potřeby.)

Když editujete soubory, bývá dobrým zvykem ukončit načítané soubory příkazem `\endinput` umístěným na samostatném řádku. Tím se vyhnete problémům s načítáním případných dalších znaků, které se neohlášeně někdy přidávají na konec souboru, když se tento soubor přenáší z jednoho systému na jiný. Do načítaných souborů však nikdy nevkládejte `\end{document}`.

Zjistěte si, kde se nachází log soubor. Některá vývojová prostředí jej před uživatelem ukrývají. Když se vám začne hroudit překlad a vy se nebudete moct podívat do log souboru a zjistit, v čem je problém, pak vás čekají hodně horké chvíle, než se vám podaří problém opravit.

A nikdy neaktualizujte svůj systém při práci na důležitém projektu. Nové verze balíčků mohou mít nové nekompatibilní vlastnosti a staré balíčky již nemusejí fungovat. Tato rada je samozřejmě k ničemu, když vám zrovna doslouží hardware. Ale vy si přece děláte průběžné zálohy všech souborů, že ano?

### 3. Testujte ve vedlejších souborech

Jestliže chyba je složitější než jen překlep v názvu řídicí sekvence, snažte se předejít možným komplikacím – udělejte si svoje místo pro ladění. Ze všeho nejdříve zálohujte své soubory nebo celou pracovní adresářovou strukturu a zálohy umístěte na bezpečné místo. Vy víte, jaký je aktuální stav vašeho dokumentu, a do tohoto stavu byste se rádi zpátky dostali, pokud by ladění neprobíhalo úspěšně. Za žádných okolností nedělejte experimentální změny v souborech, od nichž nemáte kopii. Úplně nejlépe si vytvořte nový adresář a dělejte experimenty v něm.

Jestliže se dokument skládá z více než jednoho zdrojového souboru, začnete zkopírováním pouze řídicího souboru – toho, který načítá další soubory – do tohoto nového adresáře. K dalším souborům přistupujte pomocí odkazů. Na Linuxu je na to příkaz<sup>2</sup>

```
ln -s <soubor> <odkaz>
```

Dokument překládejte L<sup>A</sup>T<sub>E</sub>Xem interaktivně. Můžete tak snadno opravit jednoduché chyby ještě předtím, než zapříčiní další chyby a nejasná chybová hlášení. (Nezapomeňte pak opravit chybu jak v testovacím, tak v původním souboru.) A jestliže je chyba složitější a nelze opravit interaktivně (jako například neznámé nebo neukončené prostředí), pak lze úlohu ihned přerušit a před pokračováním problém vyřešit.

Pokud dokument překládáte v nonstop módu (což je obvyklá situace, když pracujete ve vývojovém prostředí), pak se seznam všech chyb (do maximálního počtu 100 chyb v jednom odstavci) zapíše do log souboru. I jediná chyba ale může způsobit množství dalších chybových situací a hlášení, které by nevznikly, kdyby nedošlo k prvotní chybě.

## 4. Některé nástroje pro interaktivní diagnostiku

V T<sub>E</sub>Xu jsou k dispozici následující diagnostické příkazy, které požadovanou informaci vypíší na terminál i do log souboru:

- Příkaz `\message{...}` vypíše danou zprávu na terminál i do log souboru. Může být použitý k informování uživatele, že se překlad dokumentu dostal až do konkrétního místa. Například

```
\message{poslední sekce, strana \number\thepage^^J}
```

vypíše

```
poslední sekce, strana 66
```

- Příkaz `\show` vypíše aktuální význam tokenu. Zpracování dokumentu se pozastaví a čeká se na uživatelskou interakci. Například

```
\show\LaTeX
```

vypíše

```
> \LaTeX=macro:
->\protect \LaTeX .
```

---

<sup>2</sup>Jak v jiných operačních systémech udělat odkaz na soubor, to si čtenář může snadno dohledat na internetu.

Nebo

```
\show\protect
```

vypíše

```
> \protect=\relax.
```

Když uživatel odpoví `i`, může na aktuální místo zdrojového souboru vložit konkrétní text. Když pouze stiskne `enter`, bude překlad dokumentu pokračovat.

