

Jak szybko zacząć pisać w LaTeX?

ver 0.1
(2003-06-21)

Paweł Mazur (pmazur@pwr.wroc.pl)
<http://www.ci.pwr.wroc.pl/~pmazur/LaTeX>

Spis treści:

1. Wstęp.....	1
2. Wybór i konfiguracja środowiska	2
2.1. MiKTeX	2
2.2. EditPlus	2
2.3. SciTE.....	3
2.4. Yapp.....	4
3. Pierwszy dokument	4
4. Polskie znaczniki.....	5
5. Uwagi wstępne	8
6. Części, rozdziały, podrozdziały i spis treści	9
7. Rysunki i spis rysunków	10
8. Proste tabele i spis tabel	11
9. Tworzenie bibliografii.....	14
10. Pisanie długich dokumentów	15
11. PDF	15
12. Co dalej?.....	16
13. FAQ.....	16
14. Literatura	18

1. Wstęp

Dlaczego zdecydowałem się napisać ten tekst? Dlatego, że w momencie, kiedy podjąłem decyzję o przesiadce z MS Word na LaTeX'a skazałem się na długie szukanie materiałów po sieci, czytanie, próbowanie i zaczynanie od nowa. Postanowiłem więc spisać to, czego się nauczyłem i ułatwić innym start. Ten tekst przeznaczony jest dla ludzi całkowicie początkujących i powinien pozwolić w miarę szybko stworzyć pierwszy, nie całkiem banalny dokument LaTeXowy. Później pozostaje czytanie o kolejnych pakietach, wgryzanie się w szczegóły i niuanse (materiału do przyswojenia jest całkiem sporo ☺), ale pomocy w Internecie jest naprawdę dużo. Jeśli chodzi o źródła pomocy to zdecydowanie polecam:

- stronę <http://www.gust.org.pl> oraz <http://www.ia.pw.edu.pl/~wujek/tex/index.html>
- grupy newsowe pl.comp.dtp.tex oraz pl.comp.dtp.tex.gust
- wykorzystanie Google'a do konkretnych problemów również okazuje się być dobrym rozwiązaniem – jest całkiem sporo stron opisujących zarówno konkretne problemy i systematycznie całego LaTeX'a.

Po wybraniu z sieci kilku ciekawych artykułów dla początkujących na temat LaTeX'a udostępniam ich kopię na swojej stronie WWW.

Reguły tworzenia dokumentów LaTeX są następujące:

- dokumenty LaTeXowe piszemy jako pliki tekstowe (nazwa.tex)
- wykonujemy kompilację (powstaje plik nazwa.dvi)
- oglądamy wynik przeglądarką plików DVI.

2. Wybór i konfiguracja środowiska

To, co tu opisuję, jest tylko opisem tego, z czego ja korzystam osobiście i co mi odpowiada...

2.1. MiKTeX

Opisując swoją przygodę z LaTeXem omawiam sytuację dotyczącą systemu MS Windows. Polecam środowisko MiKTeX dostępne do ściągnięcia za darmo ze strony <http://www.miktex.org/index.html>. Dlaczego właśnie MiKTeX a nie coś innego? 1. jest dostępny za darmo 2. jest dobry i rozbudowany 3. prosty w instalacji i użytkowaniu 4. popularny.

Po zainstalowaniu MiKTeXa (dokładny opis – zrzut kolejnych ekranów – dostępny jest wraz z wersją instalacyjną) warto od razu zainstalować dodatkowe pakiety. Umożliwia to MiKTeX Package Manager, który wyświetla listę dostępnych pakietów wraz z krótkim opisem i datą instalacji (jeżeli jest zainstalowany). Prawy przycisk myszy na nazwie pakietu umożliwia zainstalowanie go. Na początek polecam doinstalować **plfonts, mwcls, Polish hyphenation patterns, graphics, platex**.

