

Pracujeme v $\text{\TeX}$ u

JOSEF BARÁK

Obchodní akademie a VOŠS, Kotlářská 9, Brno

Článek se zabývá základní orientací ve formátech a distribucích typografického systému $\text{\TeX}$, přičemž se zaměřuje zejména na instalaci a nastavení windowsové distribuce $\text{MiK}\text{\TeX}$. Obsahuje srovnání s často používanou distribucí $\text{\TeX}$ Live a zmiňuje se též o problematice PostScriptu a doplňujících prvcích.

V závěru článku je uveden konkrétní postup zápisu složitějších matematických výrazů v již nainstalovaném a nastaveném prostředí.

1. Úvod

1.1 O systému $\text{\TeX}$

$\text{\TeX}$ je software pro přípravu dokumentů vysoké typografické kvality. Původně byl navržen pro sazbu matematických publikací, jeho vlastnosti se ale mohou uplatnit při sazbě libovolných textů. V obecné rovině o tomto typografickém systému poměrně podrobně pojednává [6].

$\text{\TeX}$ není typu WYSIWYG. Při přípravě $\text{\TeX}$ ovského dokumentu musí autor dokumentu pořídit zdrojový text, ve kterém je text dokumentu promíchán s příkazy, určujícími jeho grafickou podobu. Jedná se o běžný ASCII text, který může být pořízen libovolným ASCII editorem, jako je například NotePad; programy typu MS Word či WordPad jsou k pořízení zdrojového textu méně vhodné. Většinou se však užívají specializované editory určené právě pro pořizování zdrojového textu v $\text{\TeX}$ u.

Zdrojový text je zapotřebí zpracovat překladačem, který je součástí všech $\text{\TeX}$ ových distribucí. Po úspěšném přeložení vznikne soubor ve formátu dvi (Device Independent, tj. nezávislý na výstupním zařízení – v tom

smyslu, že podaří-li se jej zobrazit na obrazovce nebo vytisknout na tiskárně, výsledek má vždy v zásadě stejnou podobu limitovanou pouze parametry daného zařízení). Ten může být tzv. prohlížečem zobrazen na monitoru, vytištěn nebo zpracován dalšími programy. Prohlížeč však nemůže dokument změnit: k tomu je třeba zásah do zdrojového textu a nový překlad.

Proto klasický způsob vytvoření $\text{T}_{\text{E}}\text{X}$ ového dokumentu vypadá takto:

1. Pořízení zdrojového textu.
2. Překlad; v případě syntaktické chyby návrat k bodu 1.
3. Prohlížení výsledku; v případě potřeby doplnění či oprav návrat k bodu 1.

Výsledný dvi soubor lze snadno převést do formátu PostScript nebo také pdf.

1.2 $\text{T}_{\text{E}}\text{X}$, plain $\text{T}_{\text{E}}\text{X}$ a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{T}_{\text{E}}\text{X}$ je v základní podobě vybaven přibližně 300 vestavěnými příkazy (tzv. $\text{T}_{\text{E}}\text{X}$ -primitivy). Protože práce s využitím pouze těchto příkazů by byla velmi těžkopádná, dají se z primitivních příkazů tvořit příkazy vyšší úrovně, tedy makra. Makra můžeme ukládat do formátů nebo stylových souborů. Nejzákladnějším formátem je plain $\text{T}_{\text{E}}\text{X}$.

Velmi populární je formátová nadstavba $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, jejíž možnosti se dají dále rozšiřovat pomocí doplňkových balíčků (packages). $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ se snaží přinutit autora dokumentu, aby oddělil jeho logickou strukturu od grafické podoby. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ poskytuje snadno použitelná makra pro všechny běžné potřeby, které se při psaní dokumentu mohou vyskytnout, a stal se i jakýmsi publikačním standardem pro mnohé matematické a odborné časopisy. V neposlední řadě je $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vhodný pro začátečníky, neboť jeho zvládnutí je (např. ve srovnání s plain $\text{T}_{\text{E}}\text{X}$ em) jednodušší.

