

## INFORMATYKA I MATEMATYKA

Jan Rusinek

### PRZETWARZANIE DANYCH LICZBOWYCH W TEX-u. PRZYKŁAD ZASTOSOWANIA W DYDAKTYCE

[**słowa kluczowe.** TeX, algorytm sortowania, informatyzacja procesu dydaktycznego, ECTS]

#### Streszczenie

W pracy podany jest sposób wykorzystania pewnych poleceń w TeX-u w celu uzyskania możliwości używania i przetwarzania dużej liczby zmiennych liczbowych, możliwości formalnie w TeX-u niedostępnej. Podany jest przykład zastosowania w procesie dydaktycznym.

#### 1. Wstęp

Podstawowy TeX pozwala używać danych liczbowych po uprzednim zarezerwowaniu na nie miejsca poleceniem `\newcount\nazwa`. Niestety te dane liczbowe mogą być tylko całkowite, oraz jest miejsce zaledwie na około 250 danych. Jest też możliwość przechowywania danych jako „długości”, ale ilość też jest mocno ograniczona. Ponadto znaczna część może być już wykorzystana przez TeX-a i jego biblioteki jako parametry składu tekstu (np. dane liczbowe: numery stron, twierdzeń, rozdziałów; dane długości: różne odstępy, wielkości różnych czcionek, itp.).

TeX na szczęście posiada możliwość pamiętania tzw. „makr”, czyli poleceń. I tych może pamiętać około 25000. Sposób wykorzystania tego do „obsługi” większej ilości danych liczbowych jest zaprezentowany w tym artykule. Podany jest też przykład zastosowania w przetwarzaniu wyników egzaminu.

#### 2. Polecenie `\csname`

W zasadzie w TeX-u nazwy poleceń mogą się składać wyłącznie z małych i dużych liter alfabetu łacińskiego. Ograniczenie to pozwala ominąć polecenie

`\csname` ([1], [2], [3]) w połączeniu z poleceniem `\expandafter` ([2], [3]). Polecenie `\expandafter\ a\b` powoduje, że najpierw zostanie rozwinięte polecenie `\b`, a potem TeX rozwija `\a` i może wykorzystać **już rozwinięte** polecenie `\b`.

Pisząc więc `\expandafter\def\csname 21\endcsname{2014}` powodujemy, że TeX pod poleceniem `\21` pamięta napis „2014” (ale nie liczbę 2014!).

Wywołujemy go pisząc `\csname21\endcsname`. Możemy uzyskać liczbę 2014 pisząc `\x=\expandafter\csname21\endcsname` (po uprzednim zadeklarowaniu zmiennej `\x`). Mając już daną (jedną lub więcej) w postaci liczby można wykonać potrzebne operacje i zapisać wyniki pod nowym poleceniem (ponownie jako napis).

Jednak prawdziwą korzyść z omawianego powyżej polecenia otrzymamy w przypadku dużej liczby takich poleceń generowanych automatycznie.

Przypuśćmy, że mamy plik pod nazwą `lista.tex` zawierający wyniki egzaminu wygenerowane przez skaner do sprawdzania testów. Wygląda on np. tak:

```
1;61205;17;4,024
2;61206;02;3,022
3;61229;22;6,027
4;61230;20;5,027
5;61231;07;4,024
6;61233;35;5,026
7;61235;19;4,026
8;61236;08;10,032
9;61237;10;4,026
10;61238;31;6,024
```

Pierwsze pole zawiera kolejny numer, drugie numer indeksu studenta, trzecie numer zestawu, czwarte wynik egzaminu, pola są odseparowane średnikiem. (Oczywiście zwykle danych jest dużo więcej).

Chcąc teraz automatycznie przypisać poleceniu pod nazwą `\an` (n jest liczbą) n-tą linię pliku postępujemy następująco:

Definiujemy poniższą rekurencyjną definicję (po uprzednim zadeklarowaniu zmiennych):

```
\newdimen\dlu
\newcount\aa
\newcount\ilelinii
```

```

\newread\wczyt
\def\wczytanie{
\ifeof\wczyt
\let\next=\relax
\else
\read\wczyt to\mojalinia
\setbox\szerok=\hbox{\mojalinia}\ddd=\wd\szerok
\ifdim\ddd>0pt
\advance\aa1
\expandafter\edef\csname a\the\aa\endcsname{\mojalinia}
\let\next=\wczytanie
\fi\fi
\next}

```

TeX wczytując po kolei linie dopisuje na końcu pustą linię. Dlatego zastosowaliśmy „sztuczkę” z instrukcją warunkową sprawdzającą, czy wczytana linia ma dodatnią długość (`\ifdim\ddd>0pt`) i uwzględnianie w nowopowstałej bazie tylko linii o długości dodatniej.

