

# Zpravodaj Československého sdružení uživatelů TeXu

---

Jiří Rybička

Automatizovaná tvorba formulářů pomocí LaTeXu

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 3 (1993), No. 1, 24–27

Persistent URL: <http://dml.cz/dmlcz/149654>

## Terms of use:

© Československé sdružení uživatelů TeXu, 1993

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>

---

---

# Automatizovaná tvorba formulářů pomocí $\LaTeX$

---

JIŘÍ RYBIČKA

V běžné kancelářské praxi je velmi často potřeba vyplňovat různé formuláře. Blankety formulářů se koupí ve větším množství (vždy musí být určitá provozní zásoba) v tiskárně, v dnešních cenách pohybujících se od 0,40 do 0,90 Kčs/kus. V okamžiku potřeby je blanket písáčkou vložen do psacího stroje, kde nastává nepříjemný a vždy nerovný souboj mezi vkládaným textem a místem, které je v blanketu rezervováno. Stane-li se, že blanket poněkud zastará (dnes velmi často změna bankovního účtu, přečíslování ulic, změna **soudruh**  $\rightarrow$  **pan** apod.), je navíc potřeba předepsané texty škrtnout nebo přepsat novým textem, v horším případě se zbytek zásob nepoužitelných blanketů zmaří na podružné účely.

Má-li být formulář alespoň trochu k světu, stojí to za těchto podmínek mnoho úsilí — vadí překlepy, nevhodně umístěný text apod. Chyby se musí často napravovat i přepisem na nový blanket.

Přirozeným řešením uvedených problémů je instalace systému formulářů na počítači. Přípravou textu „na nečisto“ na obrazovce se automaticky vyloučí obtíže s překlepy nebo nevhodnými formulacemi v textech. Je-li navíc systém obohacen o urychlení a zautomatizování správné lokalizace vpisovaných textů, je vyřešeno mnoho těžkostí.

Konkrétní programové řešení může být provedeno nepochybně mnoha způsoby. Je však vhodné využít běžně dostupného, rozšířeného a inteligentního vybavení, které se snadno doplní o jednoduché součástky.

## Popis systému

Velmi vhodným inteligentním systémem je systém  $\TeX$ , pro tento účel byl zvolen  $\LaTeX$ , v němž jsou vysázeny všechny blankety. Uživatel do zvoleného blanketu doplní konkrétní texty, formulář se vysází a zobrazí na obrazovce. Po kontrole a případných opravách se provede výsledný tisk.

V tomto místě je nutno předeslat, že uživatel, pro nějž byl systém vyvinut, je počítačový laik, bylo tedy třeba všechny systémové záležitosti ukryt tak, aby se musel učit co nejméně ovládacích kláves.

Činnost systému je řízena hlavní dávkou operačního systému MS DOS s názvem **Form**. Dávka má jeden parametr, který specifikuje vybraný formulář. Formuláře jsou očíslovány od 1 do 17, některé z nich ještě mají varianty — např. „Dohoda o ukončení pracovního poměru“ má jednu variantu pro ženu a druhou pro muže. Varianty jsou rozlišeny písmeny, např. **5a**, resp. **5b**.

Uživatel má k dispozici tabulku, v níž jsou uvedena čísla jednotlivých formulářů. Byla zkoušena varianta systému, v níž se na úvod spustil program zobrazující nabídku všech formulářů. Po provozních zkušenostech však byl tento program zrušen, neboť výběr z nabídky byl shledán daleko těžkopádnějším než zadání čísla formuláře do parametru dávky. Tato skutečnost je potvrzením všeobecně známého faktu, že programy „přepřácané“ různými nabídkami jsou spíše na obtíž než k ulehčení práce. Jedná-li se o víceméně rutinní činnost, je výběr z nabídky oproti zadání jednoho čísla skutečně unavující. Druhou (nepřímou) výhodou takto pojatého řešení je pružnost vzhledem ke změnám v repertoáru formulářů: není nutné měnit programy, stačí přidat nový blanket a říci obsluze, že existuje další číslo.