1.3 Srovnání $\text{T}_{\text{E}}\text{X}$ u se systémy typu WYSIWYG

- + vysoká typografická kvalita výsledku,
- + vynikající podpora matematické sazby,
- + vynikající podpora dlouhých dokumentů (křížové odkazy, obsah, rejstřík),
- + přenositelnost dokumentu mezi různými operačními systémy na úrovni zdrojového textu,
- + de facto standard v řadě matematických a odborných časopisů,

- + možnost práce s různými fonty, možnost tvorby vlastních fontů, podpora diakritický znamének pro různé jazyky,
- + je zdarma,
- méně komfortní pořizování zdrojového textu,
- méně pohodlná práce ne-WYSIWYG způsobem (editace-překlad-prohlížení).

1.4 Proč právě MiKTeX?

Nejčastěji se vyskytujícím operačním systémem je v současné době MS-Windows. MiKTeX je distribuce typografického systému TeX, určená právě pro platformu Win32, tj. Windows 95/98/NT/2000/XP a s plnou podporou českých dokumentů.

Kromě MiKTeXu existuje i distribuce TeXLive, podporovaná a šířená např. organizací CSTUG (Czechoslovak TeX Users Group). Tato distribuce je určena mj. i pro platformu Win32, její ISO obrazy jsou ke stažení např. na <http://ftp.cstug.cz/pub/tex/CTAN/systems/texlive/Images/>.

Výhodou distribuce TeXLive je jednoduchost instalace. Uživatel (použije-li verzi šířenou CSTUGem) v podstatě nemusí provádět žádné dodatečné úpravy a nastavení.

Existuje ale několik důvodů, proč se článek zabývá především MiKTeXem, nejdůležitější jsou asi tyto dva:

- je kompaktnější a zabírá méně místa na disku,
- zatímco TeXLive obsahuje prohlížeč dvi souborů WinDvi, MiKTeX podporuje prohlížeč YaP, který nabízí další zajímavé funkce.

1.5 TeX ve škole

TeX je vhodný zejména pro použití v matematice a přírodovědných předmětech. Řada vyučujících a dokonce i někteří studenti tento systém znají a rádi by jej používali. Ve většině případů si však nevědí zcela rady s jeho instalací a nastavením.

Většinou jsou požadovány tyto vlastnosti:

- distribuce pro operační systém typu Windows,
- relativně snadné (i z hlediska potřebných znalostí) vytvoření zdrojového textu,
- bezproblémové prohlížení a tisk výsledných dvi souborů,
- export do PostScriptu (ps) a pdf, prohlížení a tisk ps a pdf souborů,

- specializovaný editor, ze kterého lze ovládat další programy a komponenty,
- nulová cena.

S ohledem na tyto požadavky a vzhledem k důvodům uvedeným výše se zaměřím na lokální instalaci MiKTeXu a na L^AT_EX. Zájemcům, kterým není problematika L^AT_EXu dostatečně známá, mohu doporučit publikaci [4] nebo také [5].

2. Co a kde stáhnout a jak instalovat^{*)**)}

2.1 Především vlastní MiKTeX

Získáme jej např. na <http://www.miktex.org>, sekce download. Aktuální verze (v době psaní návodu) je 2.4. V části **Installing the complete system** zvolíme **Download the setup wizard**. Stáhne se nám poměrně malý instalační program, který nám po spuštění nabídne dvě možnosti: instalaci přímo z Internetu nebo stažení balíčků do lokálního úložiště. Druhá možnost se jeví jako lepší, neboť můžeme jednou stažené balíčky např. vypálit na CD a použít na různých počítačích, nehledě na lepší možnost pozdější doinstalace některých komponent.