MiKTeX Update Wizard służy do uaktualnienia oprogramowania. Na początku „instalator” zapyta się o źródło pobierania uaktualnień, wybieramy „Internet”. Następnie należy określić serwer, z którego będą ściągane pakiety. Polecam wybrać „Poland – <ftp://sunsite.icm.edu.pl/pub/CTAN/systems/win32/miktex/tm/packages>”, gdyż jest w miarę szybki. Następnie wybieramy, które elementy mają zostać zaktualizowane. Po ściągnięciu danych z sieci aktualizacja zostanie wykonana automatycznie.

Po zainstalowaniu MiKTeXa i napisaniu pierwszego dokumentu (o tym napiszę w dalszej części) należy wykonać kompilację: w linii poleceń wpisujemy **texify nazwa.tex**. W niektórych sytuacjach kompilację należy wykonać dwu-, trzykrotnie, gdyż w poszczególnych przebiegach tworzone są informacje pomocnicze, a dopiero za drugim bądź trzecim razem powstaje wersja ostateczna - poprawna.

Utworzony zostanie plik nazwa.dvi, którego możemy obejrzeć przy pomocy przeglądarki (DVI Viewer) dostępnej wraz z MiKTeXem: Yapp. Pomocny może okazać się też plik nazwa.log w którym jest zapis z przebiegu kompilacji, a więc także komunikaty o błędach.

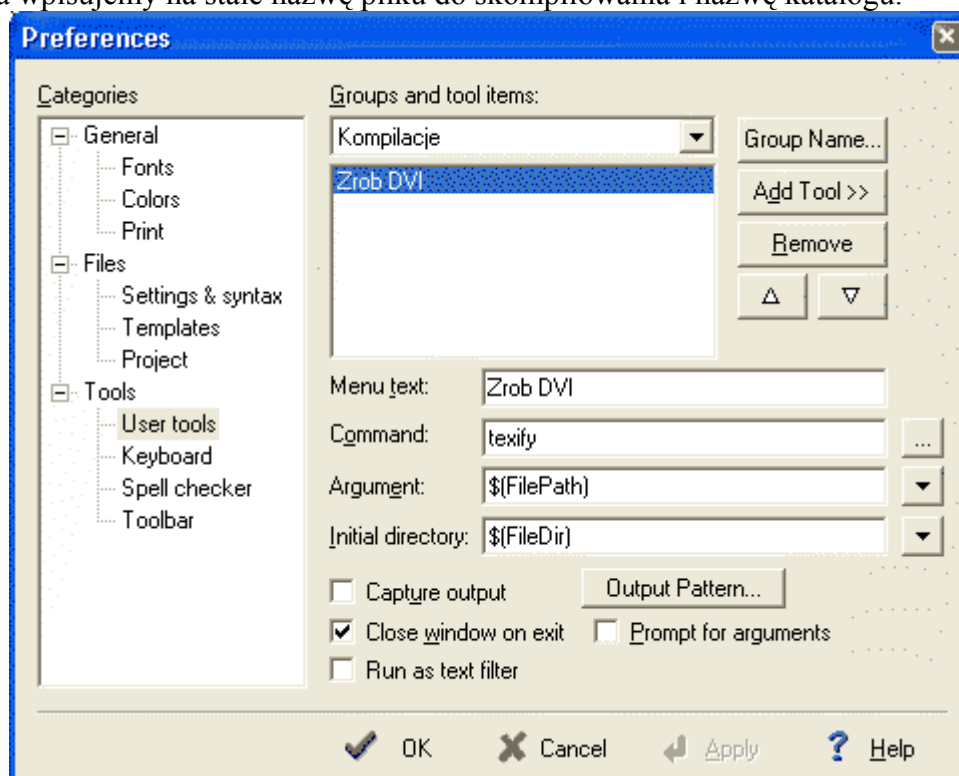
2.2. EditPlus

Dokumenty dla systemu LaTeX edytuje się w dowolnym edytorze tekstowym. Ja polecam dobrego na wszystko EditPlusa (<http://www.editplus.com>). Niestety licencja jest shareware (1 miesiąc za darmo), a koszt licencji to \$30, ale wg mnie warto, bo edytor jest bardzo dobry i przyjazny. Wersja instalacyjna zajmuje około 1 MB. Na stronie producenta można także pobrać pliki konfiguracyjne służące do kolorowania składni (stx) oraz wstawiające skróty (ctl) – wówczas nie musimy pamiętać skomplikowanych poleceń i makr. Zaletą jest możliwość napisania własnych plików konfiguracyjnych.

