

INFORMATIKA

Zpracování naměřených teplot (Úlohy z MO – kategorie P, 16. část)

PAVEL TÖPFER

Matematicko-fyzikální fakulta UK, Praha

V dnešním díle seriálu o zajímavých úlohách z programátorských soutěží středoškoláků se budeme věnovat jedné na první pohled velmi snadné úloze z celostátního kola 53. ročníku matematické olympiády – kategorie P (školní rok 2003/2004). Úlohu do soutěže připravili autoři z Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislavě. Je zajímavá tím, že její primitivní řešení jistě rychle napadne i úplného začátečníka. Pokud ale začneme usilovat o zlepšení časové složitosti algoritmu, budeme při řešení nuceni použít některé zajímavé a užitečné obraty, které si právě v tomto článku ukážeme. Začneme jako obvykle zadáním úlohy v původním znění:

Meteorologická stanice měří každou minutu teplotu vzduchu. Meteorologové by potřebovali program, který by jim v každém okamžiku sděloval, jaká nejnížší teplota byla naměřena během posledních K minut. Vaším úkolem bude napsat tento program.



Na vstupu je dáno číslo K , následuje posloupnost naměřených teplot ukončená hodnotou -1000. Váš program musí po přečtení každé teploty vypsat nejnížší z posledních K načtených teplot (resp. nejnížší ze všech načtených teplot, jestliže jich dosud bylo méně než K .)

Příklad:

Vstup	K=3	Výstup
	teploty:	
	9.0	9.0
	4.7	4.7
	5.3	4.7
	2.1	2.1
	9.0	2.1
	9.8	2.1
	17.0	9.0
	9.5	9.5
	-1000	

Počet všech naměřených hodnot označíme N . Pro každou z nich nás tedy bude zajímat minimum z posledních K hodnot, příp. zpočátku minimum ze všech již naměřených hodnot, pokud jich dosud bylo méně než K .

1. Primitivní řešení

Jednoduché řešení využívá jednorozměrné pole velikosti K , do kterého si postupně ukládáme všechny načtené hodnoty. Po naplnění pole prvními K čísly ze vstupu začneme obsah pole cyklicky přepisovat, takže v poli bude vždy uloženo posledních K načtených hodnot. Pokaždé, když ze vstupu přečteme jednu naměřenou teplotu a uložíme ji do pole, určíme minimum z čísel uložených v poli a to vypíšeme na výstup. Paměťová složitost popsaného řešení je zjevně $O(K)$. Časová složitost je $O(N \cdot K)$, neboť N -krát čteme číslo ze vstupu a každé přečtené číslo zpracujeme v čase $O(K)$ – do pole ho vždy uložíme v konstantním čase a následně hledáme sekvenčním průchodem minimum z K uložených hodnot.

Programová realizace tohoto postupu řešení je zcela přímočará a nebudeme ji zde proto ani uvádět.

2. Využití haldy

Nyní se pokusíme nalézt nějaké rychlejší řešení úlohy. Jestliže místo obyčejného neuspořádaného pole použijeme k uložení posledních K naměřených teplot datovou strukturu halda, dokážeme zpracovat každé přečtené číslo v čase $O(\log K)$, takže celou úlohu vyřešíme s časovou složitostí $O(N \cdot \log K)$. Připomeňme si jenom krátce, že halda je stromová struktura, která umožňuje přidávat a odebírat prvky v logaritmickém čase a určit minimum z uložených hodnot v čase konstantním. Implementuje se zpravidla v jednorozměrném poli velikosti K . Podrobnější popis haldy a příslušných operací můžete nalézt například v učebnici [1] nebo v článku [2]. V našem seriálu článků o úlohách z MO kategorie P jsme haldu použili podobným způsobem již v předchozím 15. dílu v [3].

Při použití haldy v řešení naší úlohy ovšem narazíme na jeden vážný problém. Dokážeme sice přidat právě přečtenou hodnotu do haldy v čase $O(\log K)$ a následně určit minimum z hodnot uložených v haldě a vypsat ho na výstup v čase konstantním, neumíme ale odstranit z haldy nejstarší hodnotu. Nemáme totiž zatím nijak evidováno, jak je která z hodnot uložených v haldě stará. Pro vyřešení tohoto problému bychom si mohli s každou přečtenou hodnotou ukládat do haldy i její pořadové číslo. Potom bychom ale při odstraňování nejstarší hodnoty museli celou haldu projít v čase $O(K)$, abychom tuto nejstarší hodnotu našli. Tím by se složitost řešení celé úlohy znehodnotila na $O(N \cdot K)$.

