Ach, te kształty

Bogusław Jackowski

Jakoś tak człowiek został skonstruowany, że w żadnej dziedzinie nie wystarcza mu jeden „kanoniczny” kształt – ani w życiu, ani w sztuce. Sztuka drukarska nie jest tu wyjątkiem, a kształtem, o którym mowa, jest w tym przypadku kształt akapitu.

Najnaturalniejszy technologicznie kształt – prostokąt – drukarze zaburzają nagminnie za pomocą wcięcia akapitowego. Prostokąt ostał się tu i ówdzie, na przykład niektóre poradniki korespondencji zalecają formatowanie akapitów bez wcięć, nie liczyłbym jednak na powszechny tryumf prostokąta, zwłaszcza w kontekście wszechobecności komputerowych systemów składu tekstu. Graficy komputerowi wprost prześcigają się w nadawaniu akapitom kształtów, jakich dotąd nie bywało, nawet za cenę czytelności tekstów.

Wcięcie z przodu, z tyłu, z góry i z dołu

TEX nie jest szczególnie życzliwie usposobiony do „rzeźbiarzy akapitów”, ale nie jest też bynajmniej usposobiony wrogo. Najprostszy przypadek, który daje się wyrazić w TEX-owym języku, to akapit z prostokątnym wcięciem niekoniecznie jednego wiersza i niekoniecznie w lewym górnym rogu akapitu. D. E. Knuth określa taki kształt jako *hanging indentation* (wcięcie zawieszone).

Służą do tego dwa parametry (rejestry, *registers*): `\hangindent` – wymiar (*dimen*) określający wielkość wcięcia, i `\hangafter` – liczba (*count*) określająca liczbę wierszy bez wcięcia. Kruczek polega na tym, że parametrom tym mogą być nadane wartości dodatnie bądź ujemne. Mamy cztery możliwe kombinacje znaków, którym odpowiadają cztery różne sytuacje. Przypuśćmy, że d jest wymiarem wpisanym do rejestru `\hangindent`, a n – liczbą wpisaną do rejestru `\hangafter`, wówczas

- jeśli $d > 0$, to oznacza to, że na wymiar d ma być wcięta *lewa* strona łamu;
- jeśli $d < 0$, to oznacza to, że na wymiar $|d|$ ma być wcięta *prawa* strona łamu;
- jeśli $n \geqslant 0$, to oznacza to, że wiersze $1, 2, \ldots, n$ mają zwykłą szerokość, a wiersze $n+1$, $n+2$, … są *krótsze* (w szczególności gdy $n = 0$, to wszystkie wiersze są krótsze);
- jeśli $n < 0$, to oznacza to, że wiersze $1, 2, \ldots, |n|$ są *krótsze*, natomiast wiersze $|n|+1$, $|n|+2, \ldots$ mają zwykłą szerokość.

A oto „zminiaturyzowany” przykład różnych realizacji wcięcia zawieszonego:

`\hangindent=30pt` `\hangafter=3`

(1) Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał…

`\hangindent=-30pt` `\hangafter=3`

(2) Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał…

`\hangindent=30pt` `\hangafter=-3`

(3) Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał…

`\hangindent=-30pt` `\hangafter=-3`

(4) Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał: Wpadł pies do kuchni i porwał mięsa ćwierć, a jeden kucharz głupi zarąbał go na śmierć. A drugi kucharz mądry, co litość w sercu miał, postawił mu nagrobek i napis taki dał…

O kilku sprawach związanych z parametrami `\hangafter` i `\hangindent` warto pamiętać:

(a) Instrukcja `\par` powoduje ustawienie `\hangafter=1` i `\hangindent=0pt`, tzn. parametry te są *lokalne* dla danego akapitu.

(b) Przy składaniu akapitu obowiązują ustawienia aktualne w momencie wykonywania instrukcji `\par` – pamiętanie o tym chroni przed zdziwieniem, dlaczego np. konstrukcja `{\it \item{`*coś*`}` *jakiś tekst*`}\par` nie działa.

(c) Na interpretację parametru `\hangafter` ma wpływ parametr `\prevgraf`. Na ogół wartość tego parametru po wykonaniu instrukcji `\par` jest równa liczbie wierszy ostatnio składanego akapitu, natomiast w momencie rozpoczęcia składania akapitu wartością parametru `\prevgraf` jest zero. Nadanie mu wartości *po rozpoczęciu* akapitu powoduje, że TEX działa tak, jakby w tym momencie zostało już przetworzonych `\prevgraf` wierszy bieżącego akapitu.