- Příkaz `\showthe` vypíše aktuální hodnotu registru. Stejně jako v předchozím případě se překlad pozastaví. Například

```
\showthe\hfuzz
```

vypíše

```
> 0.1pt.
```

Další diagnostické příkazy vypíší informaci pouze do log souboru. (Tyto příkazy většinou vypíší více informací, než potřebujeme, proto je třeba je používat s rozumem.) Z těchto příkazů autorka tohoto článku nejčastěji používá následující:

- Po nastavení registru `\tracingoutput` na kladnou hodnotu se vypíše obsah všech boxů na aktuální stránce.<sup>3</sup>
- Po nastavení registrů `\tracingmacros` a `\tracingcommands` na kladnou hodnotu se vypíší detaily o prováděné expanzi maker a o prováděných příkazech hlavního procesoru.
- Registr `\errorcontextlines` určuje, kolik se maximálně vypíše řádků s detaily expanze, kterou `TEX` aktuálně v okamžiku chyby provádí. Je dobré nastavit hodnotu tohoto registru na více než 5, aby bylo možné chybu dobře diagnostikovat i v případě složité expanze.<sup>4</sup>
- Po nastavení registru `\tracingonline` na kladnou hodnotu budou všechny příkazy `\tracing...` vypisovat informace jak do log souboru, tak na terminál.

Detaily použití všech diagnostických příkazů lze najít v *T<sub>E</sub>Xbooku* [3] nebo v knize *T<sub>E</sub>X by topic*<sup>5</sup> [4], případně v češtině v *T<sub>E</sub>Xbooku naruby* [5].

---

<sup>3</sup>Množství vypsaného obsahu boxu a počet vypsaných úrovní boxů jsou určeny registry `\showboxbreadth` a `\showboxdepth`. Tyto registry je nutno nastavit na kladnou hodnotu. Implicitně jsou v `LATEXu` oba registry nastaveny na hodnotu `-1`, což znamená, že se žádný obsah boxů nevypíše. (pozn. překl.)

<sup>4</sup>Implicitně je registr `\errorcontextlines` v `LATEXu` nastaven na hodnotu `-1`, což znamená, že se žádné detaily expanze nevypíší. (pozn. překl.)

<sup>5</sup>Kniha je dostupná také v `TEXlive`, v terminálu stačí napsat `texdoc texbytopic`.

## 5. Log soubor je váš kamarád

V log souboru ( $\LaTeX$ ) jsou zaznamenány všechny prováděné aktivity – který soubor nebo font se aktuálně načítá, jaká čísla boxů a dalších registrů se přiřazují do aktuální řídicí sekvence nebo které důležité příkazy jsou právě předefinovávány. Z pohledu ladění je asi nejdůležitější, že jsou v log souboru do detailu zaznamenány chyby a je u nich uvedeno číslo řádku ve zdrojovém souboru.

Vždy si zkontrolujte, zda se v log souboru vyskytují řádky typu

```
! Undefined control sequence.  
1.457 \fobx  
      {%
```

V log souboru jsou zaznamenány také varování, avšak bez uvedení čísla řádku. Časté varování je

```
LaTeX Warning: There were undefined references.
```

Zjistit na základě varování nebo chybového hlášení, kde přesně je chyba, může být složité. Nicméně uvedená informace by vás měla nasměrovat na konkrétní řádek, kde zahájíte pátrání po chybě. Pokud vaše vývojové prostředí ukrývá log soubor před uživatelem, zjistěte si, kde tento soubor najít. A rozhodně log soubor nemažte bez toho, abyste se do něj podívali.

Ne vždy číslo řádku uvedené v chybovém hlášení odpovídá číslu řádku, kde se nachází chyba. Například v matematickém módu (ať už uzavřeném mezi dolary nebo do matematického prostředí) není možné ukončit odstavec. Proto na chybějící dolar  $\TeX$  přijde až na konci odstavce, což může být o několik řádků později. (Toto omezení je také důvodem, proč není možné vložit prázdné řádky dovnitř matematického prostředí.) Příslušné chybové hlášení při chybějícím dolaru je

```
! Missing $ inserted.
```

Jelikož je tato chyba omezená na jeden odstavec, není problém tuto chybu najít a opravit.

Také chyba uvnitř obrázku, tabulky nebo víceřádkového vzorce se zpravidla projeví až na konci příslušného prostředí a ne okamžitě na řádku s chybou. Ale opět je oblast, kde se chyba vyskytuje, relativně malá a snadná na prohledání.