Evidenci „stáří“ jednotlivých údajů uložených v haldě proto vyřešíme šikovněji. Vedle haldy si budeme vytvářet ještě frontu, do které budeme řadit odkazy na jednotlivé prvky haldy v tom pořadí, jak jsme hodnoty četli ze vstupu. Získáme tak v každém okamžiku přímý přístup k tomu prvku haldy, který je nejstarší. Časová složitost jeho odstranění z haldy pak bude skutečně rovna nejvýše výšce haldy, tzn. $O(\log K)$. Paměťové nároky na uložení haldy i fronty jsou $O(K)$.

Poznamenejme ještě, že z technických důvodů je třeba, aby odkazy mezi prvky haldy a záznamy ve frontě byly vzájemné – to souvisí se způsobem provádění haldových operací. Při přidávání prvku do haldy a odebírání prvku z haldy se totiž musí některé záznamy uložené v haldě přemísťovat a jejich novou polohu je třeba aktuálně opravovat také ve frontě. Jeden možný způsob programové realizace celého řešení předvádí následující ukázka.

```
program Teploty1;  
{řešení úlohy s využitím haldy}
```

```

const   MaxK   = 1000; {maximální přípustná hodnota K}
        Konec  = -1000; {ukončovací hodnota na vstupu}

type Fronta = record
    A: array[1..MaxK] of integer;
        {index do haldy}
    Zac, Kon: integer;
        {index začátku a konce fronty}
end;

Halda = record
    A: array[1..MaxK] of
        record T: real;      {teplota}
          Fr: integer; {index
                        do fronty}
        end;
    Poc: integer; {počet prvků v haldě}
end;

var K: word; {délka sledovaného úseku teplot}
    T: real; {teplota}
    I: word; {pořadové číslo aktuální teploty}
    F: Fronta;
    H: Halda;

procedure Pridej(T: real);
{přidá do haldy H teplotu T, při přemísťování prvků
v haldě H průběžně opravuje také údaje o jejich poloze
uvedené ve frontě F, na konec fronty F zařadí odkaz
na novou teplotu v haldě H}

var Kde: integer; {index do haldy H, kde je nový prvek}
begin
    inc(H.Poc); {zvětšení haldy}
    Kde:=H.Poc;
    while (Kde > 1) and (T < H.A[Kde div 2].T) do
        begin
            {bublání nové teploty haldou nahoru}

```

```

    H.A[Kde]:=H.A[Kde div 2];
                                {posun záznamu v haldě z otce do syna}
    F.A[H.A[Kde].Fr]:=Kde;
                                {novou polohu opravit také ve frontě!}
    Kde:=Kde div 2
    end;
    H.A[Kde].T:=T;      {vložení teploty na své místo do haldy}
    inc(F.Kon);          {nový záznam ve frontě}
    if F.Kon > MaxK then F.Kon:=1;
    F.A[F.Kon]:=Kde;    {vzájemné odkazy mezi haldou a frontou}
    H.A[Kde].Fr:=F.Kon
end; {procedure Pridej}

```

```

procedure Zrus_nejstarsi;
{pomocí fronty F zruší nejstarší prvek z neprázdné haldy H}

var Kde: integer; {index do haldy H, kde je rušený prvek}
    Kam: integer; {index, kam se přemísťuje náhradní prvek}
begin
    Kde:=F.A[F.Zac];
    inc(F.Zac);      {zrušení nejstaršího záznamu ve frontě F}
    if F.Zac > MaxK then F.Zac:=1;
    {zrušení prvku s indexem Kde v haldě H
    -- nahradíme ho posledním prvkem haldy H.A[H.Poc]
    a ten probublá podle potřeby nahoru nebo dolů:}
    Kam:=Kde div 2; {bublání prvku haldou nahoru}
    while (Kde > 1) and (H.A[H.Poc].T < H.A[Kam].T) do
        begin
            H.A[Kde]:=H.A[Kam];
                                {posun záznamu v haldě z otce do syna}
            F.A[H.A[Kde].Fr]:=Kde;
                                {novou polohu opravit také ve frontě!}
            Kde:=Kam;
            Kam:=Kde div 2
        end;
    Kam:=2*Kde;          {bublání prvku haldou dolů}
    if Kam < H.Poc-1 then {prvek Kde má i pravého syna}