Związek między wartością rejestru `\prevgraf` i przebiegiem składania akapitu może wydać się na pierwszy rzut oka mało użyteczny. Jak zobaczymy dalej, pozwala on na przenoszenie stosownej informacji między akapitami, a w konsekwencji – na nadawanie kształtów za pomocą ustawień parametrów `\hangafter` i `\hangindent`, a zwłaszcza za pomocą instrukcji `\parshape`, nie tylko akapitom, ale i całym łamom.

Wszystkie warianty ustawień rejestrów `\hangindent` i `\hangafter` mogą być przydatne przy oblewaniu tekstem prostokątnych ilustracji.

Wariant oznaczony jako (1) w powyższym przykładzie można wykorzystać w indeksach rzeczowych (wówczas \hangindent>0, \hangafter=1); ustawienia \hangindent=\parindent, \hangafter=0 znajdujemy w definicji plain-owego makra \item; wariant oznaczony jako (3) przydaje się do wstawiania inicjałów.

Niestety, wstawianie inicjałów w TEX-u nie jest „bezpieczne" ze względu na brak parametru zakazującego łamania strony po wybranym wierszu akapitu (łamania po pierwszym wierszu akapitu można uniknąć przez ustawienie \clubpenalty=10000, łamania przed ostatnim wierszem – przez ustawienie \widowpenalty=10000), w związku z czym może się zdarzyć, że litera inicjału wystaje poniżej światła strony. Jedynym na to lekarstwem jest wstawienie w odpowiednim miejscu akapitu polecenia \vadjust{\nobreak}, pamiętać wszakże należy o ewentualnej zmianie tego miejsca w razie wprowadzenia poprawek w akapicie.

Orgia kształtów

Nieco osobliwa instrukcja \parshape pozwala na dowolne kształtowanie lewego i prawego brzegu akapitu, co można uznać za uogólnienie zabiegów dokonywanych za pomocą ustawiania rejestrów \hangindent i \hangafter. Osobliwość polega na tym, że formalnie rzecz biorąc \parshape jest rejestrem typu *count* (np. \the\parhsape daje w wyniku liczbę), ale używa się go inaczej niż zwykłych rejestrów liczbowych.

Wypowiedziane wyżej uwagi (a) – (c), odnoszące się do wcięcia zawieszonego, dotyczą w znacznej mierze również instrukcji \parshape. Popatrzmy, co D. E. Knuth ma do powiedzenia na ten temat („The TEXbook", str. 101 – 103):

Praktycznie dowolny kształt akapitu można uzyskać za pomocą instrukcji postaci \parshape = ⟨*liczba*⟩*, gdzie* ⟨*liczba*⟩ *oznacza liczbę całkowitą dodatnią* n*, po której następuje ciąg* $2n$ *wymiarów. W ogólności polecenie* \parshape = n p_1 s_1 p_2 s_2 … p_n s_n *definiuje akapit, którego pierwszych* n *wierszy ma szerokość odpowiednio* s_1, s_2, …, s_n*, a każdy z tych wierszy jest przesunięty w stosunku do lewego marginesu o* p_1, p_2, …, p_n*. Jeśli akapit liczy mniej niż* n *wierszy, to nadmiarowa specyfikacja zostanie zignorowana; jeśli wierszy jest więcej niż* n*, to specyfikacja dla ostatniego wiersza powtarzana jest ad infinitum. Instrukcja* \parshape=0 *anuluje poprzednią specyfikację* \parshape.

[…]

Jeśli zarówno \parshape *jak i wcięcie zawieszone są określone równocześnie, to pod uwagę brana jest instrukcja* \parshape. *Zwykły kształt akapitu, w którym każdy wiersz ma szerokość* \hsize*, otrzymuje się dla* \parshape=0, \hangindent=0pt *i* \hangafter=1. *TEX automatycznie odtwarza te naturalne ustawienia na końcu każdego akapitu i (na zasadzie lokalności) za każdym razem, gdy przystępuje do składania w wewnętrznym trybie pionowym.*

[…]

Jeśli eksponowane równanie (displayed equation) występuje w akapicie o niestandardowym kształcie, TEX zawsze zakłada, że równanie to zajmuje dokładnie trzy wiersze. Na przykład jeśli akapit liczy cztery wiersze, po których występuje równanie, po którym występują jeszcze dwa wiersze, to zostanie on uznany przez TEX-a za akapit dziewięciowierszowy: $4 + 3 + 2 = 9$*; równanie zostanie przesunięte i umieszczone w osi stosownie do definicji* \parshape *dla szóstego wiersza.*

[…]