Proto zvolíme možnost **Download only**, jako **Package Set** vybereme **Large** nebo ještě lépe **Total** (uložení úplně všeho) a zvolíme příslušnou lokální složku, např. `c:\instalace\miktex\repo`. Pozor, celková velikost balíčků je ve variantě **Total** přibližně 400 MB.

Po dokončení downloadu znovu spustíme instalační program (**Setup Wizard**) a zvolíme **Install MiKTeX from a package local repository** ve variantě **Large**. Dále podle potřeby vybereme **Install MiKTeX for everyone** (tedy pro všechny uživatele) nebo **Install MiKTeX just for me** (pouze pro mě, tedy pro aktuálního uživatele). Jako **Path to local package repository** zadáme lokální složku se staženými balíčky (např. zmíněnou `c:\instalace\miktex\repo`). Jako **Path to installation folder** doporučuji `c:\miktex\texmf`, dále zaškrtneme volbu **Create a local TEXMF tree** a zde jako **Path to root folder** zadáme `c:\miktex\localtexmf`.

^{*)} V případě MiKTeXu doporučuji použít alespoň verzi 2.4, do které je už zahrnuta kompletní podpora češtiny.

^{**) Vyhne-li se instalací do složky **Program Files** a jeho podsložek, nemusíme mít (na většině počítačů) dokonce ani práva administrátora.}

Ještě je vhodné přidat cestu k binárním souborům $\text{\TeX}$ u do systémové proměnné PATH. V našem případě by se jednalo o `c:\miktex\texmf\miktex\bin`, postup manipulace se systémovými proměnnými závisí na verzi OS Windows (např. pro Windows 2000 použijeme pro nastavení postup Ovládací panely -> Systém -> Upřesnit -> Proměnné prostředí -> Systémové proměnné).

2.2 GhostScript a GhostView

Jedná se o interpret a prohlížeč PostScriptových souborů. Vše je ke stažení na adrese <http://www.cs.wisc.edu/~ghost>, zde vybereme GPL GhostScript (možno i AFPL GhostScript) a GSView nejnovější verze. Při instalaci necháme jako cílové defaultní složky (tj. pro GhostScript to bude např. `c:\gs`, pro GSView např. `c:\Program Files\Ghostgum`).

2.3 Specializovaný editor

Kromě WinEdt (což je ovšem shareware) se často používá jeden z těchto editorů: PSPad nebo TeXnicCenter.

PSPad je vynikající český programátorský editor, podporující mj. i $\text{\TeX}$. Stáhneme tzv. Instalátor (instalace může být do defaultní složky, např. `c:\Program Files\PSPad`) a rovněž je vhodné doinstalovat slovníky (alespoň český a anglický – ve formátu CAB, po rozbalení se tyto umístí do podsložky spell, např. tedy do `c:\Program Files\pspad\spell`).

TeXnicCenter je editor speciálně určený pro $\text{\TeX}$.

Oba zmíněné editory jsou k dispozici zcela zdarma. Jejich srovnání a detailní nastavení bude tématem samostatného článku.

2.4 Nepovinné doplňky

- `vlna32.exe`, program na nahrazování obyčejné mezery mezi jednotlivými předložkami a spojkami mezerou pevnou; tento soubor umístíme do složky s nainstalovaným $\text{\LaTeX}$ em, v našem případě tedy do `c:\miktex\texmf\miktex\bin`,
- `Latex2e.chm`, help k $\text{\LaTeX}$ u, uložíme do libovolné složky, např. do `c:\dokumentace`,
- `cstocs2`, konverze mezi různými kódováními češtiny,
- ImageMagick, konverze mezi různými formáty obrázků (např. jpg do eps), instaluje se samostatně.