Další situací, kdy  $\TeX$  nahlásí chybu daleko za jejím výskytem, je neukončená nebo nesprávně ukončená skupina, tj. část textu zahájená `{`, `\bgroup`, `\begingroup` nebo `\begin{<prostředí>}`. V případě nesprávně ukončeného prostředí  $\LaTeX$  nahlásí

```
! LaTeX Error: \begin{<prostředí1>} on input line n  
ended by \end{<prostředí2>}.
```

Toho hlášení se vypíše okamžitě, kdy L<sup>A</sup>T<sub>E</sub>X narazí na nesprávné `\end`. Uvedené číslo řádku by mělo být správné. Na druhou stranu neukončenou skupinu T<sub>E</sub>X nahlásí až na konci dokumentu. Formát hlášení je odlišný a skládá se z několika řádků.

```
(\end occurred inside a group at level a)
### simple group (level b) entered at line m ({)
### semi simple group (level c) entered at line n (\begingroup)
### bottom level
(see the transcript file for additional information)
```

V tomto hlášení číslo *a* označuje, kolik skupin zůstalo neukončených. Tokeny v závorkách na koncích řádků určují, o jaký typ skupiny se jedná. Přitom token `{` může označovat také skupinu zahájenou tokenem `\bgroup`. Číslo řádků by opět měla správně informovat, kde byla příslušná skupina zahájena.

Další možná chybová hlášení jsou uvedena v dokumentaci různých balíčků. Většinou hlášení obsahují nějaké číslo řádku a toto číslo bývá dosti blízko místu skutečné chyby. Často je možné chybu najít bez většího hledání.

Jakmile už chybu objevíte a opravíte, pak je třeba opravit i původní soubor a pokračovat v jeho překladu. Na vedlejší testovací soubory můžete zapomenout, ty již svou funkci splnily.

Co ale dělat v případě, že se dokument skládá z více souborů? Ve kterém souboru máte hledat uvedené číslo řádku? To si popíšeme v následující sekci.

Důležitým ponaučením z této sekce je: Nikdy nemažte log soubor dříve, než z něj vyčtete všechny užitečné informace.

Někteří dokonce doporučují ukládat si log soubory průběžně pod různými názvy. Díky tomu pak je možné zjistit, co vše se změnilo mezi jednotlivými běhy T<sub>E</sub>Xu.

## 6. Jeden z mnoha – ale který?

V této sekci budeme předpokládat, že chyba je v některém našem souboru, ne v balíčku.

Když v log souboru vidíte číslo řádku, nejdříve se podívejte na tento řádek v hlavním souboru. Ale když hlavní soubor má pouze 95 řádků a chyba je na řádku 2345, pak je tato rada k ničemu. V tom případě je třeba najít ten správný soubor, jehož se dané číslo řádku týká.

Udělejte si kopii log souboru a procházejte jej pozpátku od příslušného chybového hlášení. Jestliže už některé stránky byly vysázeny do pdf souboru, pak čísla těchto stránek (uvedená v hranatých závorkách, například [69]) mohou ukazovat na kapitolu a tu v ideálním případě máte ve vlastním souboru. Když toto nepomůže, promazávejte z kopie log souboru informace, které pro účely hledání nejsou potřebné.



Zprávy o přetečených boxech můžete ignorovat – tyto řádky smažte. Spárovaná dvojice závorek obvykle uzavírá název souboru a nějaký další materiál. Vyhledejte celou uzávorkovanou skupinu jako například

```
(C:/noname/debug/uvod.tex
Úvod
[1] [2]
)
```

a smažte ji. Co nakonec zůstane, je otevírací závorka následovaná názvem souboru – souboru, který byl otevřený, když byla oznámena chyba. Řádek s daným číslem se nachází v tomto souboru.

Ale co když číslo řádku bylo uvedeno až na konci dokumentu – situace s neukončenou skupinou? Tady přichází na řadu vhodně použitý `\end{document}`. Opět pracujte pouze s vedlejšími testovacími soubory a neměňte původní soubory, dokud nenajdete a neodstraníte chybu.