TEX posiada wbudowaną zmienną liczbową (rejestr typu count) o nazwie \prevgraf*, o wartości równej liczbie wierszy ostatniego akapitu, który został całkowicie lub częściowo złożony. Użytkownik może używać rejestru* \prevgraf *w takim kontekście, w jakim używane są liczby, może też nadać zmiennej* \prevgraf *dowolną wartość nieujemną chcąc, by TEX sądził, że jest w określonym obszarze aktualnie składanego akapitu o nietypowym kształcie. Załóżmy jak poprzednio, że akapit zawiera cztery wiersze plus eksponowane równanie matematyczne plus jeszcze dwa wiersze. Przystępując do składania akapitu TEX nadaje zmiennej* \prevgraf *wartość 0; kiedy rozpoczyna składanie równania,* \prevgraf *uzyska wartość 4; kiedy równanie jest gotowe – wartość 7; kiedy akapit jest gotowy – wartość 9. Jeżeli równanie w rzeczywistości zajmuje o jeden wiersz więcej, można użyć instrukcji* \prevgraf=8 *na początku dwóch ostatnich wierszy; wówczas TEX pomyśli, że składany jest akapit dziesięciowierszowy. Wartość zmiennej* \prevgraf *jest brana pod uwagę jedynie przy niestandardowych ustawieniach* \parshape *i* \hangindent.

Uff, sporo tego, nieprawdaż? Myślę, że objętość powyższego opisu można potraktować jako przestrogę, zwłaszcza dla tych, którzy – parafrazując J. Tuwima – wyznają zasadę: „nic nie wiem i nic nie rozumiem, a wszystko umiem". Moim zdaniem nie można w sensowny sposób używać *żadnych* makr, nie mając świadomości, co się za tym kryje – chociaż być może jest to moje archaiczne myślenie...

Ale do rzeczy. Co z tym dobrodziejstwem informacji można zrobić? Oczywiście – nadawać akapitom dziwaczne kształty. Można też, jak zostało powiedziane wyżej, nadawać kształty całym łamom. Ponieważ pierwszy z aspektów jest omówiony szerzej w artykule J. M. Nowackiego w niniejszym numerze biuletynu („Przepis na wygodne używanie komendy \parshape"), a także – rzecz jasna – w podręczniku „The TEXbook", skupię się na aspekcie drugim.

Oto bodaj najprostsza „sztuka", jaką można wykonać za pomocą instrukcji \parshape przenosząc starannie informację o liczbie dotychczas złożonych wierszy między akapitami (skład dwułamowy wybrałem z przyczyn czysto estetycznych):

Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.

Odpowiedź na pytanie „jak to się robi?" [1] sprowadza się do zauważenia, że kształt łamu musi się powtarzać co pewną liczbę wierszy, powiedzmy n (w tym przypadku $n = 28$). Oznacza to, że: (1) \parshape musi definiować co najmniej n par wymiarów – w istocie trzeba użyć większej ich liczby, ale o tym za chwilę; (2) jakoś chytrze trzeba przenosić informację między akapitami w przypadku, gdy akapity liczą mniej wierszy niż n. „Chytrość" polega na wykorzystaniu rejestru \prevgraf według następującego algorytmu:

> *Niech* n *oznacza długość cyklu, czyli liczbę wierszy, po której kształt łamu się powtarza, a* p *niech będzie resztą z dzielenia sumarycznej liczby wierszy już złożonych akapitów przez* n *(*$p < n$*).*

POCZĄTEK AKAPITU

$\rightarrow$ *podstaw* \prevgraf $= p$, *a następnie wykonaj instrukcję* \parshape *definiującą odpowiednio dużą liczbę par wymiarów;*

KONIEC AKAPITU

$\rightarrow$ *zastąp* p *przez* \prevgraf *(jeśli* w *oznacza liczbę wierszy ostatnio złożonego akapitu, to w momencie zakończenia składania akapitu* \prevgraf $= p + w$*); jeśli* $p \geqslant n$, *to zastąp z kolei* p *przez* $p - n$.

Łatwo formalnie określić, co to jest „odpowiednio duża liczba par wymiarów", mianowicie jest to największa wartość zmiennej \prevgraf pojawiająca się na końcu akapitu. Wyznaczenie tej liczby przed złożeniem ciągu akapitów jest praktycznie niemożliwe i zresztą niepotrzebne, bo z dobrym przybliżeniem możemy oszacować tę liczbę od góry przez $n + m$, gdzie m oznacza liczbę wierszy najdłuższego akapitu. Chociaż precyzyjne wyznaczenie *a priori* liczby m również graniczy z niemożliwością, to jednak bez trudu można ją oszacować od góry z dużym zapasem, co w tym przypadku w zupełności wystarcza.