2.5 Programy přehledně

Adresy pro získání všech programů uvedených v této kapitole jsou přehledně uvedeny v tabulce:

Název programu	Adresa
MiKTeX	http://www.miktex.org
GhostScript, GSView	http://www.cs.wisc.edu/~ghost
PSPad	http://www.pspad.com/cz/download.php
TeXnicCenter	http://www.texniccenter.org
Vlna32.exe	http://ftp.cstug.cz/pub/tex/local/cstug/olsak/vlna/
Latex2e.chm	http://ftp.cstug.cz/pub/tex/CTAN/info/LatexHelpBook/
cstocs2	http://ftp.cstug.cz/pub/tex/local/cstex/emtex/cstocs2.zip/
ImageMagick	http://www.imagemagick.org/

3. Nastavení MiKTeXu [1]

Máme nainstalován MiKTeXv anglické verzi. Nyní je zapotřebí provést počštění.

Pro vlastní instalaci csTeXu nejdříve spustíme MiKTeX Options, např. pomocí (Start\Programy\MiKTeX\MiKTeX Options). V kartě Packages zvolíme Language Support a označíme Czech/Slovak (a všechny jeho složky: csfonts, csLaTeX, csplain, cspsfonts a Czech/Slovak hyphenation patterns). Tyto volby potvrdíme stisknutím tlačítka Použít. Tím se nainstalují potřebné balíčky. Dále aktualizujeme databázi (např. volbou Refresh Now na záložce General v MiKTeX Options).

Chceme-li používat i český pdfplainTeX nebo pdfL^AT_EX, musíme ještě provést následující činnosti (neplatí pro případ, že k vytváření souborů .pdf používáme pouze program dvipdfm):

- ze složky c:\miktex\texmf\miktex\config zkopírujeme soubor miktex.ini do c:\miktex\localtexmf\miktex\config. Pokud uvedená složka ještě neexistuje, pak ji vytvoříme.
- z adresy <http://www.muweb.cz/www/miktex/navod/mikcnf.txt> si stáhneme soubor mikcnf.txt, obsah tohoto souboru si uložíme do schránky,

- otevřeme soubor `miktex.ini` a na jeho začátek vložíme obsah schránky,
- soubor `miktex.ini` uložíme a zavřeme,
- aktualizujeme databázi,
- vytvoříme formát `pdfcsplain` a `pdfcslatex`. Spustíme `MiKTeX Options`. V kartě `TeX Formats` klikneme na tlačítko `New` a do jednotlivých editačních polí nově zobrazeného okna doplníme texty uvedené v následující tabulce:

Název pole	Hodnota	
Format Name	<code>pdfcsplain</code>	<code>pdfcslatex</code>
Compiler	<code>pdfe-TeX (Extended Mode)</code>	<code>pdfe-TeX (Extended Mode)</code>
Input	<code>csplain.ini</code>	<code>cslatex.ini</code>
Output	<code>pdfcsplain.efmt</code>	<code>pdfcslatex.efmt</code>
Description	<code>pdfcsplain</code>	<code>pdfcslatex</code>

- v poli `Preloaded Format` nic neuvádíme a zaškrtneme `Exclude when updating all format files`. Poté potvrdíme `OK`, čímž se zavře okno `Format Definition`. V kartě `TeX Formats`, ve sloupci `Name` najdeme formát `pdfplain` nebo `pdfcslatex` a označíme jej myší. Klikneme na `Použít` a pak na `Build`, přesně v tomto pořadí. Poté se začne generovat příslušný formát, čímž vzniknou soubory `pdfcsplain.efmt` a `pdfcsplain.exe`, resp. `pdfcslatex.efmt` a `pdfcslatex.exe`.
- aktualizujeme databázi.

Poznámka: *Setkal jsem se s problémem týkajícím se YAPu: po jeho prvním spuštění obvyklým způsobem, např.*

`Start\Programy\MiKTeX\DVI viewer`,
se krátce objevila hláška týkající se generování fontů a poté se už neobjevilo vůbec nic. Problém jsem vyřešil dvojitým kliknutím na náhodně vybraný dvi soubor. Touto cestou – tedy přes asociaci – se YAP kupodivu spustil a od té doby už fungoval zcela normálně.

