PLAIN

Luka z TEXbooka. Tryby

Bogusław Lichoński

Luka w mojej TEX-owej wiedzy

Uczymy się TEX-a przez lata, jednak bardzo trudno cokolwiek zrobić bez dobrej książki na biurku. Mimo iż uczę się TEX-owej maszyny mniej więcej od pięciu lat, ciągle o czymś zapominam lub łapię się na tym, że czegoś nie doczytałem lub pominąłem. Do białej gorączki doprowadzają mnie odkrycia, że przerobiony dawniej temat umknął mi gdzieś z pamięci. Muszę czytać od nowa :-(.

Jedyną receptą na moje bolączki stało się robienie notatek z naszej TEX-owej biblii, czyli z *The TEXbook*. Notatki te nie są zawsze wierną kopią TEXbook-a, mają jednak identyczny układ tematyczny. Marzy mi się po cichu dobre tłumaczenie mojej ulubionej książki, czy jest to jednak możliwe? Każdego, kto ma na ten temat jakieś refleksje lub pomysły, proszę o kontakt.

Jeśli zechcesz więc, drogi czyTEXniku, zajrzeć do moich notatek, to zapraszam. Dzisiaj zapiski z rozdziału trzynastego TB „Tryby".

Tryby

Podobnie jak każdy z nas jest w danym momencie w pewnym nastroju, tak TEX jest w pewnym trybie. Istnieje sześć trybów TEX-a:

- *tryb pionowy* (w tym trybie TEX tworzy główną listę pionową [1], z której tworzone są strony);
- *wewnętrzny tryb pionowy* (tworzy listę pionową vbox-a);
- *tryb poziomy* tzw. „akapitowy" (tworzy listę poziomą — akapit);
- *ograniczony tryb poziomy* (tworzy listę poziomą, np. hbox-a);
- *tryb matematyczny* (tworzy formuły matematyczne w trybie poziomym `$...$`);
- *tryb matematyczny eksponowany* (tworzy eksponowane formuły matematyczne umieszczane w osobnej linii, tymczasowo przerywają one aktualny akapit — `$$...$$`).

Sytuacja najprostsza. W zasadzie użytkownik nie musi zastanawiać się, w jakim trybie w danym momencie jest TEX. Problemy pojawią się dopiero wtedy, gdy ujrzymy komunikat w stylu *Nie możesz zrobić czegoś w ograniczonym trybie poziomym.* Aby uniknąć takich niemiłych niespodzianek, musimy wiedzieć, w jakim trybie znajduje się TEX w **każdym** dowolnym momencie.

W zasadzie TEX jest w trybie pionowym, gdy tworzy listę pionowych kar, klejów (np. `\vskip`) i pudełek (np. `\vbox`). Pudełka i kleje umieszczane są jedno pod drugim na stronie. Tryb poziomy tworzą kary, kleje (np. `\hskip`) i pudełka poziome umieszczane jedno za drugim, tworząc akapit. Tryb matematyczny tworzą wszystkie elementy umieszczone między znakami dolara '$'.

Sytuacja typowa. Na samym początku TEX znajduje się w trybie pionowym i jest gotowy do tworzenia listy pionowej (a potem strony). Jeśli utworzymy klej, karę lub pudełko, TEX zacznie umieszczać je jedno za drugim. Spójrzmy na krótki źródłowy pliczek plain-TEX-owy:

```
1. Gdańsk 1996
2. \vskip 1cm
3. \centerline {GUST}
4. \medskip
5. Co to jest \TeX?
6. \vfil
7. \break
8. $x$
9. \bye
```

Praktycznie każda linijka tego pliku tworzy osobny element na liście pionowej. Prześledźmy, w jakie tryby właściwie „wpadał" TEX.

- Linijka 1. wciągnęła TEX-a w tryb poziomy (ach ten Gdańsk!).
- Komenda z linii 2. spowodowała zakończenie akapitu i przejście w tryb pionowy; wszystko jasne – mamy tu przecież klej pionowy.
- Linia trzecia rozpoczyna się znaną komendą `\centerline`≡`\hbox to\hsize`, a więc TEX wkracza w ograniczony tryb poziomy.
- W linii 4. znów napotykamy na klej pionowy (przypomnijmy, że `\medskip` ≡ `\vskip 6pt plus 2pt minus 2pt`).
- Linia 5. to znów powrót do trybu poziomego.
- Linia 6. to klej pionowy (przypomnijmy, że `\vfil`≡`\vskip0pt plus1fil`).