Inna zaleta EditPlusa to możliwość zdefiniowania własnych narzędzi. Jak to się przydaje? Otóż można zdefiniować wywołanie programu zewnętrznego. Można zatem zdefiniować skrót klawiszowy, który wykona kompilację naszego dokumentu. Wyręcza nas to od

posługiwania się linią poleceń, a dzięki temu znacznie przyspiesza pracę, gdy często wykonujemy kompilację (a ponieważ jesteśmy początkującymi użytkownikami, to zapewne co chwilę sprawdzamy jaki efekt przyniesie wpisanie kolejnego makra). Stworzenie takiego narzędzia jest bardzo proste, a zysk czasowy i wygoda duże. Załączam zrzut ekranu pokazujący konfigurację narzędzia. W grupie o nazwie „Kompilacje” jest narzędzie o nazwie „Zrob DVI”, które wykonuje plik DVI z aktualnie edytowanego pliku. Ważne jest ustawienie katalogu roboczego (Initial directory), gdyż program texify tworzy pliki pomocnicze. Jeżeli mamy utworzonych kilka grup narzędzi do skrótami klawiszowymi CTRL+SHIFT+cyfra wybieramy grupę aktywną, natomiast skrótem CTRL+cyfra wywołujemy odpowiednie narzędzie z aktywnej grupy narzędzi.

Dla wygodnej pracy z dokumentami TeX, które odwołują się do innych dokumentów (\include – opisuję to w dalszej części) warto zdefiniować sobie jeszcze jedno narzędzie, które będzie kompilować plik główny. Chodzi o sytuację, gdy piszemy długi tekst i np. każdy rozdział edytujemy w osobnym pliku. Jeżeli akurat pracujemy nad rozdziałem piątym, to niestety kompilację musimy i tak wykonać na pliku głównym. Wtedy w konfiguracji narzędzia wpisujemy na stałe nazwę pliku do skompilowania i nazwę katalogu.



2.3. SciTE

To inny edytor tekstowy, który warto polecić. Dostępny jest pod adresem <http://www.scintilla.org>. Jego zaletą jest mała objętość (400KB) oraz status open source, a także całkiem niezłe możliwości. Opiszę tu krótko dostępne funkcje:

- kolorowanie składni, autouzupełnianie słów kluczowych (póki co, dla EditPlusa jeszcze nie ma plików konfiguracyjnych do autouzupełniania)
- mechanizm skrótów (wygodniejszy niż w EditPlusie)
- skrót do kompilacji CTRL+F7 (wraz z okienkiem wynikowym)

- eksport kodu źródłowego do innych formatów niż TXT np. LaTeX. Jest to bardzo przydatna opcja, gdyż w mgnieniu oka pozwala wstawić do LaTeXa kod źródłowy naszego programu napisanego np. w C++.
- wsparcie dla wielu innych języków programowania (z takimi ciekawymi funkcjami jak zwijanie np. definicji klas i funkcji w C czy Javie)

Na końcu tego dokumentu znajduje się FAQ, w którym w odpowiedzi na pytanie „Jak nauczyć SciTE języka polskiego?” opisuję, jak rozwiązać sprawę polskich liter pod SciTE.

2.4. Yapp

Yapp jest przeglądarką plików DVI dostępną razem z dystrybucją MiKTeXa. Umożliwia przeglądanie dokumentu, powiększanie, drukowanie itd. Na początku problemem może się okazać drukowanie. Należy poustawiać odpowiednie drukarki w menu View – Options – Printer. Chodzi o rozdzielczości.

W innym miejscu View – Options – Display należy zaznaczyć pole „display colored text” co spowoduje, że dokument wyświetlany jest kolorowo a nie w odcieniach szarości. Ustawienie to dotyczy nie tylko kolorów użytych w tekście, ale także grafiki.

