PLAIN

Luka z TeXbooka. Pudełka

Bogusław Lichoński

Luka w naszej TeX-owej wiedzy

Uczymy się TeX-a przez lata, jednak bardzo trudno cokolwiek zrobić bez dobrej książki na biurku. Mimo iż korzystam z TeX-a od ponad dziesięciu lat, ciągle o czymś zapominam lub łapię się na tym, że czegoś nie doczytałem.

Jedyną receptą na moje bolączki stało się robienie notatek z naszej TeX-owej biblii, czyli z *The TeXbook* (TB). Notatki te nie zawsze są wierną kopią TeXbook-a, mają jednak identyczny układ tematyczny. Marzy mi się po cichu dobre tłumaczenie mojej ulubionej książki, czy jest to jednak możliwe?

Jeśli zechcesz więc, drogi czytelniku, zajrzeć do moich notatek, to zapraszam. Dzisiaj zapiski z rozdziału jedenastego TB „Pudełka".

Czy TeX to Murarz?

Czy zastanawialiście się kiedyś z czego właściwie TeX „buduje" stronę? Jeśli spojrzymy do TB na str. 62, zobaczymy tam zapracowanego TeX-a budującego z zapałem przedziwną konstrukcję. Elementami tej budowli są pudełka połączone klejem. TeX w tym momencie przypomina nieco murarza, który buduje ścianę z cegieł.

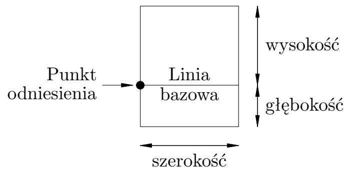
Podobieństwo do ściany jest dosyć duże, gdyż rzeczywiście dwoma głównymi elementami z jakich TeX składa stronę są *pudełka* (ang. *boxes*) oraz *klej* (ang. *glue*). Są to dwa bardzo ciekawe obiekty, które warto lepiej poznać, jeśli chcemy wiedzieć coś więcej o pracy naszego ulubionego systemu.

Co to są pudełka?

TeX-owe pudełka są w sensie matematycznym dwuwymiarowymi obiektami w kształcie prostokąta. Posiadają one trzy mierzalne określające je parametry, są to: *wysokość* (ang. *height*), *szerokość* (ang. *width*) i *głębokość* (ang. *depth*).

Pudełka posiadają jeszcze dwa istotne parametry, są to punkt odniesienia (ang. *reference point*) i linia bazowa pudełka (ang. *baseline*).

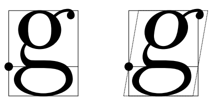
Spójrzmy na poniższy rysunek, który przedstawia typowe pudełko TeX-owe oraz opisujące je parametry.



Pudełka zawierające jeden znak. Chyba jednym z najprostszych rodzajów pudełek jest pojedynczy znak z fontu. Z punktu widzenia TeX-a każdy znak jest pudełkiem. O tym jaka będzie wysokość szerokość i głębokość oraz o tym jaki symbol znajdzie się w pudełku decyduje typograf. TeX używa tych wymiarów do określenia położeń wszystkich punktów odniesień pudełek na stronie.

Na przykład w TeX-owym formacie *plain* litera „h" antykwy `\rm`, czyli fontu `cmr10`, ma wysokość 6.9444, szerokość 5.5555, i głębokość 0 punkta; litera „g" ma wysokość 4.3055, szerokość 5 i głębokość 1.9444 punkta. Zauważmy, że tylko niektóre specjalne znaki, takie jak nawiasy mają sumę wysokości i głębokości rzeczywiście równą dziesięciu punktom, pomimo że mówi się, iż `cmr10` jest fontem dziesięciopunktowym. Nie musisz oczywiście uczyć się tych wymiarów na pamięć, ale dobrze jest zdawać sobie sprawę do czego ta informacja jest potrzebna TeX-owi.