Następnie wykorzystujemy to polecenie w poniższy sposób.

```

\openin\wczyt=lista.tex
\wczytanie
\closein\wczyt
\ilelinii=\aa

```

W tym momencie TeX pod poleceniem `\an` będzie pamiętał  $n$ -tą linię pliku `lista.tex`. Ponadto pod licznikiem `\ilelinii` będzie się kryła liczba danych, czyli liczba niepustych linii w pliku `lista.tex`.

### 3. Przetwarzanie danych

Teraz mamy możliwość bezpośrednio w TeX-u przetwarzania danych np. sortowania, permutowania ([4]), dzielenia na grupy itp.

Oto przykładowy algorytm sortowania. Zastosowano sortowanie bąbelkowe, aby do minimum zredukować konieczność pamiętania danych przez TeX-a (taka konieczność wystąpiłaby przy szybszych algorytmach).

Przypuśćmy, że chcemy posortować dane według ostatniego pola, czyli liczby punktów uzyskanych na egzaminie. W tym celu musimy najpierw rozdzielić dane w każdej linii pliku. Możemy to zrobić w sposób następujący:

Definiujemy polecenie rozdzielające te dane. Może mieć ono postać:

```
\def\rozdziel#1;#2;#3;#4,#5*{\def\jeden{#1}\ddd=#4.#5 pt}
```

Możemy przystąpić do właściwego sortowania. Proponowany algorytm jest następujący:

```
\newdimen\ddd
\newdimen\eee
\newcount\gg
\whiledo{\cc=0}{\cc=1
\aa=0\whiledo{\aa<\nn}
{\advance\aa1
\expandafter\rozdziel\csname a\the\aa\endcsname*
\gg=\aa
\advance\gg1
\eee=\ddd
\expandafter\rozdziel\csname a\the\gg\endcsname*
\ifdim\eee>\ddd
\cc=0
\edef\liniajeden{\csname a\the\aa\endcsname}
\edef\csname a\the\aa\endcsname{\csname a\the\gg\endcsname}
\edef\csname a\the\gg\endcsname{\liniajeden}
\else
\fi}}
```

Zamiast pętli z podstawowego TeX-a (`\loop`) autor zastosował pętlę `\whiledo` z pakietu `ifthen`.

Po presortowaniu mamy przyporządkowanie poleceniom `\a[liczba]` odpowiednich linii w kolejności rosnącej względem liczby uzyskanych punktów.

Może się jednak zdarzyć, że obrabiając dane (np. sortując), chcemy oprócz danych posortowanych zachować dane w kolejności wyjściowej. Wtedy trzeba definiować dwa ciągi danych: wyjściowe i posortowane. Niesie to niebezpieczeństwo „przeciążenia” pamięci TeX-a. Zwłaszcza, gdy korzystamy z dużej liczby bibliotek, a każda z nich ma wiele pomocniczych poleceń.

Wtedy może nam pomóc polecenie omawiane w następnym rozdziale.

#### 4. Polecenie `\write18`

Jeśli uruchomimy TeX-a z opcją `--shell-escape`, to możemy używać polecenia `\write18`. Polega ono na tym, że w miejscu wykonywania polecenia

```
\write18{program}
```

TeX zatrzymuje się, „wychodzi na zewnątrz”, wywołuje wykonanie programu `program`, po czym powraca na koniec powyższej linii i wykonuje kolejne linie pliku źródłowego.

Program mógł m.in. wyprodukować pewien plik, z którego TeX może korzystać w dalszych krokach.

W szczególności program może być również plikiem źródłowym TeX-a zastępującym pewną procedurę (np. sortowania) i zapisującym wynik swojego działania w pliku. Powoduje to, że uruchamiając tę procedurę jako oddzielny plik źródłowy można wykorzystać od początku pełną liczbę dopuszczalnych makr czy liczników.