3. Pierwszy dokument

Jak już wspomniałem edytujemy najzwyczajniejszy dokument tekstowy. Na początku jest **preambuła**, w której określamy typ dokumentu, dołączamy pakiety oraz możemy stworzyć własne definicje pomocnicze. Przykładowa preambuła ma postać:

```
\documentclass[12pt,a4paper,twoside,fleqn]{report}
\usepackage[latin2]{inputenc}
% \usepackage[T1]{fontenc}
\usepackage[OT4,plmath]{polski}
\usepackage{indentfirst}
\usepackage{graphicx}
\pagestyle{uheadings}

% DEFINICJE

\newtheorem{pyt}{Pytanie}
\newtheorem{lema}{Lema}
\newtheorem{tw}{Twierdzenie}
```

Od znaku % do końca linii jest komentarz.

W pierwszej linijce „mówimy” do jakiej klasy należy dokument.

```
\documentclass[opcje]{klasa}
```

Przykładowe klasy to: article, report, book, slides, letter. Jeżeli zainstalujemy klasy mwcls (zob. punkt 2), to mamy do dyspozycji także klasy: mwart, mwrep, mwbk.

Niektóre opcje klas to: podstawowy rozmiar czcionki (10pt, 11pt, 12pt), rozmiar papieru, sposób eksponowania wzorów matematycznych, liczba kolumn itd. Dokładniejszy opis znajduje się w [88minut] na stronach 8-9.

Poleceniem **\usepackage** dołącza się pakiety, które poszerzają nasze możliwości.

Standardowe style stron to plain, heading oraz empty. Klasy mwcls dostarczają nam nowych stylów: outer, myheadings, uheadings oraz myuheadings. Wybierając określony styl strony decydujemy o takich parametrach jak: zawartość nagłówka i stopki strony, miejsce umieszczenia numeru strony, oddzielenie nagłówka i stopki poziomą linią od części głównej strony. Szczegółowy opis znajduje się w [88minut] strona 11 oraz [MWCLS] także strona 11. Następnie widzimy jakieś trzy definicje, które będą przydatne w dalszej części dokumentu.

Po preambule jest część główna dokumentu:

```
\begin{dokument}
  Tu jest treść dokumentu.
\end{dokument}
```

4. Polskie znaczki

Jak już można było zauważyć w punkcie trzecim, aby móc korzystać z polskich znaczków, należy dodać odpowiednie pakiety. Zachęcam do zapoznania się z opisem na temat PL znaczków znajdującym się w dokumencie [88minut] strony 21-26. Ja tutaj opiszę to nieco krócej.

- a) najpierw ustalamy w jakim kodowaniu będziemy pisać dokument. Jeżeli będzie to Windows (CP-1250) to wstawiamy linijkę

```
\usepackage[cp1250]{inputenc}
```

- a jeżeli ISO-8859-2 to wpisujemy

```
\usepackage[latin2]{inputenc}
```

Dzięki dołączeniu pakietu inputenc polskie litery możemy wpisywać bezpośrednio z klawiatury, a nie musimy stosować zapisów prefiksowych lub innych pakietu Babel opisanych w [88minut]. Należy jednak pamiętać, że jeżeli decydujemy się na standard ISO, to znaki, jakie wpisujemy do naszego dokumentu, rzeczywiście muszą być w ISO. Można to osiągnąć dwoma sposobami:

- stosować programy konwertujące z CP-1250 na ISO, co może być mało wygodnym rozwiązaniem, ze względu na to, że przed każdą kompilacją trzeba wykonać konwersję.
- zainstalować polską klawiaturę ISO pod Windowsem. Opis, jak to zrobić, znajduje się w odpowiedzi na pytanie „Jak kodować dokumenty znakami ISO-8859-2?” znajdującą się w FAQ na końcu tego dokumentu.