- W linii 7. mamy karę, która spowoduje złamanie strony (\break $\equiv$ \penalty10000).
- Nowa strona rozpocznie się od przejścia do trybu matematycznego
- i bye*.

Z przedstawionego przykładu wynikają pewne ogólne wnioski. Gdy TeX znajduje się w trybie pionowym lub wewnętrznym trybie pionowym, to pierwszy żeton (ang. *token*) nowego akapitu zmienia tryb na poziomy. Tryb ten utrzymuje się aż do zakończenia akapitu. Inaczej mówiąc obiekty typu poziomego automatycznie zmieniają tryb pionowy na poziomy. Dzieje się tak wówczas, gdy napiszemy jakikolwiek znak (kategoria 11.) lub pojawi się któryś z żetonów \char, \accent, \hskip, \␣, \vrule, czy też znak początku trybu matematycznego '$'. Idąc dalej, TeX wstawi pudełko o szerokości wcięcia akapitowego i odczyta jeszcze raz dany żeton już w trybie poziomym.

Istnieje możliwość celowego zmuszenia TeX-a do przejścia w tryb poziomy. Nieufnym proponuję makra \indent i \noindent, które zmieniają tryb na poziomy oraz odpowiednio tworzą lub zakazują utworzenia wcięcia akapitowego. \indent wstawia na początku akapitu puste pudełko o szerokości \parindent (domyślnie jest to 20pt). Dlatego też ciąg niewinny \indent \indent spowoduje wstawienie podwójnego wcięcia. Analogiczny ciąg \noindent \noindent nie spowoduje nic, poza tym co zrobi pojedynczy \noindent.

Jak zacząć akapit od hbox-a. Jeśli umieścimy \hbox{...} w trybie poziomym, TeX złoży akapit uwzględniając w nim pudełko poziome (na przykład TeX nigdy nie złamie takiego pudełka). Podobnie jeśli umieścimy \hbox{...} w trybie pionowym, TeX złoży stronę uwzględniając takie pudełko.

No dobrze, ale jak rozpocząć akapit pudełkiem \hbox{...}?

Kilka linijek wyżej udzielono właściwie odpowiedzi na to pytanie. Można zastosować konstrukcję typu \indent \hbox{...}, nie jest to jednak zbyt rozsądne. Wyobraźmy sobie bowiem, że chcemy zdefiniować makro, które rozwija się do hbox-a. Ponadto chcemy, aby działało poprawnie zarówno na początku, jak i w środku akapitu — w takim wypadku użycie makra \indent nie jest zachęcające.

Plain TeX posiada złoty środek, jakim jest makro \leavevmode, które rozwija się do \unhbox \voidbox.

W „trybach" TeX-a

W trybie poziomym i matematycznym. W typowych publikacjach TeX większość czasu spędza w trybie poziomym, składając akapity. Na krótko robi wycieczki do trybu pionowego pomiędzy nimi. Akapit kończy się, gdy użytkownik wyda polecenie \par lub pozostawi w pliku źródłowym pustą linię (pusta linia jest zamieniana na \par według zasad z rozdziału 8. TB). Akapit uda się także zakończyć, wydając polecenie \vskip 1cm (patrz linia 2. powyższego przykładu). Nie musimy wpisywać wówczas komendy \par, gdyż klej pionowy vskip oczywiście nie może być częścią akapitu i spowoduje przejście TeX-a do trybu pionowego.

Przełącznik trybu matematycznego (żeton '$') umieszczony w trybie poziomym spowoduje, że TeX „zanurkuje" w tryb matematyczny do czasu napotkania drugiego żetonu '$'. TeX dołączy formułę matematyczną do akapitu i powróci do trybu poziomego. Na przykład

```
Myślę $\ldots$, więc jestem.
Różnimy się o~$\varepsilon$.
$\alpha$-Centauri
```

Inaczej jest w eksponowanym trybie matematycznym. Jeśli użyjemy pod rząd dwóch żetonów '$$', TeX przerwie składanie akapitu (dokona przy tym różnych obliczeń, na przykład wyliczy sobie długość ostatniej linijki poprzedzającej wzór) i wyśle dotychczas złożoną część akapitu na listę pionową. Wskoczy następnie w tryb matematyczny eksponowany, złoży formułę (koniec formuły oznaczamy kolejną parą żetonów '$$') i powróci do trybu poziomego. Jeśli akapit nie został zakończony, TeX dalej będzie go składał. Na przykład

```
Liczba $$\pi \approx 3.1415926536$$
jest bardzo ważna.
```

spowoduje złożenie następującego akapitu:

*: Na końcu dokumentu plain-TeX-owego umieszczamy zwykle makro \bye, które jest równoważne \vfill \eject \end. Klej \vfill przełącza TeX-a w tryb pionowy, \eject kończy stronę, zaś \end kończy pracę TeX-a.