Kontur znaku nie musi zawierać się dokładnie w swoim pudełku. Na przyład niektóre znaki, które są używane do tworzenia większych symboli matematycznych, takich jak nawiasy macierzy, celowo wystają troszeczkę poza „swoje" pudełko, tak aby optycznie obejmowały zawartość macierzy. Również litery pochylone wystają niekiedy po prawej lub lewej stronie pudełka, aby uniknąć „białych dziur" wokół liter. Pudełka są prostokątami, więc nie można ich przekrzywiać na boki! Dla przykładu porównajmy literę „g" z fontu `cmr10` i z fontu `cmsl10` (`\rm` i `\sl`):



W obu wypadkach TeX przyjmuje, że pudełka mają szerokość 5 punktów, a więc obie litery są traktowane w ten sam sposób. TeX nie orientuje się, jak będzie wyglądał drukowany znak –

to wie tylko urządzenie wyjściowe. Pochyłe litery zostaną jednak rozmieszczone właściwie, pomimo „niedostatecznej" wiedzy TEX-a, ponieważ linie pisma będą sobie odpowiadały. Pamiętajmy, że TEX-a nie obchodzi w ogóle kształt znaku lecz tylko jego pudełko! Dlatego podczas kompilacji TEX zagląda tylko do plików `.tfm`, gdzie odczytuje m.in. wymiary pudełek. Kształty znaków interesujące są tylko dla urządzeń wyjściowych, takich jak podglądarki, drukarki czy naświetlarki.

Zwykle twórcy fontów informują TEX-a między innymi o tak zwanej *korekcie kursywy*. Dla każdego znaku podana jest liczba, która wskazuje, jak daleko dany font wystaje w prawo poza granicę swojego pudełka, plus małe „coś". Na przykład korekta kursywy dla litery „g" w foncie `cmr10` wynosi 0.1389 pt, natomiast dla `cmsl10` – 0.8565 pt. W rozdziale 4 TB podano, że ta korekta dodawana jest do normalnej szerokości znaku, jeżeli zaraz po nim wstawisz „`\/`". Powinieneś więc pamiętać o użyciu `\/` wtedy, gdy następuje zmiana fontów z pochyłego na wyprostowany, a zwłaszcza w takim wypadku:

```
tak zwana {\sl korekta kursywy\/}:
```

gdy nie ma przestrzeni na zrekompensowanie straty związanej z pochyleniem fontu.

Czarne pudełka. TEX ma do czynienia także z innym rodzajem pudełek, które można by nazwać *czarnymi pudełkami*, a mianowicie z prostokątem takim jak ▮, który zostanie całkowicie zaczerniony w czasie drukowania. Możesz określić wysokość, szerokość i głębokość takiego pudełka – lepiej byłoby jednak, aby nie zajmował zbyt dużej powierzchni, mogłoby to bowiem zdenerwować drukarkę.

Zwykle te czarne pudełka są bardzo wąskie, tak że sprawiają wrażenie *linii poziomych* (ang. *horizontal rule*) lub *linii pionowych* (ang. *vertical rule*). Drukarze tradycyjnie nazywają takie obiekty liniami. W środowisku TEX-a do utworzenia czarnych pudełek używane są polecenia `\hrule` i `\vrule`. Nawet jeśli pudełko jest kwadratem, jak tu ▪, musiałeś użyć `\hrule` lub `\vrule`. Więcej interesujących szczegółów na temat czarnych pudełek znajdziesz w rozdziale 21 TB.

W zwartym szeregu. Wszystko to, co zostało na stronie złożone TEX-em, jest kombinacją przeróżnych pudełek. Można powiedzieć, że TEX ustawia pudełka na dwa sposoby: *poziomo* lub *pionowo*.

Gdy TEX tworzy poziomą listę pudełek, ustawia je w linii tak, aby ich punkty odniesienia znalazły się na tym samym poziomie, dlatego linie pisma sąsiednich znaków będą pasować do siebie. Podobnie, gdy TEX buduje pionową listę pudełek, układa je tak, by ich punkty odniesienia znalazły się w tej samej pionowej kolumnie.