4. Příklad použití

V libovolném ASCII editoru (třeba i v Notepadu) zapíšeme tento zdrojový text:

XXXXXXXXXXXXXXXXXXXX

Následující příklady obsahují matematické výrazy včetně číslování a křížových odkazů na ně. Příklady jsou vybrány pro ilustraci možností `\LaTeX` u. Výraz `(\ref{derivace})` se týká derivace, výraz `(\ref{pravd})` kombinačních čísel a výraz `(\verb/\ref{matice}/)` uvedený na str. `\verb/\pageref{matice}/` řešení soustavy rovnic.

```
\begin{enumerate}
```

```
\item
```

Co je derivace $f'(x)$? Je to limita

```
\begin{equation}
```

```
\label{derivace}
```

```
\lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x},
```

```
\end{equation}
```

což je v geometrické interpretaci směrnice tečny v bodě x .

```
\item Pravděpodobnost výhry při losování  $k$  čísel z  $n$ 
```

možných lze vypočítat pomocí kombinačního čísla

$\binom{n}{k}$. Například při losování 6 čísel ze 49 je

```
\begin{equation}
```

```
\label{pravd}
```

```
\binom{49}{6} = \frac{49!}{43! \cdot 6!} =
```

```
\frac{49 \cdot 48 \cdot \dots \cdot 44}{6!} =
```

```
49 \cdot 4 \cdot 47 \cdot 46 \cdot 3 \cdot 11 \cdot 14 \cdot 10^6,
```

```
\end{equation}
```

tedy 1:14 milionům.

```
\item Při řešení soustavy rovnic zapíšeme koeficienty do
```

matice a provedeme naznačenou úpravu:

```
\begin{equation}
```

```
\label{matice}
```

```

\left[\begin{array}{cccc|c}
a_{11}&a_{12}&&\cdots &a_{1m}&b_1 \\\
a_{21}&a_{22}&&\cdots &a_{2m}&b_2 \\\
\vdots&\vdots&&\ddots &\vdots&\vdots \\\
a_{n1}&a_{n2}&&\cdots &a_{nm}&b_n \\\
\end{array}\right]
\Longrightarrow
\left[\begin{array}{cccc|c}
c_{11}&c_{12}&&\cdots &c_{1m}&d_1 \\\
0&c_{22}&&\cdots &c_{2m}&d_2 \\\
\vdots&\vdots&&\ddots &\vdots&\vdots \\\
0&0&&\cdots &c_{nm}&d_n \\\
\end{array}\right]
\end{equation}

```

⌞⌞⌞⌞⌞⌞⌞⌞⌞⌞⌞⌞⌞⌞⌞⌞

Soubor uložíme (do libovolné složky pod libovolným názvem, zachována musí být pouze přípona `.tex`), předpokládáme např. `C:\texty\prvni.tex`.

Spustíme emulátor příkazové řádky MS-DOSu (např. `Start -> Programy -> Příslušenství -> Příkazový řádek`) a zde přejdeme do uvedené složky a zavoláme překladač: `cslatex --tcx=cp1250cs prvni.tex`

Pokud překlad proběhl správně, tedy v dokumentu nejsou syntaktické chyby, objeví se hlášky typu `Output written on prvni.dvi` a `Transcript written on prvni.log`. Jinak se zobrazí hlášení s popisem chyby, pravděpodobným číslem řádku s chybou a systémový prompt (otazník). V takovém případě napíšeme `x`, potvrdíme (**Enter**) a vrátíme se ke zdrojovému textu, kde se pokusíme chybu najít a opravit. Poté překlad zopakujeme.