Liczba
$$\pi \approx 3.1415926536$$
jest bardzo ważna.

W wewnętrznym trybie pionowym. TEX wędruje do wewnętrznego trybu pionowego, gdy użyjemy jednej z komend \vbox, \vtop, \vcenter, \valign, \vadjust, \insert.

W ograniczony tryb poziomy TEX wpada, gdy zastosujemy \hbox lub \halign. W późniejszych rozważaniach (patrz też rozdziały 12. i 21. TB) dowiemy się bardziej szczegółowo, iż różnice między wywnętrznym trybem pionowym a trybem pionowym oraz ograniczonym trybem poziomym a trybem poziomym są niewielkie. Jadnak trzeba pamiętać, że trybów tych TEX używa do różnych celów.

Różny nastrój w różnym trybie

Gdy TEX połyka żeton, musi podjąć decyzję, co z nim zrobić. Decyzja TEX-a w znacznym stopniu zależy od trybu, w jakim się znajduje.

Przykładowo makro \break działa zupełnie inaczej w trybie pionowym (łamie stronę — patrz powyższy przykład) i poziomym (łamie linijkę akapitu). Dwa znaki '$$' użyte w trybie poziomym spowodują przejście TEX-a do trybu matematycznego eksponowanego, ale użycie ich w ograniczonym trybie poziomym spowoduje utworzenie pustej matematycznej formuły i może powodować komunikaty o błędach.

TEX korzysta z faktu, że niektóre operacje są niedopuszczalne w danym trybie, i pozwala nam unikać błędów, które mogą pojawiać się w naszych dokumentach.

TEX będąc w danym trybie, zwykle przerywa pracę, aby wykonać jakieś zadanie w innym, a następnie wraca do trybu pierwotnego. Na przykład użycie '\hbox{' w dowolnym trybie „wrzuca” TEX-a w ograniczony tryb poziomy do czasu napotkania nawiasu zamykającego '}'. W tym sensie TEX może znajdować się w kilku trybach naraz, jednak sposób działania TEX-a zależy od trybu, w którym jest w danym momencie „najgłębiej”; tryby zewnętrzne, w których jest zagnieżdżony, nie obchodzą go.

Zdarzyć się może, że zupełnie nie wiemy, w jakim trybie znajduje się TEX. Co wtedy zrobić? Najlepiej zapytać o to największego eksperta w tych sprawach, czyli samego TEX-a. Spójrzmy na plik tryby.tex:

```
1. \tracingcommands=1
2. \hbox{
3. $
4. \vbox{
5. \noindent$$
6. xA\showlists
7. $$}
8. $}
9. \bye
```

Pierwsza linia tego pliku zmusza TEX-a do zapisywania w pliku tryby.log informacji o każdej komendzie, którą napotka (TEX robi tak, gdy wartość \tracingcommands jest dodatnia). Oto jakie nowe informacje pojawią się w pliku tryby.log:

```
{vertical mode: \hbox}
{restricted horizontal mode: blank space}
{math shift character $}
{math mode: blank space}
{\vbox}
{internal vertical mode: blank space}
{\noindent}
{horizontal mode: math shift character $}
{display math mode: blank space}
{the letter x}
{the letter A}
...
```

A więc wiadomości są następujące:

TEX, będąc w trybie pionowym, ujrzał żeton \hbox, co zmusiło go do wejścia w ograniczony tryb poziomy.

Pierwszym żetonem, jaki ujrzał, była spacja, jako że koniec linii oznacza dla TEX-a właśnie spację (patrz koniec linii 2.)

Będąc ciągle w ograniczonym trybie poziomym zobaczył przełącznik trybu matematycznego — żeton '$', co spowodowało przejście do trybu matematycznego.

Pierwszym żetonem jaki ujrzał była znowu spacja, która w tym trybie nic nie znaczy — jest ignorowana.

Następnie TEX napotyka na żeton \vbox, czyli witamy w wewnętrznym trybie pionowym, aby po ujrzeniu żetonu \noindent wejść w tryb poziomy.