Začněte na konci řídicího souboru a vložte řádek `\end{document}` mezi dva řádky s `\include`. Je vhodné použít metodu bisekce – vložte `\end{document}` tak, aby zhruba polovina řádků s `\include` byla nad ním a polovina pod ním. (Více si o tom povíme v sekci 8.) Přeložte dokument  $\text{\LaTeX}$ em. Jestliže stále zůstala neukončená skupina, je problémový soubor v první polovině. Jestliže nezůstala, je ve druhé polovině. Nyní přesuňte řádek `\end{document}` doprostřed té poloviny, ve které byl problém. Takto postupujte, dokud neodhalíte problémový soubor.

Situace je komplikovanější, pokud je neukončených více skupin (tj.  $a > 1$ ), ale princip je stejný.

## 7. Pročištění souborů

V tuto chvíli již tušíte, ve kterém souboru by měla být chyba. A možná také tušíte, poblíž kterého řádku by ta chyba měla být. A nebo také máte pouze obecnou představu, kam byste se měli podívat. Aby se vám lépe pracovalo, zpracovávejte pouze jeden soubor. Všechny pro vaše účely nepotřebné věci promažte.

V řídicím souboru makrem `\includeonly` specifikujte ten podezřelý soubor, který jediný se má zpracovávat. Zakomentujte nepotřebné části kódu, které jsou mimo `\include`:

- balíčky nepotřebné pro testování,
- `\tableofcontents`
- `\printindex`
- věci související se seznamem literatury.

Pročistěte také podezřelý soubor. Nemusíte se bát, že soubor zničíte, je to pouze kopie, že? Následující věci můžete opatrně vymazat:

- řádky začínající procentem,
- řádky `\begin{comment}` a `\end{comment}` a vše mezi nimi,
- řádky mezi `\iffalse` a `\fi` včetně těchto řádků (tato konstrukce je ekvivalentní s komentářem).

Ujistěte se, že jsou všechny skupiny správně uzavřeny. To znamená zkontrolovat spárování všech `\begin` a `\end` a všech možných uzávorkování. Je užitečné tady použít funkci editoru, která zjistí počet výskytů daných řetězců. V bezchybném dokumentu by mělo platit

- počet `{` = počet `}` (pozor na řetězec `%`), který se občas ve zdrojovém souboru vyskytuje),
- počet `\begin{` = počet `\end{`,
- počet `\begingroup` = počet `\endgroup`,
- počet `\bgroup` = počet `\egroup`,
- počet `\(` = počet `\)`,
- počet `\[` = počet `\]`,
- počet `$` je sudý,
- počet `$$` je sudý.

Nyní přeložte soubor  $\text{\LaTeX}$ em a přečtěte si log soubor. Mnoho chyb je způsobeno nespárováním, proto pokud budete mít štěstí, podaří se vám takto chybu odhalit.

V dalších sekcích si ukážeme, co dělat v případě, že stále není jasné, kde je chyba.

## 8. Rozděl a panuj

Co chceme udělat, je izolovat jediný odstavec nebo nejmenší část souboru, které způsobují chybu. (Pracujte s kopií souboru a pro jistotu si vytvořte další kopii.)

Za vhodný odstavec někde v půlce souboru vložte prázdný řádek a pod něj příkaz `\endinput`. Ujistěte se, že jste takto nerozdělili `\begin` a `\end` nebo libovolnou skupinu. Přeložte dokument  $\text{\LaTeX}$ em. Pokud se chyba nevyskytne, je problém ve druhé (nezpracované) polovině souboru. Smažte první (funkční) polovinu souboru a posunujte příkaz `\endinput`, dokud nenarazíte na problémový odstavec. Jestliže je řešení problému jednoduché, problém opravte a otestujte. Poté stejnou opravu proveďte a otestujte na celé testovací kopii souboru. A až si budete jisti, že jste problém opravili, proveďte a otestujte opravu na původním souboru.

Ale co když řešení problému není jednoduché? Co když část souboru, která zůstala, je pořád příliš obrovská na to, aby se rychle podařilo najít chybu? (Například se může jednat o dlouhý důkaz, kde jsou jednotlivé kroky vloženy jako

položky výčtového prostředí.) V tom případě vložte tuto část do nového souboru a upravujte tento nový soubor. (Mnohokrát se autorce tohoto článku stalo, že upravovala testovací kopii souboru, ale řídicí soubor stále načítal původní soubor. To dokáže naštvat. Dejte si pozor, ať načítáte ten soubor, který upravujete.)