W powyższym przykładzie przyjąłem, że spełniona jest nierówność $m \leqslant n$ i że tym samym wystarczy zdefiniować \parshape dla $2n = 56$ wierszy. Dokładniej, zdefiniowane zostało makro \setparshape w następujący sposób:

```
\def\setparshape{\parshape 56
    0pt \hsize  2pt \hsize  4pt \hsize
    6pt \hsize  8pt \hsize 10pt \hsize
   12pt \hsize 14pt \hsize 16pt \hsize
   ...
   52pt \hsize 54pt \hsize
% koniec 1-go cyklu
    0pt \hsize  2pt \hsize  4pt \hsize
    6pt \hsize  8pt \hsize 10pt \hsize
   12pt \hsize 14pt \hsize 16pt \hsize
   ...
   52pt \hsize 54pt \hsize
% koniec 2-go cyklu
}
```

1: Pomijam sprawy związane z użyciem formatu wielołamowego – techniki tu omawiane nie zależą w istotny sposób od liczby łamów na stronie.

Makro to zostało wykorzystane w dość oczywistej realizacji opisanego wyżej algorytmu:

```
\newcount\nlin % w opisie algorytmu: n
\newcount\plin % w opisie algorytmu: p
\def\cyclicparshape#1{%
  \nlin=#1\relax
  \everypar{%          początek akapitu
    \prevgraf\plin
    \setparshape
  }%
  \def\par{%             koniec akapitu
    \endgraf
    \plin=\prevgraf
% if p>=n then p:=p-n:
    \ifnum\plin<\nlin \else
      \advance\plin by -\nlin
    \fi
  }
}
```

Umieszczając bezpośrednio przed składanym tekstem instrukcję \cyclicparshape{28} (oczywiście liczba *28* odnosi się do tego konkretnego przypadku) uzyskuje się żądany efekt.

Ot i wszystko. No, prawie wszystko. Warto zauważyć, że kształty akapitów niekoniecznie muszą być wypisywane pracowicie przez użytkownika. Ponieważ TEX jest systemem otwartym, chętnym do współpracy z programami przetwarzającym pliki tekstowe, można więc i inne programy zaprząc do pomocy. Może to być CorelDRAW! + AWK (jak we wspomnianym artykule J. M. Nowackiego), może być METAFONT albo METAPOST, może być PERL – może to być naprawdę cokolwiek, z czego umiemy skorzystać.

Jednym z programów nadających się do wyliczania definicji \parshape jest sam TEX. W omawianym przypadku można za pomocą prostej pętli skonstruować makro \setparshape:

```
\newcount\currshift %      \currshift=0
\def\setparshape{}%       inicjalizacja
\loop
  \edef\setparshape{%       kontynuacja
    \setparshape
    \number\currshift pt %      wcięcie
    \hsize %                 szerokość
  }%
 \ifnum\currshift<54
   \advance\currshift by 2
\repeat
\edef\setparshape{%        zakończenie
  \parshape56
  \setparshape \setparshape
}
```

Można oczywiście bawić się w cyzelowanie i uogólnianie powyższej definicji, ale nie o to tu chodzi, ważna jest sama idea konstruowania za pomocą pętli stosownego ustawienia \parshape.

Niewielkie modyfikacje powyższej definicji pozwalają na uzyskanie całej gamy kształtów, takich jak na przykład:

Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota!
Poszli razem do Niedzieli, tam porządnie
odpoczęli. Tydzień drapie się w przedziałek –
no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.

czy też:

Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.
Mieli pracę skończyć w Piątek, a to ledwie był
początek.
Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek.
Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek.
Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek.
Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek.
Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?
Poniedziałek już od Wtorku poszukuje kota
w worku, Wtorek Środę wziął pod brodę
– chodźmy sitkiem czerpać wodę! Czwartek
igłą w górze grzebie i zaszywa dziury w niebie.

Pokrótce omówię pierwszą z modyfikacji, drugą zostawiając czytelnikowi jako proste ćwiczenie wraz z zachętą do wymyślania innych modyfikacji.

Pomysł zasadza się na niezbyt głębokiej obserwacji matematycznej: jeżeli $f(x)$ jest parabolą, to ciąg $f(2)-f(1)$, $f(3)-f(2)$, $f(4)-f(3)$, ... tworzy postęp arytmetyczny. Ta obserwacja pozwoli nam na wygenerowanie krzywej przypominającej sinusoidę, będącej fragmentami parabolą (gdyby TEX miał wbudowaną funkcję sinus, można by sobie tych korowodów oszczędzić).