Spójrzmy teraz za kulisy TEX-a, aby porównać metody komputerowe z tym, co musiałbyś zrobić, gdybyś składał ręcznie metalowe czcionki. Robiąc to tradycyjną metodą, wybierasz potrzebne ci czcionki z kaszty drukarskiej. Wielkie litery znajdują się w górnej kaszcie, układasz je w *wierszowniku*. Gdy wiersz tekstu jest ułożony, wyrównujesz odstępy, dobierając odpowiedni *justunek*. Rezultat przekładasz do *szufelki drukarskiej* przy pomocy *zecliny*, gdzie łączysz go z innymi wierszami pisma. (Przypomnę, że *zeclina*, czyli *składaczka*, to mosiężna linia z występującymi z obu boków uszkami, dostosowana do szerokości składanego wiersza, ułatwiająca ustawianie czcionek w wierszowniku oraz przenoszenie składu na szufelkę.)

Nie różni się to wiele od tego co robi TEX, oprócz tego, że używa się tu trochę innych określeń. Gdy TEX tworzy linie, powstaje „coś", co nazywa się „`\hbox`" (pudełko poziome). Dzieje się tak, ponieważ „składniki" linii są połączone ze sobą razem w poziomie. Zauważmy, że owe składniki są również pudełkami poziomymi. Możesz w swoim tekście źródłowym użyć instrukcji:

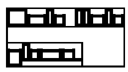
```
\hbox{Linia pisma.}
```

Jest to polecenie dla TEX-a pobrania pudełek znakowych odpowiednich liter z bieżącego fontu i zamknięcia ich w pudełku poziomym. Litera „L" jest więc pudełkiem „□", a litera „p" jest „□". Tak więc podana instrukcja sprawia, że TEX formuje pudełko poziome następującej postaci:

przedstawiające napis „Linia pisma". W końcu pojedyncze linie pudełek poziomych umieszczane są razem w pudełku pionowym. Na przykład, zapis:

```
\vbox{\hbox{Dwie linie}\hbox{pisma.}}
```

TEX przetworzy to na:



Dwie linie
pisma.

Główna różnica pomiędzy TEX-em, a składem tradycyjnym jest taka, że metalowe czcionki są zwykle odlewane tak, aby każdy znak miał taką samą wysokość i głębokość; ułatwia to ręczne składanie wierszy. Znaki TEX-a mają zmienną wysokość i głębokość, gdyż komputer nie ma trudności z ułożeniem znaków wzdłuż linii pisma. Poza tym dodatkowe informacje o wysokości i głębokości pomagają w usytuowaniu akcentów i symboli matematycznych.

Inną ważną różnicą pomiędzy składem TEX-a, a ręcznym jest fakt, iż TEX będzie dzielił linie automatycznie, zecer musiał robić to ręcznie. Oczywiście nie musisz używać poleceń `\hbox` i `\vbox` podczas składu zwykłego tekstu. Jeżeli jednak zechcesz decydować, gdzie należy wstawić każdą literę z osobna i użyjesz poleceń `\hbox` i `\vbox`, to będziesz przypominał starego drukarza i zecera Bena Franklina. (Musisz jednak pamiętać, że zapachy i odgłosy drukarni są całkiem inne niż te, jakie wydaje twój komputer z zainstalowanym TEX-em.)

Strona, którą teraz właśnie czytasz, TEX widzi jak pudełko. Jest to duże pudełko złożone z pionowej listy mniejszych pudełek, reprezentujących poszczególne wiersze tekstu. Każdy wiersz tekstu składa się z poziomej listy pudełek reprezentujących pojedyncze znaki. W bardziej skomplikowanych sytuacjach, wzorach matematycznych z potęgowaniem, w tablicach możesz mieć pudełka zawierające wewnątrz pudełka składające się z pudełek ... i tak aż do ostatniego poziomu.