Zastosowanie tej metody wydaje się bardzo praktyczne w sytuacji, gdy do gotowego już i sprawdzonego programu trzeba dodać dodatkową procedurę. Włożenie jej do środka programu naraża nas na niebezpieczeństwa: 1) wspomniana już możliwość przekroczenia dopuszczalnej pamięci TeX-a; 2) możliwość kolizji nowych oznaczeń ze starymi; 3) mimowolny błąd powodujący usterki w wyjściowym programie.

Wykonanie procedury jako zewnętrzny program, wprawdzie może nieco zwolnić wykonanie całego programu, ale za to te niebezpieczeństwa redukuje prawie do zera.

## 5. Przykład

Na koniec zaprezentujemy konkretny problem rozwiązany w opisywany powyżej sposób. Autor miał gotowy program do przetwarzania wyników egzaminu ([5]). W pewnym momencie pojawiła się potrzeba, aby program udoskonalić poprzez automatyczne dostosowanie wyników do wymogów sugerowanych przez Europejski System Transferu i Akumulacji Punktów (ECTS) ([7]). Są one następujące: wśród ocen pozytywnych ma być po 10% ocen A i E (u nas 5 i 3), po 25% ocen B i D (u nas 4,5 i 3,5) oraz 30% ocen C (u nas 4).

W początkowym programie progi ocen były ustalane przez egzaminatora.

Okazuje się, że aby te progi otrzymywać zgodnie z powyższymi wymogami wystarczy wykonać następujące kroki:

- 1) ustalić tylko próg, od którego jest ocena dostateczna;
- 2) zebrać dane zawierające **wyłącznie** wyniki punktowe powyżej progu z punktu 1) (bez konieczności zachowania przypisania ich studentowi!);
- 3) posortować je;
- 4) na podstawie tej posortowanej listy automatycznie ustalić progi ocen spełniające żądane warunki;

5) wykorzystać te progi tak samo jak progi ustalane przez wykładowcę w podstawowym programie.

Zostało to przez autora zrealizowane w omawiany w artykule sposób. Oto fragment podstawowego programu, w którym to fragmencie wywołujemy nowy program:

```
...
%
\immediate\write18{latex prog1.tex}
\input{param.wun}
%
\oceny
\else\fi
\def\abc{h }\ifx\abc\cowybierasz\clr
\immediate\write16{OD ILU PUNKTOW JEST DOSTATECZNY+?}
\read-1 to\pukty\trojkaplus=\pukty pt\oceny\else\fi
\def\abc{i }\ifx\abc\cowybierasz\clr
\immediate\write16{OD ILU PUNKTOW JEST DOBRY?}
\read-1 to\pukty\czworka=\pukty pt\oceny\else\fi
\def\abc{j }\ifx\abc\cowybierasz\clr
\immediate\write16{OD ILU PUNKTOW JEST DOBRY+?}
\read-1 to\pukty\czworkaplus=\pukty pt\oceny\else\fi
\def\abc{k }\ifx\abc\cowybierasz\clr
\immediate\write16{OD ILU PUNKTOW JEST BARDZO DOBRY?}
\read-1 to\pukty\piatka=\pukty pt\oceny\else\fi\next}
...
```

Jak widzimy latex wywołuje zewnętrzny program TeX-owy `prog1.tex`. Program ten wygląda następująco:

```
\documentclass{article}
\usepackage{ifthen}
\newcount\aa
\newcount\bb
\newcount\nn
\newcount\gg
\newdimen\bbb
\newdimen\ddd
\newcount\nnmj
\begin{document}
```

```

\newdimen\punktydodos
\immediate\write16{OD ILU PKT. JEST DOSTATECZNY?}
\read-1 to\pukty
\punktydodos=\pukty pt
\def\wydziel#1;#2;#3;#4,#5*{\edef\wydz{#4.#5 pt}}
\def\wczytanie{
\ifeof\wczyt
\let\next=\relax
}else
\read\wczyt to\mojalinia
\setbox\szerok=\hbox{\mojalinia}\ddd=\wd\szerok
\ifdim\ddd>0pt
\advance\aa1
\expandafter\wydziel\mojalinia*
\bbb=\wydz
\ifdim\bbb<\punktydodos
}\else
\expandafter\edef\csname a\the\aa\endcsname{\wydz}
\fi
\let\next=\wczytanie
\fi\fi
\next}
\nn=\aa
%WCZYTYWANIE
\immediate\openout3=lista.tex
\wczytanie
\closeout3
\nnmj=\nn\advance\nnmj-1
%SORTOWANIE
\whiledo{\cc=0}{
\cc=1
\aa=0\whiledo{\aa<\nnmj}
{\advance\aa1
\bbb=\csname a\the\aa\endcsname
\gg=\aa
\advance\gg1
\ddd=\csname a\the\gg\endcsname
\ifdim\bbb>\ddd
%przestawianie
\cc=0