Vzhledem k tomu, že v příkladu jsou použity křížové odkazy (zajišťované příkazy `\label`, `\ref` a případně `\pageref`), je (po odstranění chyb) nutné překlad provést *dvakrát za sebou*. Při prvním překladu se totiž pouze vytvoří pomocný soubor s příponou `aux`, do něhož se uloží aktuální hodnoty příslušných čítačů a aktuální čísla stránek vytvořené na základě příkazů `label`, čili $\text{T}_{\text{E}}\text{X}$ si v podstatě poznamená informace o jednotlivých objekty, na něž se chceme odkazovat. Při druhém překladu se vytvořený (resp. změněný) soubor `aux` načte a údaje v něm uložené jsou využity pro potřeby příkazů `\ref` a `\pageref`.

Po úspěšně provedeném překladu si můžeme prohlédnout výsledek (`yap -1 prvni.dvi`):

⌘⌘⌘⌘⌘⌘⌘⌘⌘⌘

Následující příklady obsahují matematické výrazy včetně číslování a křížových odkazů na ně. Příklady jsou vybrány pro ilustraci možností L^AT_EXu. Výraz (1) se týká derivace, výraz (2) kombinačních čísel a výraz (3) uvedený na str. 8 řešení soustavy rovnic.

1. Co je derivace $f'(x)$? Je to limita

$$\lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}, \quad (1)$$

což je v geometrické interpretaci směrnice tečny v bodě x .

2. Pravděpodobnost výhry při losování k čísel z n možných lze vypočítat pomocí kombinačního čísla $\binom{n}{k}$. Například při losování 6 čísel ze 49 je

$$\binom{49}{6} = \frac{49!}{43! 6!} = \frac{49 \cdot 48 \cdot \dots \cdot 44}{6!} = 49 \cdot 4 \cdot 47 \cdot 46 \cdot 3 \cdot 11 \doteq 14 \cdot 10^6, \quad (2)$$

tedy 1 : 14 milionům.

3. Při řešení soustavy rovnic zapíšeme koeficienty do matice a provedeme naznačenou úpravu:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2m} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} & b_n \end{bmatrix} \Rightarrow \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} & d_1 \\ 0 & c_{22} & \cdots & c_{2m} & d_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & c_{nm} & d_n \end{bmatrix} \quad (3)$$

⌘⌘⌘⌘⌘⌘⌘⌘⌘⌘

Jako další akce je možné provést převod do PostScriptu, přičemž výsledek je možné prohlížet nebo tisknout např. programem GhostView, nebo do pdf.

5. Závěr

Výsledkem by mělo být funkční prostředí $\text{\TeX}$, zatím bez využití specializovaného editoru.

Tvar zdrojového textu $\text{\TeX}$ se na první pohled zdá komplikovaný a obtížně zapamatovatelný. Opak je pravdou; zápis je logický a zejména v případě matematické sazby je po základním zvládnutí syntaxe daleko efektivnější než v programech typu WYSIWYG.

Obzvlášť v případě dlouhých odborných dokumentů je velkým přínosem $\text{\TeX}$ stoprocentní funkčnost i v případě velkého množství komplikovaných křížových odkazů, správným způsobem jsou umísťovány netextové objekty. Samozřejmostí je snadné vytvoření odkazů na literaturu, obsahu, rejstříku, seznamu obrázků a tabulek.

$\text{\TeX}$ (resp. $\text{\LaTeX}$) ve škole je vhodný zejména pro použití v matematice a přírodovědných předmětech (fyzika, chemie . . .). Ale své uplatnění může najít rovněž v případě výuky cizích jazyků (podporuje různé cizojazyčné znaky včetně např. japonských) i jinde (podporuje mj. sazbu not a šachových partií).

Pokud tento článek alespoň trochu napomohl při základní orientaci v problematice $\text{\TeX}$ a také při instalaci a nastavení jedné z jeho často používaných distribucí, pak splnil svůj účel.