Zaczniemy od TEX-owej maszynki generującej „piłokształtny” ciąg liczb $0, 1, 2, \ldots, q-1, q, q-1, \ldots, 1, 0, -1, \ldots, -q+1, -q, -q+1, \ldots, -1, 0, 1, \ldots$ itd. Jest to ciąg okresowy o okresie $4q$, będący fragmentami postępem arytmetycznym. Jeśli $a_1, a_2, a_3, \ldots$ jest takim właśnie ciągiem, to – zgodnie z powyższą obserwacją – ciąg $s_1 = a_1$, $s_2 = a_1 + a_2$, $s_3 = a_1 + a_2 + a_3$, ... będzie szukaną „pseudo-sinusoidą”. Oczywiście ciąg $\{s_i\}$ jest również okresowy i również ma okres $4q$.

Nietrudno sprawdzić, że zdefiniowane poniżej makro \nextaseq generuje ciąg $\{a_i\}$:

```
\newcount\qcycle %          w opisie: q
\newcount\pmone %       zawsze +1 lub -1
\newcount\aseq % kolejny element ciągu
\def\nextaseq{%
  \ifnum\aseq=\qcycle
    \ifnum\pmone>0 \pmone=-\pmone \fi
  \else \ifnum\aseq=-\qcycle
    \ifnum\pmone<0 \pmone=-\pmone \fi
  \fi\fi
  \advance\aseq by \pmone
}
```

Aby się o tym przekonać, wystarczy albo chwilę pomyśleć, albo uruchomić prosty program testowy zawierający powyższą definicję i pętlę:

```
\qcycle=4 %         przykładowa wartość
\pmone=1 %      początkowo ciąg rośnie
\aseq=-1 %  następny element ciągu = 0
\loop
\nextaseq
\iftrue %                nieprzerwanie
\message{\the\aseq} %  pisz po ekranie
\repeat
```

TEX w nieskończoność będzie wypisywał na ekran ciąg liczb:

```
0 1 2 3 4 3 2 1 0 -1 -2 -3 -4 -3 -2 -1
0 1 2 3 4 3 2 1 0 -1 -2 -3 -4 -3 -2 -1
0 1 2 3 4 3 2 1 0 -1 -2 -3 -4 -3 -2 -1
0 1 2 3 4 ...
```

Generowanie ciągu $\{s_i\}$ to już doprawdy dziecinna igraszka:

```
\newcount\sseq % kolejny element ciągu
\def\nextsseq{%
  \nextaseq
  \advance\sseq by \aseq
}
```

podobnie zresztą jak modyfikacja definicji makra \setparshape, która jako że odwołuje się do zmiennej \nlin, powinna wystąpić *po wywołaniu* makra \cyclicparshape:

```
⟨definicje \nextaseq i \nextsseq⟩
% dodatkowe deklaracje: =============
\newcount\tmpc %   rejestr pomocniczy
\newdimen\tmpd %   rejestr pomocniczy
\newdimen\iunit %        miara wcięcia
% inicjalizacja zmiennych: ==========
\qcycle=4 %      przykładowa wielkość
\iunit=3pt %     przykładowa wielkość
\pmone=1 %         wartość początkowa
\aseq=-1 %         wartość początkowa
\tmpc1 %           wartość początkowa
\cyclicparshape{38}%   przykł. użycie
% wyznaczanie \setparshape: =========
\def\setparshape{}%     inicjalizacja
\loop
  \nextsseq \tmpd=\sseq\iunit
  \edef\setparshape{%     kontynuacja
    \setparshape
    \the\tmpd \hsize
  }%
  \ifnum\tmpc<\nlin
    \advance\tmpc by 1
\repeat
\tmpc\nlin \advance\tmpc by \nlin
% \tmpc = 2\nlin
\edef\setparshape{%       zakończenie
  \parshape\the\tmpc\space
  \setparshape\setparshape
}
```

Zauważmy, że makro \setparshape zostało tym razem zdefiniowane w nieco ogólniejszy sposób – procedura wyznaczania makra nie zawiera występujących w sposób jawny liczb. Dalsze uogólnienie (nie jest dla mnie oczywiste, czy warte zachodu) mogłoby polegać na parametrycznym specyfikowaniu maksymalnej liczby wierszy w akapicie, powyższa definicja bowiem wykorzystuje założenie, że żaden akapit nie liczy więcej niż \nlin wierszy.

Kształty mniej obfite

Parę lat temu zetknąłem się z następującym zadaniem: *należy złożyć tekst w taki sposób, by wcięcie akapitowe było równe szerokości ostatniego wiersza poprzedniego akapitu*[2]:

Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek. Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?

Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek. Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?

Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek. Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?

Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek. Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?

Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek. Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?