```

```

    if H.A[Kam+1].T < H.A[Kam].T then
        {pravý syn je menší než levý}
        Kam:=Kam+1;
    while (Kam < H.Poc) and (H.A[H.Poc].T > H.A[Kam].T) do
        begin
            H.A[Kde]:=H.A[Kam];
                {posun záznamu v haldě ze syna do otce}
            F.A[H.A[Kde].Fr]:=Kde;
                {novou polohu opravit také ve frontě!}
            Kde:=Kam;
            Kam:=2*Kde;
            if Kam < H.Poc-1 then
                if H.A[Kam+1].T < H.A[Kam].T then
                    Kam:=Kam+1;
                end;
            H.A[Kde]:=H.A[H.Poc]; {přemístit poslední prvek haldy
                                    na své nově nalezené místo}
            F.A[H.A[Kde].Fr]:=Kde; {novou polohu opravit ve frontě}
            dec(H.Poc);           {zmenšení haldy}
        end; {procedure Zrus_nejstarsi}

```

```

function Minimum: real;
{vrací hodnotu minimální teploty z haldy H}
begin
    Minimum:=H.A[1].T
end; {function Minimum}

```

```

begin {program}
H.Poc:=0;      {prázdná halda}
F.Zac:=1;      {prázdná fronta}
F.Kon:=0;

```

```

write('Délka sledovaného úseku K: ');
read(K);
writeln;
writeln('Posloupnost naměřených teplot ukončená -1000:');

```

```

for I:=1 to K do {zpracování prvních K čísel ze vstupu}

```

```

begin
read(T);
if T = Konec then
    begin writeln; halt end;
    Pridej(T);    {do haldy H i do fronty F}
    writeln(Minimum:8:2) {hodnota z kořene haldy H}
end;

read(T);          {zpracování dalších čísel ze vstupu}
while T <> Konec   do
    begin
    Pridej(T);      {do haldy H i do fronty F}
    Zrus_nejstarsi; {z haldy H i z fronty F}
    writeln(Minimum:8:2); {hodnota z kořene haldy H}
    read(T)
    end;

writeln
end.

```

To ale ještě není všechno. Zatímco autorské řešení úlohy, jehož český překlad byl publikován v ročence [4], končí u výše popsaného postupu s časovou složitostí $O(N \log K)$, my si ukážeme ještě lepší způsob řešení. Použijeme v něm stejný trik, jaký jsme uplatnili v předchozím článku [3].

(Pokračování)

Literatura

- [1] *Töpfer, P.*: Algoritmy a programovací techniky, Prometheus, Praha, 1995.
- [2] *Kára, J. – Král, D. – Mareš, M.*: Recepty z programátorské kuchačky Korespondenčního semináře z programování – 1. část. RMF, r. 80 (2005) č. 1.
- [3] *Töpfer, P.*: Optimalizace cesty (Úlohy z MO – kategorie P, 15. část). MFI r.16 (2006/07), č. 3.
- [4] *Horák, K. a kol.*: 53. ročník matematické olympiády na středních školách, JČMF, Praha, 2006.

Editory pro T_EX

JOSEF BARÁK

Obchodní akademie a VOŠS, Brno

Článek volně navazuje na [6] a zabývá se porovnáním, instalací a nastavením dvou freewareových specializovaných editorů PSPad a TeXnicCenter pro spolupráci s MikT_EXem. Oba editory jsou vhodné nejen pro pořízení zdrojového textu v T_EXu, ale mohou tvořit kompletní integrované prostředí (IDE), neboť při vhodné konfiguraci z nich lze řídit překlad, prohlížení, tisk i volání doplňkových programů.