- b) fonty EC

Pakiet fontenc definiuje układ znaków. Wstawiając linijkę `\usepackage[T1]{fontenc}` wskazujemy, że mają być użyte fonty EC, których zalety można znaleźć w [88minut] na stronie 22-23 oraz 24

- c) pakiet polski

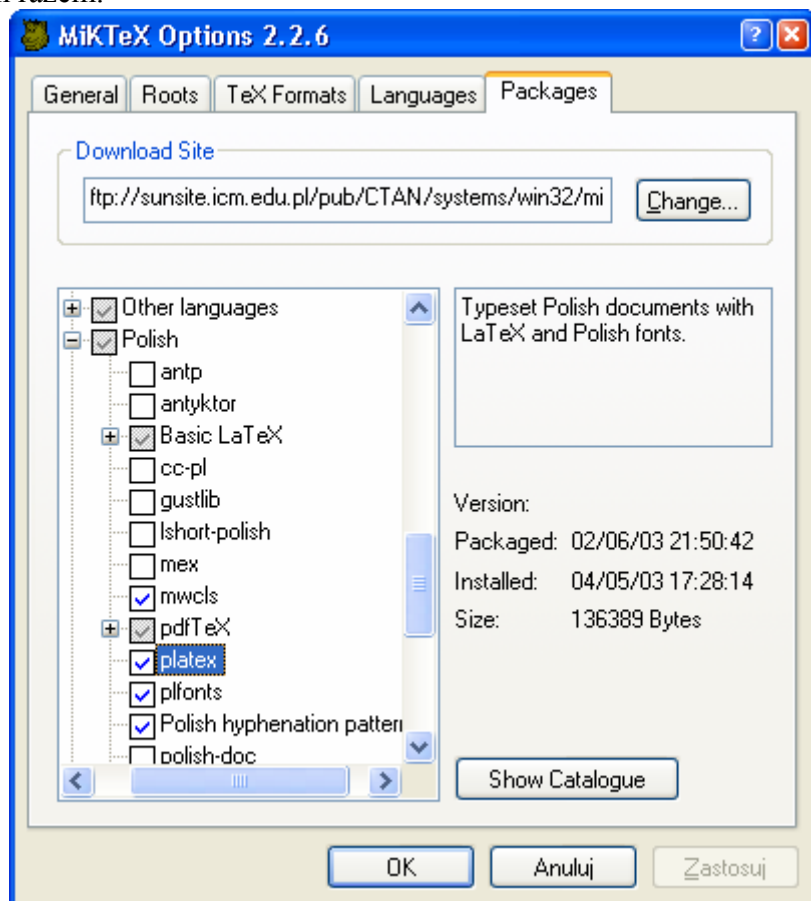
Pakiet polski został opracowany przez Mariusza Olko i Marcina Wolińskiego i stanowi część zestawu polonizacyjnego platex. Służy on do polonizacji dokumentu. Polonizacja polega m.in. na tym, że elementy generowane automatycznie są w języku polskim. Przykładem takich elementów są: „spis treści”, „bibliografia”, podpisy pod rysunkami i tabelami itd. Pakiet polski obejmuje także polskie nazwy i symbole funkcji trygonometrycznych, poprawne dzielenie wyrazów złożonych.

Dołączenie pakietu polskiego wygląda następująco:

```
\usepackage[opcje]{polski}
```

Dostępne opcje to: OT4, T1, OT1, plmath, nomathsymbols, MeX, które dobrze opisane są w [88minut] na stronach 25-26. Krótko tylko wspomnę, że OT4 oznacza składanie fontami PL, T1 to wybór fontów EC, OT1 to fonty CM, plmath zmienia definicję znaków mniejszy-równy i większy-równy (muszą być zainstalowane fonty PL), MeX to pełna polonizacja (OT4+plmath).

