Zpravodaj Československého sdružení uživatelů TeXu

Stanislav Brabec

Lunisolární výpočty v TeXu

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 5 (1995), No. 1-4, 16–19

Persistent URL: http://dml.cz/dmlcz/149738

Terms of use:

© Československé sdružení uživatelů TeXu, 1995

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

pozici je v CS-fontech alternativní hyphenchar. Použijete-li ve vstupním souboru ‚—‘ bez definice, dostanete obyčejný spojovník (nedělitelný). Po použití výše zmíněné definice, dosáhnete kýženého efektu opakování spojovníku na začátku řádky.

Přemapování znaků v tcp-tabulce můžete nejlépe provést pomocí programu `maketcp`. Pokud jste na tom jako já a tento program nemáte momentálně po ruce, můžete provést změny přímo v souboru `kamenic.tcp` nebo `pclatin2.tcp` (fuj, to je ale ošklivý způsob). Obsah bytu na offsetu c8h (počítáno od nuly) změňte na hodnotu c4h; obsah bytu na offsetu 1c8h změňte na hodnotu 9ch. Nezapomeňte po provedení změn v tcp-tabulce znovu vygenerovat formát, aby začaly změny účinkovat. (V CS-TEXu slouží ke generování formátu dávka `initex` uložená v adresáři `BATS`.)

Lunisolární výpočty v TEXu

Bc. Stanislav Brabec

Tento článek patří svým zaměřením spíš k TEXovým kuriozitám. Chce ukázat, co vše lze v TEXu dělat a jak neuvěřitelná makra lze psát. K napsání mě inspirovala kniha [1], zvláště pak její kapitola o „bílých trpaslících“ a „introvertech“ mezi programy, kde byl jako příklad popsán Gaussův algoritmus pro výpočet data velikonoc. A protože jsem též přečetl známý článek [2] a z valné části i knihu [3] s krásnou ukázkou Erastothenova síta, došel jsem k závěru, že nejkrásnější ukázku tohoto typu programu je možné připravit v TEXu. Po jistém úsilí vznikl program, jenž vám předkládám. Na první pohled připomíná spíš poruchy na lince, nežli makro v TEXu, ale skutečně funguje, jak jistě každý nahlédne (tj. vyzkouší a uvěří výsledku). Komentáře nejsou vloženy (viz [2]: Opravdový programátor nepotřebuje poznámky – vlastní kód je zřejmý.). Navíc se domnívám, že komentář by stejně příliš nepomohl. Tak alespoň přikládám podobný program v jazyce C (ten však umí pouze česky, neboť standardní ošetření formátovaného výstupu nezná přehození parametrů).

Velikonoce patří k starobylým svátkům. Již pohané slavili příchod jara o prvojarní úplňkové noci (z těchto dob pochází i název велико ноцє). Křesťanská kultura umístila do těchto pohyblivých svátků oslavu Kristova zmrtvýchvstání. Tím došlo k posunu tohoto svátku na následující neděli. A tak nám zde naši předkové připravili matematický rébus – co nejjednodušeji vypočíst datum velikonoc. Jeden z největších matematických géniů – Carl Friedrich Gauß nám zde zanechal pravděpodobně nepřekonatelně jednoduchý algoritmus, díky němuž mohl vzniknout i tento článek.

Co se týče vlastního Gaussova algoritmu, žel neznám nikoho, kdo by ho pochopil a uměl rozumně vysvětlit. Kromě toho se domnívám, že důkaz jeho správnosti by vyžadoval (vzhledem k rozsahu tohoto bulletinu) několik jeho ročníků.

Lunisolární algoritmy lze napsat v TEXu samozřejmě i přehledněji. Příklad nalezneme v souboru `HEBCAL.STY` od Michaila Rozmana a Ramy Porratové, jenž je součástí hebrejského TEXu. Rozdíl délky souborů je však zřejmý (stonásobný).

Ještě malé upozornění: program by (alespoň dle mého úsudku) měl správně fungovat pouze pro data mezi roky 1900 a 2099. Implementace pro ostatní léta je též možná. Pro zájemce mám připravený věčný kalendář pro TEX, jenž náležitě ošetřuje i juliánský kalendář. Bude-li někdo v TEXu sázet třeba efemeridy, nic mu nebrání je v TEXu i počítat. Nakonec, jaký jiný typografický systém by něco podobného dokázal?