V závěru článku je uveden postup zápisu a celkového zpracování matematického výrazu v L^AT_EXu s využitím obou editorů.

1. Úvod

Pro pořízení zdrojového textu v T_EXu lze samozřejmě využít jakýkoliv editor, ve windowsovém prostředí např. NotePad. Většinou se však užívají editory specializované. Jejich výhodou je nejen barevné zvýraznění syntaxe, kontrola párových závorek, doplňování klíčových slov, kontextová nápověda apod., ale i možnost řízení dalších činností souvisejících se zpracováním dokumentu.

Editorů úzce spolupracujících s T_EXem je mnoho. Zaměříme-li se na OS Windows a na požadavek, aby se jednalo o freeware, zjistíme, že se v českém prostředí často používají dva takové editory: PSPad a TeXnicCenter.

1.1 PSPad

PSPad je vynikající český programátorský editor, podporující mj. i T_EX, nejedná se tedy o editor jednoúčelový. Jeho univerzálnost může být v některých případech výhodou v tom smyslu, že uživatel si nemusí zvykat na různá prostředí – ve stejném editoru lze např. také programovat v C/C++ nebo PHP, vytvářet HTML nebo CSS a mnoho dalších činností.

Editor je dostupný na adrese <http://www.pspad.com>.

1.2 TeXnicCenter

TeXnicCenter je jednoúčelový editor speciálně určený pro T_EX (se zaměřením na L^AT_EX). Rozeznává mnoho L^AT_EXových konstrukcí a barevně

je zvýrazňuje, navíc obsahuje klikací nástroje pro vkládání běžnějších příkazů do zdrojového textu.

Editor je dostupný na adrese <http://www.toolscenter.org>.

1.3 Srovnání editorů

Oba editory mají společné vlastnosti. Podporují barevné zvýraznění syntaxe, doplňování klíčových slov, kontextovou nápovědu a oba mohou být nastaveny jako $\text{\TeX}$ ové IDE.

Následující hlavní odlišnosti a relativní výhody (+) a nevýhody (–). Je-li uvedeno (+/–), jedná se o výhodu resp. nevýhodu v závislosti na konkrétním uživateli.

PSPad

- + kompletně v češtině, včetně nápovědy (existuje ale i lokalizace do jiných jazyků),
- + neustále se vyvíjí, jsou doplňovány nové funkce a vlastnosti,
- + /– je univerzální, lze využívat i pro zápis jiného než $\text{\TeX}$ ového kódu,
- chybí klikací nástroje pro vkládání $\text{\TeX}$ ových příkazů.

TeXnicCenter

- + obsahuje klikací nástroje na vkládání $\text{\TeX}$ ových příkazů, např. kliknutím na symbol se zlomkovou čarou se do textu vloží příkaz $\backslash\text{frac}\{\}\{ \}$,
- + autodetekce některých $\text{\TeX}$ ových distribucí (např. Mik $\text{\TeX}$ u) a doplňkových programů,
- + /– jednoúčelový, určený pro tvorbu dokumentů v $\text{\TeX}$ u (se zaměřením na $\text{\LaTeX}$),
- není lokalizován do češtiny,
- slabší vývoj nových verzí; verze jsou značeny jako Beta a např. mezi po sobě následujícími verzemi 1 Beta 6.31 a 1 Beta 7.01 byla cca roční prodleva.

1.4 Konfigurace editorů

Následuje popis nastavení obou editorů pro spolupráci s windowsovou $\text{\TeX}$ ovou distribucí Mik $\text{\TeX}$. V obecné rovině o $\text{\TeX}$ u poměrně podrobně pojednává [5], windowsovými distribucemi se zaměřením na Mik $\text{\TeX}$, instalací této distribuce a doplňkových programů se zabývá [6].

V dalším textu předpokládáme instalaci Mik $\text{\TeX}$ u v `c:\miktex`. Help k $\text{\LaTeX}$ u je umístěn v `c:\dokumentace`. GhostView je ve složce

c:\Program Files\Ghostgum\gsview. Soubor vlna32.exe je ve složce c:\miktex\texmf\miktex\bin. Pokud tato umístění neplatí, je zapotřebí patřičně upravit cesty.