Lecz każda taka skomplikowana sytuacja utworzona jest z pionowej lub poziomej listy pudełek połączonych w ten sam sposób. W rzeczywistości, gdy piszesz prosty tekst, zwykle nie myślisz o pudełkach, TEX ponosi odpowiedzialność za zebranie pudełek znaków w słowa, słowa w wiersze, wiersze w strony. O pudełkach przypominasz sobie jedynie, gdy chcesz zrobić coś odmiennego, np. wyśrodkować nagłówek.

Z punktu widzenia TEX-a tekst stanowi kolejkę żetonów (ang. *token*), a żetony przerobione zostają na kolejkę pudełek. Każdy żeton na wejściu jest instrukcją podstawową lub fragmentem instrukcji, np. żeton '$\mathtt{A}_{11}$' oznacza, że TEX wstawi pudełko znakowe litery `A` na końcu bieżącego pudełka poziomego, używając aktualnego fontu. Żeton `vskip` oznacza, że TEX zrobi odstęp pionowy w bieżącym pudełku pionowym o wielkości podanej w następnym żetonie.

Wysokość, szerokość lub głębokość pudełka może być ujemna. W takim wypadku jest to pudełko-cień, trudne do opisania. TEX nie unika wielkości ujemnych, dla niego jest to zwykła arytmetyka. Na przykład szerokość dwóch sąsiadujących pudełek jest sumą ich szerokości nawet gdy szerokość nie jest dodatnia. Typograf może zaprojektować ujemną szerokość znaku i w tym wypadku będzie on cofnięty w stosunku do innych. Tą cechę wykorzystuje się w językach w których tekst czyta się z prawa na lewo, ale tylko do pewnej określonej wielkości, od której TEX-owy algorytm dzielenia wierszy jest oparty na założeniu, że słowa nie mają wartości ujemnych.

TEX potrafi podnosić i opuszczać pojedyncze pudełka na liście poziomej; w ten sposób tworzy się wykładniki dolne i górne we wzorach matematycznych, także wysokość umieszczenia akcentów i kilka innych rzeczy. Na przykład poniżej pokazano sposób utworzenia pudełka zawierające TEX-owe logo wstawione do wewnętrznego rejestru `\box0`:

```
\setbox0=\hbox{T\kern-.1667em\lower.5ex%
                   \hbox{E}\kern-.125em X}
```

Tutaj `\kern-.1667em` oznacza wstaw pustą przestrzeń `-.1667 em` z bieżącego fontu, tzn. „cofnij troszeczkę”; a `\lower.5ex` oznacza, że pudełko `\hbox{E}` zostało opuszczone o połowę litery x bieżącego fontu, w stosunku do innych. Zamiast `\lower.5ex` można napisać także `\raise-.5ex`. Rozdziały 12 i 21 TB omawiają dokładnie jak konstruować pudełka, aby osiągnąć efekty specjalne.

TEX przedstawi zawartość dowolnego rejestru pudełka, jeżeli go o to poprosisz. Jeżeli na przykład napiszesz `\showbox0`, to po umieszczeniu w `\box0` TEX-owego logo jak powyżej, twój plik z rozszerzeniem `.log` będzie zawierał następujące czary mary:

```
\hbox(6.83331+2.15277)x18.6108
.\tenrm T
.\kern -1.66702
.\hbox(6.83331+0.0)x6.80557, shifted 2.15277
..\tenrm E
.\kern -1.25
.\tenrm X
```

Pierwsza linia oznacza, że `\box0` jest pudełkiem poziomym o wysokości, głębokości i szerokości odpowiednio 6.83331 pt, 2.15277 pt, and 18.6108 pt. Następne linie zaczynające się od '`.`' wskazują, że są one *wewnątrz* pudełka. Na pierwszym miejscu w omawianym pudełku znajduje się litera `T` fontu `\tenrm`, potem pojawia się klej `kern`, dalej pudełko poziome `hbox` zawierający literę `E` – to pudełko ma wysokość, głębokość i szerokość litery `E`, ale jest przesunięte w dół o 2.15277 pt (przez co zwiększa głębokość zewnętrznego pudełka).