Zmenšete velikost souboru zakomentováním položek, které vypadají neškodně. Zatím ale nic nemažte – co se zdá být neškodné, může ve skutečnosti být součástí problému. Tento proces opakujte tak dlouho, dokud vám nezůstane část souboru, jejímž vymazáním se eliminuje i (ta dosud nenalezená) chyba. Tato část souboru se nazývá „minimální (ne)funkční příklad“.

Prozkoumejte, co v souboru zůstalo, a využijte záchytné body z log souboru.

Až chybu najdete a je vám jasné, jak ji opravit, pak tu opravu proveďte a otestujte ve vaši kopii souboru a až pak opravu proveďte v původním souboru. Jestliže na žádnou další chybu nenarazíte, jste na dobré cestě.

Jestliže narazíte na nějaký další problém, jste opět na startu. Ale teď už víte, jak postupovat.

## 9. Když je potřeba další pomoc

Na internetu jsou dostupné různé užitečné stránky. Je možné, že nejste první, kdo narazil na podobný problém.

Je dobré se podívat do archivu `tex.sx`. Když nenajdete nic, co by se podobalo vašemu problému, vytvořte nový dotaz. (K vytvoření dotazu je nutné se zaregistrovat.) Aby se povedlo problém vyřešit, vložte celý minimální příklad. Vymažte z něj všechny komentáře a pokud je to vhodné (a možné), „anonymizujte“ příklad vložím smyšleného textu namísto reálného textu. Udělejte příklad co nejmenší, přitom aby pořád vykazoval problém. K příkladu přiložte příslušné řádky log souboru a také komentář, o co se v příkladu pokoušíte. Účastníci fóra `tex.sx` jsou učenliví a přátelští a mají rádi záhadné problémy. K tomu ale potřebují mít dostatek informací, aby mohli experimentovat. A když jim popíšete minimální příklad, který si mohou zkopírovat a vložit k sobě, pak jim experimenty půjdou rychleji.

## 10. Chyby před `\begin{document}`

Stále jsme neřešili situaci, kdy se chyba vyskytne už před `\begin{document}`. V tom případě pomůžou následující kroky.

- Udělejte si kopii log souboru a najděte problémový soubor, podobně jako v sekci 6.
- Pokud to není balíček, dále hledejte soubor, který se balíčkem načítá.
- Máte zkušenosti s vnitřními makry a registry  $\text{\LaTeX}$ u?

Chybové hlášení na terminálu:

```
Overfull \hbox (23.1113pt too wide) in paragraph at lines 3288--3301
\OML/cmm/m/it/10.95 A$\T1/ptm/m/n/10.95 , as in
```

Příslušná část log souboru:

```
Overfull \hbox (23.1113pt too wide) in paragraph at lines 3288--3301
\OML/cmm/m/it/10.95 A$\T1/ptm/m/n/10.95 , as in Ÿ[], is a degree-
```

Příslušná část zdrojového kódu:

```
..., as in \S\ref{SS:changing}, is a degree-1 ...
```

Obrázek 1: Záhadné chybové hlášení

- Pokud zkušenosti nemáte, je ideální čas obrátit se na odborníka. Běžte na `tex.sx`. Jestliže podobný problém zatím nikdo neřešil, vytvořte nový dotaz. Budte konkrétní, vložte svůj problémový zdrojový kód, který máte před `\begin{document}` a příslušnou část log souboru.
- Pokud zkušenosti máte, pokuste se problém vyřešit sami. Podívejte se na `tex.sx`. Jestliže podobný problém zatím nikdo neřešil, obraťte se na autora balíčku.

Tímto končí rozbor problémů, které se mohou vyskytnout ve vašich souborech. V další části autorka popisuje skutečný problém, na který narazila a dlouho se s ním trápila, než se ukázalo, že to ve skutečnosti nebyl problém  $\text{\LaTeX}$ u, ale pouze se až v  $\text{\LaTeX}$ u projevil.

## 11. Opravdová záhada

Jednou za čas nás ani důkladné krokování nepřivede k vyřešení problémů. Zdůrazním dvě související věci.

- Žiji a pracuji v USA a podle toho mám počítač nastavený, tj. například používám kódování ASCII.
- Spouštím  $\text{\LaTeX}$  z terminálu a nepoužívám `\batchmode` ani `\nonstopmode`.