2. Instalace a konfigurace PSPadu

2.1 Instalace

Česká varianta je ke stažení na <http://www.pspad.com/cz/download.php>, zde vybereme první možnost Instalátor. Instalace může být provedena do defaultní složky (např. c:\Program Files\PSPad). Vhodná je rovněž doinstalace slovníků (alespoň českého a anglického – ve formátu CAB, po rozbalení se umístí do podsložky `spell`, např. tedy do c:\Program Files\pspad\spell)

2.2 Konfigurace

Spustíme PSPad a zvolíme Nastavení\Nastavení zvýrazňovačů..., zde zaškrtneme TeX.

V záložce Upřesnění nastavíme cestu k souboru s nápovědou:
Soubor s nápovědou: c:\dokumentace\latex.chm

Dále v záložce Kompilátor vyplníme následující údaje:

```
Kompilátor = cslatex.exe
Parametry = --src-specials --tcx=cp1250cs %Name%.tex
LOG soubor = %Name%.log
Log parser = 1.%L ; LaTeX
```

Zaškrtneme volbu: Před kompilací uložit všechny soubory.

Poznámka: Parametr `--src-specials` slouží k tzv. zpětnému prohledávání (klikneme-li na řádek zobrazený v prohlížeči dvi souborů YAP, přesuneme se tím na odpovídající řádek zdrojového souboru), parametr `--tcx=cp1250cs` je informací o vstupním kódování češtiny ve zdrojového textu (v tomto případě se jedná o Windows CP-1250).

Na záložce Externí programy postupně přidáme

Prohlížení dvi souborů

```
Program: DVI View
Jméno: yap -1 "%Dir%%Name%.dvi"
```

Převod dvi do PostScriptu

Program: DVIPS

Jméno: dvips.exe "%Dir%%Name%.dvi"

Prohlížení a tisk PostScriptových souborů

Program: GSView

Jméno:

"c:\Program Files\Ghostgum\gsview\gsview32.exe" ->
"%Dir%%Name%.ps"

Pozn: šipka -> značí pokračování řádku

Převod dvi do pdf

Program: dvi2pdf

Jméno: dvipdfm.exe "%Dir%%Name%.dvi"

Nahrazení mezer za jednopísmennými předločkami pevnými mezerami

Program: Vlna32

Jméno: "c:\miktex\texmf\miktex\bin\vlana32.exe" "%File%"

2.3 Nastavení PSPadu jako chybového editoru

Pokud $\text{\TeX}$ zjistí při překladu zdrojového souboru chybu, překlad se přeruší a v okně se objeví prompt ? (tedy otazník). Na tuto výzvu můžeme reagovat různým způsobem. Pokud budeme chtít, aby se nám zobrazil text zdrojového souboru tam, kde se podle $\text{\TeX}$ u pravděpodobně chyba vyskytuje, odpovíme stiskem klávesy e. Použijeme-li následující nastavení, spustí se v takovém případě PSPad, a v něm příslušný zdrojový soubor. V tomto souboru se kurzor přemístí na řádek (přesněji odstavec), kde pravděpodobně došlo k chybě, a tento řádek se zvýrazní.

Postup:

Pokud ještě neexistuje soubor

c:\miktex\localtexmf\miktex\config\miktex.ini, pak jej vytvoříme zkopírováním tohoto konfiguračního souboru z

c:\miktex\texmf\miktex\config\miktex.ini.

Pokud už uvedený soubor miktex.ini existuje, pak nic nekopírujeme!

Nyní otevřeme soubor ... \localtexmf\... \miktex.ini a upravíme v něm v sekci [TeX] řádek

```
Editor=notepad "%f" do tvaru
Editor="{cesta}\PSPad.exe" -%l "%f", tedy např.
Editor="c:\Program Files\PSPad\-\PSPad.exe" -%l "%f"
(prvním parametrem je malé L, nikoliv číslice 1).
```

2.4 Zpětné hledání (Inverse Search)

Princip zpětného hledání využijeme tehdy, když se budeme chtít ve zdrojovém souboru přesunout na místo, na které se právě díváme v souboru dvi (prohlížečem YAP). V YAPu stačí pouze myší dvakrát kliknout na místo, kam se chceme ve zdrojovém souboru přesunout. Poté se nám spustí PSPad, v něm se otevře daný zdrojový soubor s kurzorem na řádce „blízké“ místu, kde jsme klikali. Tento princip umožňuje velmi podstatné zrychlení orientace ve zdrojovém kódu.