Zadania

Zadanie 1. Dlaczego w zapisie '`..\tenrm E`' pojawiły się dwie kropki?

Odpowiedź: Ponieważ `E` jest wewnątrz dwóch pudełek.

Zadanie 2. Jak przy pomocy TEX-a odkryć w jaki sposób reprezentowana jest korekcja italikowa.

Odpowiedź: Dobrym pomysłem jest utworzenie pudełka i rozwinięcie jego zawartości:

```
\setbox0=\hbox{\sl g\/} \showbox0
```

widać, że `\/` jest nierozciągliwym klejem (kern). Dalsze eksperymenty ujawnią, że kern pojawia się nawet wówczas, gdy korekta italikowa jest równa zero.

Zadanie 3. Utwórzmy zniekształcone logo TEX-a, wyglądające następująco T E X:

```
\setbox1=\hbox{T\kern+.1667em\raise.5ex%
              \hbox{E}\kern+.125em X}
```

Zastanów się (bez uruchamiania TEX-a, co pojawi się w rozwinięciu powyższego pudełka (`\showbox1`).

Odpowiedź: Wysokość, głębokość i szerokość zewnętrznego pudełka jest zależna od elementów jakie zawiera, więc rezultatem jest:

```
\hbox(8.98608+0.0)x24.44484
.\tenrm T
.\kern 1.66702
.\hbox(6.83331+0.0)x6.80557, shifted -2.15277
..\tenrm E
.\kern 1.25
.\tenrm X
```

Najprawdopodobniej przewidziałeś wysokość pudełka 8.9861 `:-)`; pamiętajmy, że wewnętrznie TEX liczy w jednostkach sp (ang. *scaled point*), więc zaokrągla do piątego miejsca po przecinku.

Zadanie 4. Zdefiniuj makro `\demobox`, służące do pisania na przykład takich jak ten artykułów, aby po instrukcji `\demobox{Składamy TeX-em.}` otrzymać wynik .

Odpowiedź:

```
\def \dolist
    {\afterassignment\dodolist%
     \let\next= }
\def \dodolist
    {\ifx\next\endlist
     \let\next\relax
     \else \\\let\next\dolist \fi
     \next}
\def \endlist {\endlist}
\def \hidehrule #1#2%
    {\kern-#1%
     \hrule height#1 depth#2 \kern-#2 }
\def \hidevrule #1#2%
    {\kern-#1{\dimen0=#1\advance\dimen0 by#2%
     \vrule width\dimen0}\kern-#2 }
\def \makeblankbox #1#2%
    {\hbox{\lower\dp0\vbox{\hidehrule{#1}{#2}%
    \kern-#1 %
    \hbox to \wd0{\hidevrule{#1}{#2}%
      \raise\ht0\vbox to #1{}% wysokość
      \lower\dp0\vtop to #1{}% głębokość
      \hfil\hidevrule{#2}{#1}}%
    \kern-#1\hidehrule{#2}{#1}}}}
\def \maketypebox {\makeblankbox{0pt}{1pt}}
\def \makelightbox {\makeblankbox{.2pt}{.2pt}}
\def \\{\expandafter\if\space\next\
 \else \setbox0=\hbox{\next}\maketypebox\fi}
\def \demobox #1%
    {\setbox0=\hbox{\dolist#1\endlist}%
     \copy0\kern-\wd0\makelightbox}
```

Zadanie 5. Zdefiniuj makro `\frac`, które po wywołaniu `\frac1/2` TEX złoży następujący ułamek „$^1/_2$".

Odpowiedź:

```
\def \frac #1/#2%
    {\leavevmode\kern.1em
     \raise.5ex\hbox{\the\scriptfont0 #1}
     \kern-.1em/\kern-.15em
     \lower.25ex\hbox{\the\scriptfont0 #2}}
```

◇ Bogusław Lichoński
lichonski@wp.pl