Při překladu jednoho dokumentu  $\text{\LaTeX}$  neustále zamrzal a s ním zamrzal i terminál. Bylo nutné spustit nový terminál a z něj  $\text{\LaTeX}$  ukončit. Poslední zobrazená informace byla o přetečeném boxu, vizte obrázek 1. Tato informace stačila k přesné lokalizaci problému. Nicméně zdrojový soubor se v daném místě zdál zcela v pořádku. Naštěstí se vytvořil log soubor, i když ne kompletní.

Po provedení výše uvedených kroků se povedlo redukovat dokument na jediný krátký odstavec. Po odstranění textu ze začátku tohoto odstavce se chyba přestala

vyskytovat. Zdálo se, že chyba souvisí s oním přetečeným boxem. Na tomto místě jsem se poradila s kolegy, kteří mají lepší znalosti operačního systému.

Po důkladném prozkoumání log souboru je vidět záhadný znak Ÿ. (Znak je na pozici "78 ve fontu `cmsy` a má unicodový kód U+0178.) Protože pracuji s anglickými dokumenty a málokdy se setkám s akcenty, nevídávám tak často znaky mimo ASCII a určitě ne v log souboru příslušném čistě anglickému textu. Ukázalo se, že problém byl způsoben systémovým prostředím, které nebylo nastaveno na vstup v kódování UTF8 a jako důsledek zamrzalo. Řešením se ukázalo vložit do souboru `.i18n` v domovském adresáři řádek

```
LANG="en_US.utf8"
```

Někdy to, co vypadá jako L<sup>A</sup>T<sub>E</sub>Xová chyba, ve skutečnosti může být chyba úplně jinde.

## 12. Drobnosti (přidáno po konferenci)

Jsou problémy, které se vyskytují relativně často a dají se dobře identifikovat, avšak najít problémové místo ve zdrojovém souboru bývá obtížné.

- Varování o chybějícím znaku

```
There is no ; in font nullfont!
```

je téměř jistě důsledkem syntaktické chyby (chybějícího středníku) v prostředí `tikzpicture`. Další chybějící znaky interpunkce odkazující se na `nullfont` také mohou souviset s chybou uvnitř `tikzpicture`.

- Podobná varování odkazující se na znaky v jiném fontu se musejí prozkoumat důkladně. V log souboru není uvedeno číslo řádku, na němž k problému došlo. Nicméně je možné se podívat, jaké je číslo poslední vysázené strany, a podle toho projít následující vysázenou stranu, v čem se text na ní liší od zdrojového textu.

## 13. Poděkování

Autorka děkuje sdružení GUST za uspořádání konference TUG'17 v Bachtoku společně s jeho každoroční konferencí. Dále děkuje účastníkům konference za jejich dotazy a užitečné připomínky.

## Odkazy

1. BEETON, Barbara. Debugging L<sup>A</sup>T<sub>E</sub>X files: Illegitimi non carborundum. *TUGboat* [online]. 2017, roč. 38, č. 2, s. 159–164 [cit. 2021-12-06]. Dostupné z: <http://www.vim.tug.org/TUGboat/tb38-2/tb119beet.pdf>.

2. *Stack Exchange* [online]. Stack Exchange [cit. 2021-12-06]. Dostupné z: <https://tex.stackexchange.com/>.
3. KNUTH, Donald E. *The T<sub>E</sub>Xbook*. Reading, MA: Addison-Wesley, 1986. Computers & Typesetting.
4. EIJKHOUT, Victor. *T<sub>E</sub>X by Topic: A T<sub>E</sub>Xnician's Reference*. Wokingham, England: Addison-Wesley, 1992.
5. OLŠÁK, Petr. *T<sub>E</sub>Xbook naruby*. Konvoj, 2001.

## Summary: Debugging L<sup>A</sup>T<sub>E</sub>X Files

Every L<sup>A</sup>T<sub>E</sub>X user has, at least once in her career, been faced with a thorny problem when compilation shuts down for some obscure reason. How to deal with simple problems is reasonably well known, but there are situations when the time-honored methods fall short.

This article presents strategies and tactics for dealing with the many types of problems that have arisen during long experience as a member of the AMS technical support staff, handling questions from authors and the editorial staff. Both common and uncommon glitches are visited, with a bias toward avoiding problems in one's own work – something for everyone.

**Keywords:** L<sup>A</sup>T<sub>E</sub>X, debugging, errors, log file

*Barbara Beeton, American Mathematical Society, Providence, RI, USA*  
*bnb@ams.org*