Literatura

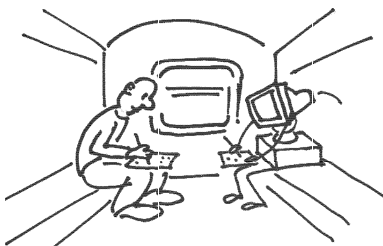
- [1] *Karolík, A.*: Instalace a počestění $\text{\MiKTeX}$
<http://www.muweb.cz/www/miktex/navod/newmiktex24.html>
- [2] $\text{\LaTeX}$ diskusní fórum VUT
<http://www.utko.feec.vutbr.cz/latex/index.php>
- [3] Československé sdružení uživatelů $\text{\TeX}$
<http://www.cstug.cz>
- [4] *Rybička, J.*: $\text{\LaTeX}$ pro začátečníky, 3. vydání, Konvoj, Brno, 2003.
- [5] *Kopka, H. – Daly, P. W.*: $\text{\LaTeX}$ – kompletní průvodce, Computer Press, Brno, 2004.
- [6] *Závodný, M.*: Typografický systém $\text{\TeX}$. MFI r. 9 (1999/2000), č. 4.

Sudoku

STANISLAV TRÁVNÍČEK

Přírodovědecká fakulta UP, Olomouc

(Dokončení)



Můžeme se o to pokusit „hrubou silou“, tj. tak, že u zbývajících neobsazených elementů vyzkoušíme všechny možnosti; použijeme na to rekursivní proceduru *Akce*. Bylo by však neúčelné zkoušet na každém místě všechny možnosti 1 až 9 (mohlo by to vést k příliš velkému počtu zkoušek – víme,

že jde o exponenciální složitost), když vlastně máme v $H[R, S, T]$ pro $T = 1$ až 9 pro každý element zaznamenáno, jaké možnosti jsou tu ještě přípustné, tj. budeme zkoušet jen přípustné možnosti. Pro potřeby rekurze si dosud neurčené elementy uspořádáme do posloupnosti úrovní (jejich počet je označen *Ur*) a pro záznam přípustných možností si zavedeme nové dvojrozměrné pole *Ind*; tento záznam přípustných možností se provede v proceduře *PripravAkci*. Doplňme také deklarace programu:

```
var
  Ur: Byte;
  PocRes: Integer;
  Ind: array [1..81,1..12] of Byte;
```

Vynulujeme pole *Ind*:

```
procedure NulujInd;
var U, V: Byte;
begin
  for U:= 1 to 81 do
    for V := 1 to 12 do Ind[U, V] := 0
end; {NulujInd}
```

a v proceduře *PripravAkci* si do pole $Ind[U, V]$ zaznamenáme potřebné údaje. Procedura prochází každým elementem. Když narazí na $H[R, S, 0] = 0$, zapíše index tohoto elementu do U (postupně od indexu 1 s rostoucím indexem). K tomuto indexu se pro $V = 1, 2, \dots, 9$ zapisují číslice přípustné v daném elementu (teoreticky by jich mohlo být 9, ale zpravidla je jich mnohem méně), pro $V = 10$ se do $Ind[U, V]$ zapíše jejich počet, pro $V = 11$ se zapíše číslo řádku a pro $V = 12$ číslo sloupce (tj. hodnoty proměnných R, S). Jestliže např. v řádku 5 a ve sloupci 7 budou přípustné jen hodnoty 2, 8, 9, bude mít $Ind[U, V]$ (po proběhnutí procedury *PripravAkci* pro $V = 1, 2, \dots, 12$ hodnoty: 2, 8, 9, 0, 0, 0, 0, 0, 0, 3, 5, 7.

```

procedure PripravAkci;
var R, S, I, J: Byte;
begin {PripravAkci}
  Ur := 0;
  for R := 1 to 9 do
    for S := 1 to 9 do
      if H[R, S, 0] = 0 then
        begin
          J := 0; Inc(Ur);
          for I := 1 to 9 do
            if H[R, S, I] = 0 then
              begin
                Inc(J); Ind[Ur, J] := I
              end;
            Ind[Ur, 10] := J; Ind[Ur, 11] := R; Ind[Ur, 12] := S
          end
        end
      end;
    {PripravAkci}
  end;
end;