Postup:

Spustíme YAP (např. `Start\Programy\MiKTeX\DVI viewer`). Zvolíme nabídku `View\Options...` a zde záložku `Iverse Search`. Dále do pole `Command Line` zapíšeme volání PSPadu (včetně uvozovek):

```
"{cesta}\PSPad.exe" -%l "%f"
tedy např. "c:\Program Files\PSPad\PSPad.exe" -%l "%f"
```

2.5 Vytvoření souboru `Default.tex`

Pro co nejpohodlnější práci s editorem je vhodné, aby se již při založení nového souboru vygenerovala základní struktura $\text{\TeX}$ ového (a v našem případě $\text{\LaTeX}$ ového) dokumentu. K tomuto účelu si vytvoříme soubor `Default.tex` (pozor na velká a malá písmena) s tímto obsahem:

```
⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈
```

```
\documentclass[a4paper]{article}
\usepackage{czech}
\begin{document}
```

```
\end{document}
```

```
⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈⋈
```

Soubor uložíme do složky `c:\Program Files\PSPad\Template`.

3. Instalace a konfigurace TeXnicCenter

3.1 Instalace

Ke stažení na <http://sourceforge.net/projects/texniccenter>. Instalace může být provedena do defaultní složky (např. C:\Program Files\TeXnicCenter). Vhodná je doinstalace českého slovníku pro kontrolu pravopisu (využívá se slovník z OpenOffice). Slovník je ke stažení např. na

http://ftp.linux.cz/pub/localization/OpenOffice.org/devel/Czech/spell_checking/cs_CZ.zip,

nebo se k němu můžeme dostat prostřednictvím odkazu *Download Dictionaries* v menu TeXnicCenter (Tools\Options, záložka Spelling). Obsah zip archívu se slovníkem je nutné rozbalit do podsložky Language.

3.2 Konfigurace

TeXnicCenter má v případě spolupráce s MikTeXem velkou výhodu: dokáže automaticky najít existující instalaci MikTeXu a na základě ní si nastavit základní konfiguraci. Rovněž dokáže automaticky detekovat některé doplňkové programy (GhostView, Acrobat Reader apod.). Z toho důvodu je velmi vhodné instalovat TeXnicCenter jako poslední, až po instalaci všech ostatních programů.

Bohužel, vzhledem k tomu, že autoři editoru neuvažovali o existenci české varianty MikTeXu, se jistým ručním úpravám konfigurace nevyhneme.

Konfigurace editoru TeXnicCenter je založena na tzv. profilech. Defaultně se takové profily vytvoří tři: LaTeX⇒DVI, LaTeX⇒PS a LaTeX⇒PDF, což pro základní práci postačuje.

Konfigurace profilů se provádí prostřednictvím nabídky Build\Define Output Profiles nebo zkratky <alt><F7>. Pro každý z nich se nastavují jednotlivá pole v záložkách (La)TeX, Postprocessor a Viewer.

Při dodržení správného postupu instalace MikTeXu, doprovodných programů i samotného TeXnicCenter by se měly příslušné hodnoty těchto polí vyplnit automaticky (např. v poli Path to (La)TeX Compiler bude uvedeno něco jako C:\miktex\texmf\MiKTeX\bin\latex.exe).

Vzhledem k tomu, že používáme českou variantu T_EXu, musíme v konfiguraci provést ruční úpravu. Všechna místa (ve všech profilech), kde je uvedeno volání programu **latex**, musíme upravit na **cslatex**, podobně je

nutné nahradit `pdflatex` `pdfcslatexem`. Takže v profilech $\text{LaTeX} \Rightarrow \text{DVI}$ a $\text{LaTeX} \Rightarrow \text{PS}$ bude nyní uvedeno `C:\miktex\texmf\MiKTeX\bin\cslatex.exe`, podobně v profilu $\text{LaTeX} \Rightarrow \text{PDF}$ bude `C:\miktex\texmf\MiKTeX\bin\pdfcslatex.exe`.