Poniedziałek już od Wtorku poszukuje kota w worku, Wtorek Środę wziął pod brodę – chodźmy sitkiem czerpać wodę! Czwartek igłą w górze grzebie i zaszywa dziury w niebie. Mieli pracę skończyć w Piątek, a to ledwie był początek. Zamyśliła się Sobota – to dopiero jest robota! Poszli razem do Niedzieli, tam porządnie odpoczęli. Tydzień drapie się w przedziałek – no a gdzie jest Poniedziałek?

Można się zgodzić, że zadanie jest nieco egzotyczne, ale cóż zrobić – artysta każe, TEX-owiec musi... Nie wdając się więc w dyskusję nad sprawami estetyki, skupmy się nad rozwiązaniem zadania w TEX-u.

Problem sprowadza się do następującego pytania: *czy od TEX-a można się dowiedzieć, jaka jest długość ostatniego wiersza akapitu?* Odpowiedź jest częściowo pozytywna, co jest trochę zaskakujące, jako że D. E. Knuth bardzo się postarał, żeby wszelkie liczby związane z wynikowym rozmieszczeniem tekstu na stronie nie dostały się w niepowołane ręce, to jest w ręce TEX-owca.

Istnieje uzasadniony powód tej pozornie przesadnej ostrożności. Otóż TEX niektóre obliczenia wykonuje zmiennoprzecinkowo, co przyśpiesza proces składania, ale zarazem powoduje, że wynik obliczeń zależy od konkretnie użytego procesora. Gdyby użytkownik (zwłaszcza szalony, a można spokojnie założyć, że każdy, kto używa TEX-a, jest do pewnego stopnia szalony) miał dostęp do takich liczb, to wynik składu na różnych maszynach mógłby być drastycznie inny, a przecież jednym z podstawowych założeń przy tworzeniu TEX-a była pełna przenośność oprogramowania między różnymi platformami sprzętowymi. Ukrycie wyników obliczeń zmiennoprzecinkowych przed użytkownikiem powoduje, że chociaż pliki `DVI` wygenerowane z tego samego źródła na różnych komputerach mogą się różnić, to jest to różnica praktycznie bez znaczenia, natomiast wszystkie pliki tekstowe generowane przez TEX-a (plik `LOG` oraz pliki tworzone za pomocą instrukcji `\write`) są identyczne, niezależnie od implementacji.

Rozwiązanie postawionego zadania jest możliwe dzięki istnieniu wbudowanej zmiennej typu *dimen* o nazwie `\predisplaysize`. Na ogół `\predisplaysize=0pt`, jedynie po rozpoczęciu składania równania eksponowanego, to znaczy po napotkaniu dwóch znaków dolara `$$`, TEX wpisuje do rejestru `\predisplaysize` wartość określającą szerokość wiersza bezpośrednio poprzedzającego równanie.

Zanim przejdziemy do szczegółów rozwiązania, zajrzyjmy – podobnie jak to uczyniliśmy w przypadku instrukcji `\parshape` – do podręcznika „The TEXbook" (str. 188):

TEX nadaje specjalne wartości trzem zmiennym typu ⟨dimen⟩ bezpośrednio po napotkaniu znacznika `$$` *rozpoczynającego równanie:* `\displaywidth` i `\displayindent` *otrzymują wartości równe odpowiednio szerokości z i przesunięciu* s *wiersza o numerze* `\prevgraf`$+2$ *w zależności od bieżącego ustawienia* `\parshape` *lub zawieszonego wcięcia. (Zazwyczaj* `\displaywidth` *ma taką samą wartość jak* `\hsize`*, a* `\displayindent` *ma wartość zero, aczkolwiek kształt akapitu może być zmienny, jak to zostało opisane w rozdziale 14.) Ponadto do rejestru* `\predisplaysize` *wpisywana jest efektywna szerokość* p *wiersza poprzedzającego równanie zgodnie z następującymi regułami: jeżeli przed równaniem nie wystąpił żaden wiersz (na przykład jeśli znacznik* `$$` *wystąpił bezpośrednio po instrukcji* `\noindent` *lub po zamykającym inne równanie znaczniku* `$$`*),* p *przyjmuje wartość* -16383.99999pt *(*$-$`\maxdimen`*, czyli najmniejszy dopuszczalny wymiar). W przeciwnym razie TEX sprawdza pudełko utworzone przez poprzedni wiersz i określa wartość* p *jako równą szerokości wnętrza pudełka, plus przesunięcie pudełka w prawo, plus dwie jednostki em z bieżącego fontu. Jeśli jednak szerokość wnętrza pudełka zależy od ściśliwości lub rozciągliwości odstę-*

2: W istocie makro opisane w niniejszym rozdziale powstało przy innej okazji. Mniejsza jednak o okazję, ciekawostką jest data powstania: makro to wykorzystuję w postaci praktycznie nie zmienionej od roku 1990 – czy ktoś dziś pamięta, jakie „edytory" wiodły prym w owym czasie i wie, co się z nimi teraz dzieje?