Použití je triviální: Do konzole napíšete „`csplain \year=`rok `\language=`jazyk `\input Easter`“ a program vytvoří `DVI` soubor s datem Velikonoc. Vynecháte-li zadání roku, počítá se letos, vynecháte-li zadání jazyka, bude to anglicky.
Příklad: „`csplain \year=1968 \language=\czech \input Easter`“.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                %
%           TeX algorithm to compute an Easter date              %
%          =========================================             %
%                                                                %
% File: Easter.tex v 1.5 95-04-18 12:20:00                       %
% Author: (c) 1995 Stanislav Brabec utx@k332.feld.cvut.cz        %
%                                                                %
% Copyright: Easter.tex is freely distributable as a sample of   %
%            compact astronomic computations in plain TeX        %
%                                                                %
% Usage:                                                         %
```

```
% virtex [&fmt] [\year[=]year] [\language[=]\xxx] \input Easter %
%  (writes result to Easter.dvi, not console, sorry)            %
%                                                               %
% Years available: 1900-2099                                    %
% Languages: czech, slovak, german, none specified = english    %
%       You can simply add any other.                           %
% Requires language token defined in format (from hyphen.lan)   %
%                                                               %
% Based on Easter algorithm of Carl Freidrich Gauss.            %
% Warning: Trying to understand the algorithm can take a very   %
%       long time and can cause a headache! Don't try it.       %
% ... (And if you find a shorter algorithm, let me know!)       %
% Note: Single-language algorithm can be even much shorter.     %
%                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\def~#1{\catcode`#113}~:\let:\let~+:+\advance~`:`\year~$
:$\newcount~^:^\def~[:[\ifnum~?:?\the~]:]\fi~*:*\multiply
~/~'~(~)~;$($'$)$;^/#1#2{(#1\divide(#2*(-#2+#1(}'`/'{19}*'19
+'24/'{30})5;`/;4*;2+);;`/;7*;4+);;'*;6+);/)7+)'+)21
[)>55+)-7](31^:{; }^'{+)1 [)>(+)-(^:{/ }]}'^*{}[)=(^*{:}]^~{?)}
^$#1#2#3#4{\ifx#1\undefined\else[\language=#1^;{#2}^/{#3}#4]]}
$0{March}{April}{Easter falls on :~.\ and '*~., ?`.}
$\czech{března}{dubna}{Velikonoce připadají na ~.\ *'a ~.\ :?`.}
$\slovak{marca}{apríla}{Veľkonočné sviatky pripadajú na ~.\ *'a ~.\ :?`.}
$\german{März}{April}{Die Ostern fiel sein am ~.\ *'und ~.\ :?`.}
\end
```

Program v jazyce C:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

char x[32],y[32];
struct tm *curtime;
char *day(char *x,const d,const short om)
{
sprintf(x,"%i. %s",d>31?d-31:d,(d>31&&!om?"dubna "
           :(d==31)||!om?"března ":""));
return(x);
}
int main(int argc,char *argv[])
{
int d,r;
time_t t;
if (argc==2) r=atoi(argv[1]); else {
  time(&t);
  curtime=gmtime(&t);
  r=1900+curtime->tm_year;
```

```
}
if ((d=22+(d=(19*(r%19)+24)%30)+(5+2*(r%4)+4*(r%7)+6*d)%7)>56) d=d-7;
printf("Velikonoce připadají na %sa %s%i.\n",day(x,d,1),day(y,d+1,0),r);
return(0);
}
```

Seznam literatury:

[1] Ivan Kopeček, Jan Kučera: *Programátorské poklesky*, Mladá Fronta, Praha 1989

[2] Ed Post: *Real Programmers Don't Use Pascal*, Datamation, July 1983, pp. 263–265 (Readers' Forum).

[3] D. E. Knuth: *The TEX Book*, Addison-Wesley Publishing, 1991

[4] Kalendáře na roky 1995, 1996, 1990 a 1991

Addendum:

Předminulý týden se mi dostal do ruky další geniální příklad z panoptika programů v [1] – program „želva“. Je jím program `wp2latex`. Na mém počítači (bez matematického koprocesoru) tráví přibližně 95 % veškerého času vypisováním toho, kolik procent je již převedeno (během převodu 100 kB souboru vyšle do konzole desítky až stovky tisíc znaků (polovina jsou znaky `BS`)). 3 % času tráví výpočtem toho, kolik procent je již převedeno (v aritmetice s dvojnásobnou přesností), zbylé 2 % času pak tráví převodem – to jest neustálým vyhledáváním znaků v naprosto neefektivně organizované tabulce. Takový dokonalý příklad „želvy“ nevymysleli ani autoři [1].

To je překrásné využití TEXu; pokud si ale pro začátek chcete TEXovskou aritmetiku vyzkoušet na něčem jednodušším, leč stejně praktickém, zkuste sestavit makro s jedním parametrem `#1`, které posune aktuální datum o `#1` dní, tedy např. `\advancedate{14}` udělá v roce 1995 z 26. února 12. března, ale následující rok správně 11. března (jedno z možných řešení najdete na konci tohoto čísla).

Karel Horák