Niezbyt długo trwał ten stan, gdyż dwa znaki $$ zanurzyły TEX-a w eksponowany tryb matematyczny. Bul, bul, bul...

Bardzo interesująca staje się zawartość pliku tryby.log po komendzie \showlists. Jest to kolejna bardzo użyteczna możliwość śledzenia poczynań TEX-a. Dzięki niej możemy dowiedzieć się rzeczy, które TEX zwykle trzyma w ukryciu.

\showlists powoduje bowiem wypisanie list, które buduje TEX w aktualnym trybie oraz we wszystkich trybach nadrzędnych (czyli tych, w których zawiera się tryb aktualny).

```
### display math mode entered at line 5
\mathord
.\fam1 x
### internal vertical mode entered at line 4
prevdepth ignored
### math mode entered at line 3
### restricted horizontal mode entered at line 2
\glue 3.33333 plus 1.66666 minus 1.11111
spacefactor 1000
### vertical mode entered at line 0
prevdepth ignored
```

Zauważmy, że w momencie, gdy pojawiło się \showlists (linia 6.), TEX zdążył już **pięć** razy zmienić tryb. Trybem bieżącym w linii 6. jest eksponowany tryb matematyczny (patrz linia 5.), który zawiera jeden znak 'x' z rodziny 1. (patrz TB rozdział 17.).

Rozważany przez nas eksponowany tryb matematyczny zawarty jest w wewnętrznym trybie pionowym (linia 4.). Wewnętrzna lista pionowa, którą TEX tworzy w tym trybie, jest pusta — pojawiająca się spacja

```
\vbox{␣
```

jest ignorowana. Komunikat:

```
prevdepth ignored
```

oznacza, że wartość \prevdepth* jest mniejsza lub równa −1000pt, więc najbliższy klej międzylinijkowy (*interline glue*) w ogóle się nie pojawi (patrz TB rozdział 12.).

Wewnętrzny tryb pionowy zawiera się w trybie matematycznym (linia 3.) — lista wewnętrzna jest pusta (spacja po znaku '$' jest ignorowana).

Pusta natomiast nie jest lista wewnętrzna ograniczonego trybu poziomego (linia 2.), zawiera ona klej. Skąd się wziął? Otóż w tym trybie spacja

```
\hbox{␣
```

nie jest ignorowana.

---

*: Specjalny wymiar \prevdepth przechowuje głębokość ostatniego pudełka z listy pionowej.

Ostatecznie wszystko zawiera się w głównym trybie pionowym. Komunikat:

```
entered at line 0
```

oznacza, że tryb ten pojawił się, zanim TEX zaczął czytać plik tryby.tex.

Ciekawostki

Ciekawostka 1. Czy można stworzyć TEX-em dokument, w którym TEX nigdy nie opuszcza trybu pionowego?

Odpowiedź: Można, dokument taki składać się może z klejów pionowych \vskip oraz linii utworzonych komendą \hrule.

Ciekawostka 2. Niektóre z trybów TEX-a nie mogą bezpośrednio zawierać innych trybów; na przykład wewnętrzny tryb poziomy nie może zawierać eksponowanego trybu matematycznego, nawet jeśli formuła matematyczna (*display*) pojawia się w środku akapitu. Dzieje się tak, gdyż budowany (w trybie poziomym) i przerwany akapit jest zawsze kończony i usuwany z pamięci TEX-a zanim formuła się rozpocznie. Czy potrafisz stworzyć charakterystykę wszystkich kolejnych par trybów, jakie mogą pojawić się w pliku .log po użyciu komendy \showlists?

Odpowiedź: Pozostawiam ten problem jako rozrywkę umysłową dla szanownych czyTEXników. Powodzenia! (Odpowiedź można znaleźć w TB str. 313 zad. 13.5).

Bardzo dziękuję red. Jackowi Foromańskiemu za unicestwienie wielu błędów czyhających w tym artykule. CzyTEXnik z pewnością dostałby zawrotu głowy, gdyby zobaczył wersję przed korektą Jacka.

Bibliografia

[1] Marek Ryćko. *Lista pionowa.* Grupa Użytkowników Systemu TEX, III Konferencja w Bachotku, Kwiecień 1995.

[2] Donald E. Knuth. *The TEXbook.* Addison-Wesley, 1984.

◇ Bogusław Lichoński
b.lichonski@gwo.com.pl