pów – jeśli na przykład klej \parfillskip *ma wartość skończoną – wówczas* p *przyjmuje wartość* \maxdimen. *(Sytuacja taka nie pojawia się często, ale dzięki przyjętym regułom TEX jest sprzętowo niezależny, ponieważ* p *nie zależy od wielkości, które mogłyby być różnie zaokrąglane na różnych komputerach.) Zauważmy, że na wartości zmiennych* \displaywidth *i* \displayindent *nie mają wpływu kleje* \leftskip *oraz* \rightskip, *natomiast kleje te mają wpływ na wartość zmiennej* \predisplaysize.

Jak widać z powyższego opisu, na ogół – choć nie zawsze – można uzyskać od TEX-a informację o szerokości wiersza poprzedzającego równanie eksponowane.

Chciałoby się więcej, dobre jednak i to. Poniżej przedstawione zostało makro \meashor pozwalające w zasadzie zmierzyć odległość dowolnego miejsca akapitu od lewego brzegu. Określenie „w zasadzie" jest tu jak najbardziej na miejscu, bowiem proces mierzenia wiąże się z wstawieniem w danym miejscu akapitu równania eksponowanego w sposób możliwie niezaburzający składu, aczkolwiek dość zasadniczego zaburzenia uniknąć się nie da: wstawienie równania eksponowanego oznacza, że materiał poprzedzający równanie i materiał następujący po równaniu jest składany jakby to były dwa osobne akapity, w związku z czym odstępy w otoczeniu miejsca pomiaru mogą być niejednolite. Dlatego zalecałbym korzystanie z tego makra głównie do mierzenia ostatniego wiersza akapitu.

Wystarczy już teoretyzowania – oto definicja makra \meashor.

```
\newif\ifmeasdisp %                ,,kosmetyka''
\measdispfalse %              ustawienie domyślne
\newcount\measnum %                         wynik
\newdimen\measdim %                         wynik
\def\meashor{%
  \ifhmode
    \bgroup %  zmiany parametrów są lokalne
      \leftskip0pt \rightskip0pt
      \parfillskip=0pt plus 1fil
      \predisplaypenalty=10000
      \postdisplaypenalty=10000
      \displaywidowpenalty=\widowpenalty
      \abovedisplayskip=-\baselineskip
      \belowdisplayskip=-\baselineskip
      \abovedisplayshortskip=-\baselineskip
      \belowdisplayshortskip=-\baselineskip
      \null % pozostaw poprzedzającą spację
      $$\global\measdim=\predisplaysize$$%
      \ifnum\prevgraf=0
        \immediate\write16{!
          perhaps restricted hmode:
          MEASURING IGNORED}%
      \else
        \global\advance\measdim by -2em
        \global\measnum=\prevgraf
        \global\advance\measnum by -3
        \ifmeasdisp
          \immediate\write16{>>
            \the\measnum: \the\measdim}%
        \fi
% \advance\prevgraf to operacja zabroniona
        \global\advance\measnum by -1
        \global\prevgraf=\measnum
        \global\advance\measnum by 1
        \kern\measdim
      \fi
    \egroup
  \else
    \immediate\write16{! vmode:
      MEASURING IGNORED}%
  \fi
}
```

W kodzie makra parametry związane z równaniami eksponowanymi są zmieniane lokalnie tak, by zminimalizować wspomniane efekty uboczne pomiaru; samo równanie eksponowane – główny punkt makra – służy jedynie do zarejestrowania wartości zmiennej \predisplaysize; obliczenia w rodzaju \advance \measdim by -2em czy \advance \measnum by -3 powinny być jasne po przestudiowaniu cytowanych wyjaśnień z podręcznika „The TEXbook". Poza tym pojawiają się komunikaty, niespecjalnie skomplikowane – i to wszystko.

Użycie makra polega na jego wywołaniu w wybranym miejscu akapitu. Na przykład w tym zdaniu umieściłem instrukcję \meashor bezpośrednio przed słowem „Na" rozpoczynającym zdanie. W wyniku zadziałania makra rejestr \measnum (numer wiersza w którym wykonany został pomiar) uzyskał wartość 2, a rejestr \measdim (odległość punktu pomiaru od lewej krawędzi akapitu) – wartość 111.52782pt. Proszę sprawdzić, na pewno się zgadza. Makro \meashor powinno być wywołane w trybie składania akapitu. Wywołanie makra w trybie pionowym lub w ograniczonym trybie poziomym, tj. wewnątrz konstrukcji \hbox, spowoduje zignorowanie polecenia pomiaru i stosowny protest.