Zásadně musíme změnit hodnotu pole **Command line arguments to pass to the compiler**. Aby byly ve výsledném dokumentu správně zobrazeny všechny české znaky a aby fungovala vlastnost **Inverse Search**, je ve všech profilech nutné přidat parametry `--tcx=cp1250cs` a `--src-specials`. Tedy příklad správné podoby hodnoty tohoto pole (profil $\text{LaTeX} \Rightarrow \text{DVI}$):

```
--src-specials --tcx=cp1250cs -interaction=nonstopmode ->
-max-print-line=120 "%Wm",
```

Poznámka: Pokud nechceme používat **BibTeX** a **MakeIndex**, je vhodné zaškrtnout příslušná políčka (**Do not use ...**). Zrychlí se tím překlad a zprehlední log soubor.

3.3 Zpětné hledání (Inverse Search)

Spustíme YAP (např. `Start\Programy\MiKTeX\DVI viewer`). Zvolíme nabídku `View\Options...` a zde záložku **Iverse Search**. Dále do pole **Command Line** zapíšeme volání `TeXnicCenter`:

```
{cesta}\TEXCNTR.EXE /ddecmd "[goto('%f', '%l')]",
tedy např. C:\Program Files\TeXnicCenter\TEXCNTR.EXE /ddecmd ->
"[goto('%f', '%l')]"
```

3.4 Vytvoření šablony LaTeX ového dokumentu

Pro automatické vytvoření základní struktury LaTeX ového dokumentu je nutné vytvořit tzv. šablonu, podle níž se zakládají nové dokumenty. Šablony mohou být projektové nebo dokumentové, přičemž v rámci článku se budeme zabývat především šablonami dokumentovými. V obou případech se jedná o soubor s příponou `.tex` umístěný do správné složky (standardně se předpokládá `{cesta}\Templates\Projects` pro projektové nebo `{cesta}\Templates\Documents` pro dokumentové šablony). Pokud chceme mít vytvořeny skupiny šablon, můžeme v těchto složkách vytvořit ještě podsložky (tedy např. `{cesta}\Templates\Documents\LaTeX` pro LaTeX ové dokumentové šablony).

Postup tvorby dokumentové šablony:

Vytvoříme soubor se základní strukturou LaTeX ového dokumentu:

XXXXXXXXXXXX

```
\documentclass[a4paper]{article}
\usepackage{czech}
\begin{document}

\end{document}
```

XXXXXXXXXXXX

Tento soubor uložíme např. jako `article.tex` (název může být libovolný, povinná je pouze přípona), a to do složky `{cesta}\Templates\Documents\LaTeX`.

Pak pomocí nabídky `Tools\Options`, záložka `Directories` přidáme uvedenou složku tlačítkem `Add` do seznamu `Document template directories`.

4. Konkrétní příklad použití editorů

4.1 Matematický výraz

Zdrojový text následujícího matematického výrazu byl v rozšířené podobě použit již v [6], nyní si na jeho první části ukážeme možnosti popisovaných editorů, nakonfigurovaných dle výše uvedených postupů. Samozřejmě je možné vyzkoušet si příklad v jeho úplné podobě.

XXXXXXXXXXXX

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 .

XXXXXXXXXXXX

4.2 PSPad

Založíme nový soubor: `Soubor\Nový` (nebo `<ctrl><N>`), typ souboru `TeX`. Nyní se vytvoří soubor s vygenerovanou základní strukturou `LATEX`ového dokumentu. Text příkladu uvedeného v kap. 4.1 zapišeme mezi části

`\begin{document}` a `\end{document}`. Při psaní můžeme využívat seznamu klíčových slov `TeXu` dostupných prostřednictvím nabídky **Nástroje\Seznam šablon** (nebo lépe zkratky `<ctrl><space>`), případně automatické dokončování klíčových slov (**Nástroje\Automatické dokončování**, resp. `<ctrl><J>`).

Překlad provedeme volbou **Soubor\Kompilovat**, resp. zkratkou `<ctrl><F9>`. Pokud jsme udělali syntaktickou chybu, objeví se systémový prompt (?). V tomto případě postačí zadat **e**, potvrdit klávesou **Enter** a kurzor se automaticky umístí na řádek s chybou.