```

Při samotné akci se do dosud nevyplněných (tj. vynulovaných) elementů postupně dosazují všechny možnosti od $Uro = 1$ až po $Uro = Ur$. Pokud po dosazení čísla úroveň Uro je Sudoku v pořádku (tj. není v něm spor s jeho základní vlastností) a Sudoku je zaplněno, bylo vytvořeno řešení, které se zobrazí na displeji, není-li Sudoku zaplněno, přejde se k úrovni $Uro + 1$ (tj. doplní se číslo v dalším elementu). Jestliže Sudoku v pořádku není, pak se přejde k dalšímu číslu téže úrovně, a pokud už žádné není, přejde se na předchozí úroveň $Uro - 1$, na níž se přejde k dalšímu číslu, atd.

```

procedure Akce(Uro: Byte);
var
  R, S, T, V: Byte;
begin {Akce}
  R := Ind[Uro, 11]; S := Ind[Uro, 12];
  for V := 1 to Ind[Uro, 10] do
    begin
      H[R, S, 0] := Ind[Uro, V];
      if (Uro < Ur) and VPoradku then Akce(Uro + 1);
      if VPoradku and Zaplneno then
        begin
          Inc(PocRes);
          WriteLn(PocRes, '. reseni s volbou alternativ:');
          VystupCtverce;
          WriteLn('Dale klavesou Enter..');
          ReadLn
        end
      end;
      H[R, S, 0] := 0
    end; {Akce}
  end;

```

Program, jehož první částí začíná tento článek, dokončíme pak např. tak, že místo jeho posledních tří řádků napíšeme:

```

begin
  WriteLn('Jednoznacny vysledek bez volby alternativ:');
  VystupCtverce; PocRes := 1
end else
begin
  PocRes := 0;
  NulujInd; PripravVolbu; Akce(1)
end;
if PocRes > 0 then WriteLn('Konec.')
else WriteLn('Reseni neexistuje.. ');
ReadLn
end. {program}

```

Samozřejmě nás napadne, jestli nebylo možné postupovat hned od začátku „hrubou silou“. Ano, bylo, ale pak by úloha nebyla tak zajímavá. A navíc: Když bychom v programu Sudoku v 1. části tohoto článku do konce těla příkazu `if` v proceduře `VlozNove` vložili proceduru `TiskCtverce` a `ReadLn` (pozastavení chodu programu), dostali bychom vždy informaci, kterou buňku bylo možno oproti předchozímu stavu obsadit, tedy tento program by nás vlastně *učil řešit* Sudoku.

Sudoku, kterým jsme v první části ukázkově začali, má řešení dle obr. 2.

5	7	9	2	8	3	1	4	6
3	8	1	4	6	9	2	5	7
4	6	2	5	7	1	8	3	9
1	3	8	6	2	7	4	9	5
7	9	4	3	5	8	6	2	1
6	2	5	9	1	4	7	8	3
9	4	6	1	3	2	5	7	8
2	5	7	8	9	6	3	1	4
8	1	3	7	4	5	9	6	2

Obr. 2

Vyskytují se i speciální případy Sudoku, např. kde k základním vlastnostem patří i to, aby se také ve dvou nejdelších diagonálách vyskytovaly všechny číslice. Pole působnosti elementů v diagonálách se tím tedy zvětšuje.

Vytvořte novou verzi programu Sudoku tak, že na vhodném místě vhodně upravíte zde uvedené procedury.

Existují i jednodušší schémata Sudoku s polem o rozměrech 4×4 elementy, i složitější s poli o rozměrech 16×16 elementů, ale jsou i některé další verze, na nichž si můžeme ověřit účinnost námi probraných algoritmů.

(Autorkou ilustrace je *Mgr. Jaroslava Čermáková* z Hlinska v Čechách.)