Text řídící dávky **Form** je následující:<sup>1)</sup>

1. @echo off
2. if not exist for\%1.txt goto chyba
3. copy for\%1.txt prac.txt
4. : znovu
5. formedit prac.txt
6. call latex prac
7. if not exist prac.dvi goto znovu
8. call view prac
9. : rozhod
10. jakdal
11. if errorlevel 6 goto laser
12. if errorlevel 4 goto dotprn
13. if errorlevel 2 goto znovu
14. goto konec
15. : dotprn

---

<sup>1)</sup> V textu dávky jsou řádky pro lepší orientaci očíslovány.

```

16.  call dot prac
17.  goto rozhod
18.  : laser
19.  call las prac
20.  goto rozhod
21.  : chyba
22.  echo Číslo formuláře není udáno nebo je chybné.
23.  : konec

```

Úvodní příkazy dávky zjišťují, zda je korektně zadáno číslo formuláře. Test je proveden na základě existence blanketu zadaného formuláře. Blankety jsou uloženy v souborech s názvy `for*.txt`, kde hvězdička představuje příslušné číslo formuláře. V kladném případě je blanket zkopírován do pracovního souboru, v němž jsou prováděny všechny úpravy.

Klíčovou součástí systému je editor. V dávce je na řádku 5 volán editor `formedit`, který představuje obyčejný textový editor, u něhož jsou definována klávesová makra, usnadňující pohyb v textu. V tomto konkrétním případě se jedná o editor `dbFast`, u něhož je v konfiguraci definována klávesa `F7` pro pohyb o jedno vpisovací pole vpřed a klávesa `F8` pro pohyb o jedno pole vzad. Vpisovací pole je definováno textem `{\vstup }`, přičemž příkaz `vstup` s definicí `\def\vstup{\hskip 3ex \large \bf}` zajišťuje odstup vkládaného textu od „předtisku“ o `3ex` a zvýrazňuje písmo tak, aby bylo na formuláři na první pohled patrné, co je text blanketu a co je vpisovaný text.

Stiskem klávesy `F7`, resp. `F8` se kurzor nastaví těsně za slovo `vstup`. Na řádku následujícím je komentář, vysvětlující, jaká položka je právě editována a jaký vstupní údaj se očekává, např.:

```

...
...
v souladu s \S~65 zákoníku práce
mu bude přiznána odměna ve výši
{\vstup } Kčs
%%%%%%%% ~ výše přiznané odměny
...
...

```

Po skončení editace je vyvolána dávka `latex` pro překlad pracovního souboru do tvaru `.dvi`. Dávka `latex.bat` obsahuje volání překladače `TEX`u v dávkovém režimu (s parametrem `/b`), čímž je zamezeno „zakousnutí“ překladu. V případě výskytu chyby (zjištěno testem `if errorlevel 2`) se smaže soubor `prac.dvi` a na obrazovku se vypíše

stručná zpráva o nalezených chybách (ze souboru `prac.log` jsou vyfiltrovány řádky se specifikací chyb, tj. řádky začínající vykřičníkem a řádky popisující umístění chyby). Je-li pak v dávce `form` zjištěno, že soubor `prac.dvi` neexistuje, pokračuje se skokem zpět na editaci, v opačném případě se vyvolá prohlížeč (dávka `view`).

Po prohlédnutí formuláře na obrazovce se aktivuje program `jakdal` (řádek 10), který na obrazovce zobrazí okno se čtyřmi možnostmi volby:

tisk Laserová tiskárna
tisk Jehlová tiskárna
Oprava formuláře
Konec práce

Podle uživatelské volby vydá program `jakdal` příslušný výstupní kód, jehož hodnotu testují následující příkazy v dávce. Jestliže je zvolen tisk, aktivují se příslušné tiskové dávky (`laser`, resp. `dot`) a pak se znovu aktivuje tatáž nabídka. Při volbě `Oprava` se řízení vrátí na řádek 5 — formulář je možné opět editovat.

Program `jakdal` je programován v Turbo Pascalu. Nabídku je možné řešit mnoha různými způsoby, lze přijmout i velmi jednoduché řešení, které lze zapsat několika málo řádky. V tomto případě byl použit systém Turbo Vision, celý program sestává pouze z volání předdefinovaných procedur vytvářejících příslušnou nabídku.

Systém uchovává pouze jediný (poslední) vyplněný formulář. Z praktických zkušeností vyplývá, že uchovávání formulářů v elektronické podobě by nebylo přínosné — bylo by nutné vést agendu vyplněných formulářů a dělat spoustu jiných (servisních) činností, čemuž zdaleka neodpovídá potřeba návratu ke starším verzím.

Systém je již provozován zhruba rok, zkušenosti z provozu ukazují, že je tímto způsobem možné řešit zpracování standardních textů v kancelářské praxi k plné spokojenosti uživatelů i jejich nadřízených.