Proběhl-li překlad bez chyby (v dolním logovém okně se objeví hláška typu **Output written ...**), můžeme si v prohlížeči YAP prohlédnout resp. vytisknout výsledek: **Soubor\Otevřít v programu\DVIPs View** (nebo klávesa **F9**). Protože máme nastaveno zpětné hledání (Inverse Search), postačí kdekoliv ve výsledku dvojité kliknutí levým tlačítkem myši a zavolá se PSPad s kurzorem nastaveným na odpovídajícím řádku zdrojového textu.

Můžeme též provést další činnosti, především převod do PostScriptu (**Soubor\Otevřít v programu\DVIPS**, resp. `<alt><F9>`), prohlížení a tisk PostScriptového výsledku (**Soubor\Otevřít v programu\GsView 32**, resp. `<shift><F9>`) nebo třeba export do pdf (**Soubor\Otevřít v programu\dvi2pdf**).

4.3 TeXnicCenter

Založíme nový soubor: **File\New from Template** (nebo `<ctrl><N>`). Nyní vybereme záložku příslušné skupiny (v našem případě **LaTeX**) a šablonu (v našem případě **article**). Nyní by se měl objevit dokument s vytvořenou základní strukturou **LaTeX**ového dokumentu.

Text příkladu uvedeného v kap. 4.1 zapíšeme mezi části `\begin{document}` a `\end{document}`. Při psaní můžeme využívat plovoucích panelů s nástroji.

Překlad provedeme volbou **Build\Current File\Build**, resp. zkratkou `<ctrl><F7>`. O jeho průběhu a případných chybách jsme informováni ve spodním logovém okně.

Proběhl-li překlad bez chyby, můžeme si v prohlížeči YAP prohlédnout resp. vytisknout výsledek: **Build\View Output** (nebo klávesa **F5**). Protože máme nastaveno zpětné hledání (Inverse Search), postačí kdekoliv ve výsledku dvojité kliknutí levým tlačítkem myši a zavolá se TeXnicCenter s kurzorem nastaveným na odpovídajícím řádku zdrojového textu.

Můžeme též provést převod do PostScriptu, jeho následné prohlížení a tisk (překlad a prohlížení se zvoleným profilem $\text{LaTeX} \Rightarrow \text{PS}$) nebo totéž pro pdf (překlad a prohlížení se zvoleným profilem $\text{LaTeX} \Rightarrow \text{PDF}$).

4.4 Výsledná podoba příkladu

❖❖❖❖❖❖❖❖❖❖

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 .

❖❖❖❖❖❖❖❖❖❖

5. Závěr

Oba popisované editory se jeví jako v podstatě rovnocenné a jako velmi vhodné pro spolupráci s $\text{T}_{\text{E}}\text{X}$ em, přičemž mohou plnit i funkci IDE; to vše navíc zcela zdarma. Pro využití ve školách se zdá být vhodnější PSPad, a to jednak pro svůj český původ, ale též z důvodu jeho univerzální (nejen $\text{T}_{\text{E}}\text{X}$ ové) využitelnosti – programování, tvorba www stránek a podobně.

Konečné rozhodnutí však vždy vychází z konkrétních požadavků a potřeb.

L i t e r a t u r a

- [1] $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ diskusní fórum VUT <http://www.utko.feec.vutbr.cz/latex/index.php>
- [1] Československé sdružení uživatelů $\text{T}_{\text{E}}\text{X}$ u <http://www.cstug.cz>
- [3] *Rybička, J.*: $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ pro začátečníky, 3. vydání, Konvoj, Brno, 2003.
- [4] *Kopka, H. – Daly, P. W.*: $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ – kompletní průvodce, Computer Press, Brno, 2004.
- [5] *Závodný, M.*: Typografický systém $\text{T}_{\text{E}}\text{X}$. MFI r. 9 (1999/2000), č. 4.
- [6] *Barák, J.*: Pracujeme v $\text{T}_{\text{E}}\text{X}$ u. MFI r. 16 (2006/2007), č. 5.