Wybór miejsca pomiaru w poprzednim akapicie podyktowany został chęcią unaocznienia ubocznych skutków takiego pomiaru – odstępy po obu stronach miejsca pomiaru nieznacznie acz *zauważalnie* się różnią. Użyta technika nie pozwala na wyeliminowanie tego efektu. Jedyną w pełni skuteczną, a przy tym niezaburzającą techniką ustalania położenia na stronie jest analiza pliku DVI za pomocą dodatkowych programów – ale to już zupełnie inna historia.

Wróćmy więc do naszych baranów. Mając makro \meashor bez trudu możemy rozwiązać postawione zadanie. W tym celu wystarczy umieścić bezpośrednio przed składanym tekstem następujące instrukcje:

```
\global\measdim=0pt %       inicjalizacja
\global\measnum=0 %         inicjalizacja
\everypar{\kern-\parindent\kern\measdim}
\def\par{\meashor\endgraf}
```

Nie wygląda to zawile, prawda? Komentarza być może wymaga użycie atrybutu \global przy inicjalizacji zmiennych \measdim i \meashor. Otóż nadawanie zmiennym wartości raz globalnie, a raz lokalnie obciąża znacznie bardziej pamięć TEX-a niż konsekwentne trzymanie się globalności lub lokalności. Ponieważ makro \meashor z natury rzeczy wykorzystuje ustawienia globalne, więc i tu użyłem atrybutu \global. Ale w tym prostym przypadku nie ma to większego znaczenia.

Uważny czytelnik na pewno w tym miejscu węszy oszustwo – czy rozwiązanie aby nie jest *zbyt* proste? Owszem, jest. Nie zostały rozpatrzone sytuacje, w których wartością rejestru \measdim jest ±\maxdimen. Uwzględnienie tych przypadków nie jest trudne, a wchodzenie zbyt głęboko w szczegóły techniczne zaciemniłoby jedynie obraz.

Wypada na ostatek odnotować, że makro \meashor może się przydać nie tylko do tak egzotycznych zastosowań. Korzystam zeń często na przykład do ustalania pionowego odstępu między akapitem a znaczkiem wstawianym po akapicie (mogą być to umieszczane w osi strony trzy gwiazdki bądź zaczerniony prostokąt umieszczany z prawej strony na końcu artykułu lub dowodu matematycznego) – im krótszy wiersz kończący akapit tym mniej światła należy zostawić. Makro \meashor świetnie się tu sprawdza.

Kształty nie w TEX-a guście

Ktoś, kto wyniósłby z niniejszego artykułu wrażenie, że TEX żadnych kształtów (akapitów) się nie lęka, byłby w błędzie.

Po pierwsze, niemal wszystkie fajerwerki akapitowe wykorzystujące instrukcję \parshape biorą w łeb, jeżeli interlinia nie jest sztywna. W takim przypadku automatyczne oblewanie tekstami ilustracji, które mają wyznaczone miejsce na stronie, jest – bez przetwarzania pliku DVI na zewnątrz TEX-a – po prostu niemożliwe.

Po drugie, nawet tak proste zadanie, jak zmiana szerokości łamu przy zmianie strony, nie daje się zrealizować w TEX-u automatycznie. Bierze się to stąd, że TEX najpierw składa akapity, a potem dopiero łamie kolumny – gdy akapit jest już złożony, to nic nie jest w stanie zmienić długości wierszy. Jedyny ratunek to ręczne wymuszanie łamania stron.

O kłopotach z pozycjonowaniem inicjałów już wspominałem.

Owe niedostatki TEX-a (powyższa lista nie pretenduje do bycia kompletną) dają się czasem we znaki w praktyce, ale przecież nie przekreślają ogromnych możliwości TEX-a, jeżeli chodzi o kształowanie akapitów, zwłaszcza przy wspomaganiu ze strony innych programów. A poza tym – czy tylko TEX ma wady?

Prywatny słowniczek

Ponieważ nie istnieją jak na razie powszechnie akceptowane polskie określenia niektórych pojęć związanych z TEX-em, pozwalam sobie przypomnieć używane przeze mnie w tekście nieoczywiste tłumaczenia trzech takich pojęć:

internal variable – zmienna wbudowana
hanging indentation – wcięcie zawieszone
displayed equation, *display* – równanie eksponowane

Podziękowania

Jestem wdzięczny Januszowi Nowackiemu, Markowi Rydko, Staszkowi Wawrykiewiczowi, Włodkowi Bzylowi, Rysiowi Kubiakowi oraz Tomkowi Przechlewskiemu za inspirację i poświęcony mi czas.

◇ Bogusław Jackowski
jacko@ipipan.gda.pl