```

```

\expandafter\edef\csname a\the\aa\endcsname{\the\ddd}
\expandafter\edef\csname a\the\gg\endcsname{\the\bbb}
\else
\nie przestawianie
\fi
}}
%USTALANIE PROGOW
\newdimen\trop %prog dla 3+
\newdimen\czw %prog dla 4
\newdimen\czwp %prog dla 4+
\newdimen\pia %prog dla 5
\bb=\nn\multiply\bb10\divide\bb100
\advance\bb1
\trop=\csname a\the\bb\endcsname
\immediate\write16{trojka plus od \the\trop}
\bb=\nn\multiply\bb35\divide\bb100
\advance\bb1
\czw=\csname a\the\bb\endcsname
\immediate\write16{czworka od \the\czw}
\bb=\nn\multiply\bb65\divide\bb100
\advance\bb1
\czwp=\csname a\the\bb\endcsname
\immediate\write16{czworka plus od \the\czwp}
\bb=\nn\multiply\bb90\divide\bb100
\advance\bb1
\pia=\csname a\the\bb\endcsname
\immediate\write16{piatka od \the\pia}
%ZAPISYWANIE PROGOW DO PLIKU param.wun
\immediate\openout3=param.wun
\immediate\write3{\string\trojka=\the\punktydodos}
\immediate\write3{\string\trojkaplus=\the\trop}
\immediate\write3{\string\czworka=\the\czw}
\immediate\write3{\string\czworkaplus=\the\czwp}
\immediate\write3{\string\piatka=\the\pia}
\immediate\closeout3
\end{document}

```

Program ten wykonuje wszystkie zadania 1)–5) i zapisuje wyniki swojej pracy w pięciolinijkowym pliku param.wun, który zawiera wygenerowane automatycznie progi ocen 3+, 4, 4+ i 5 (w wypadku większej liczby prac z „gra-



niczną” liczbą punktów program dobiera oceny „na korzyść studenta”). Plik ten jest następnie czytany i wykorzystywany przez program podstawowy. Potem wykładowca może wprowadzić swoje korekty do progów wygenerowanych automatycznie.

### ***Bibliografia***

- [1] Abrahams P., Hargreaves K., Berry K. (1990); *TeX for the Impatient*, Addison-Wesley
- [2] Eijkhout V., *TeX by Topics. A TeXnician's Reference*, <http://www.gust.org.pl/doc/documentation>
- [3] Knuth D. E. (2005); *TeX Przewodnik użytkownika*, WNT
- [4] Rusinek J. (2007); *Algorytm permutowania w TeX-u zastosowany do informatyzacji procesu egzaminacyjnego*, „Rocznik Naukowy Wydziału Zarządzania w Ciechanowie”, 1-4 (I), (153–174)
- [5] Rusinek J. (2008); *Pliki do odczytu i zapisu w TeX-u – zastosowanie do przetwarzania wyników egzaminu*, „Rocznik Naukowy Wydziału Zarządzania w Ciechanowie”, 1-2 (II), (107–124)
- [6] Rusinek J. (2009); *Testy egzaminacyjne z matematyki*, „Rocznik Naukowy Wydziału Zarządzania w Ciechanowie”, 3-4 (III), (101–111)
- [7] (2009); *Europejski System Transferu i Akumulacji Punktów, przewodnik użytkownika*, Fundacja Rozwoju Systemu Edukacji [ekspercibolonscy.org.pl/sites/ekspercibolonscy.org.pl/files/przewodnik\\_ECTS\\_2009\\_pol.pdf](http://ekspercibolonscy.org.pl/sites/ekspercibolonscy.org.pl/files/przewodnik_ECTS_2009_pol.pdf)