Jednak zanim będziemy mogli dołączyć pakiet polski do naszego dokumentu musimy go zainstalować. W MiKTeX nie stanowi to problemu: uruchamiamy MiKTeX Options, wchodzimy do zakładki Packages, rozwijamy znakiem plusa Languages, Polish. Następnie zaznaczamy platex do zainstalowania i klikamy Zastosuj. Pakiet zostanie ściągnięty z serwera podanego w polu „Download site” – polecam serwer `ftp://sunsite.icm.edu.pl/pub/CTAN/systems/win32/miktex/tm/packages/`. Na szczęście adres serwera wystarczy wpisać raz, a w przyszłości w przypadku kolejnych instalacji nowych pakietów MiKTeX pamięta, z jakiego serwera korzystaliśmy poprzednim razem.



Innym sposobem zainstalowania platexa jest uruchomienie MiKTeX Package Manager i po odnalezieniu pakietu platex, wykonujemy prawoklik na nim i wybieramy install. Lista dostępnych pakietów zawiera pakiety o nazwach zaczynających się zarówno dużą jak i małą literą – należy pamiętać, że ich sortowanie odbywa się niezależnie. W celu szybkiego sprawdzenia, jakie pakiety mamy obecnie zainstalowane w systemie MiKTeX należy posortować listę pakietów po dacie instalacji.

d) dzielenie wyrazów

Ostatnim krokiem jest określenie sposobu dzielenia wyrazów. Niestety informacje, jakie znalazłem na temat w sieci są sprzeczne i niektóre z opisanych metod nie działają. Zakładam, że korzystamy z MikTeXa. Uruchamiamy programik dostarczony wraz z MiKTeXem o nazwie MiKTeX Options, wchodzimy do zakładki Languages i zaznaczamy interesujące nas języki: np. english, german, polish oraz nohyphenation. Następnie wchodzimy do zakładki General i klikamy „Update Now”. W ten sposób tworzymy plik opisujący formatowanie. Po drodze może wyskoczyć kilka błędów, ale nimi nie należy się przejmować.

A teraz mały komentarz do stron 24-25 z [88minut]. Nie modyfikujemy pliku `language.dat`, bo to nic nie daje, ale korzystamy z zakładki Languages w MiKTeX Options. Po wykonaniu polecenia „`initex latex.ltx`” otrzymujemy plik `latex.fmt`, ale nie bardzo wiadomo gdzie go wkopiować, bo... stara wersja tego pliku w katalogach MikTeX'a nie istnieje (co, więcej nawet po wykonaniu „Update Now” taki plik nie istnieje). Okazuje się, że powstały plik `latex.fmt` należy skopiować do katalogu, w którym mamy nasz plik `nazwa.tex`. Jest to baaardzo niezadowolające rozwiązanie, gdyż, jeżeli piszemy wiele prac, artykułów i trzymamy je w różnych katalogach na dysku, to w każdym z nich musi być plik `latex.fmt`. Nie działa nawet skopiowanie tego pliku do jakiegoś arbitralnego katalogu i dopisanie go w zmiennej systemowej PATH. Nie działa także metoda skopiowania tego pliku do katalogu `/miktex/miktex/fmt` (taką radę znalazłem w Internecie). Zatem widać, że najlepszym rozwiązaniem jest kliknąć przycisk „Update Now” i po problemie.



Podsumowanie:

Zalecam korzystać z pakietu `polski`. Nie trzeba wówczas dołączać pakietu `fontenc`. Należy także pamiętać, że nowsze ustawienia nadpisują stare, tzn. jeżeli najpierw umieścimy linijkę `\usepackage[T1]{fontenc}` a następnie `\usepackage[OT4,plmath]{polski}` to w dokumencie obowiązuje ustawienie OT4, a nie T1.

Jeżeli wiemy, że nie będziemy mieli do czynienia z systemem innym niż Windows korzystajmy z kodowanie CP1250, jeżeli zamierzamy w trakcie pisania dokumentu korzystać z różnych systemów operacyjnych należy używać kodowania Latin2 (czyli ISO-8859-2).

Ostatecznie w preambule powinno się znaleźć (wg mojego zalecenia):

```
\usepackage[cp1250]{inputenc}
\usepackage[OT4,plmath]{polski}
```

lub

```
\usepackage[latin2]{inputenc}
\usepackage[OT4,plmath]{polski}
```

5. Uwagi wstępne

a)

W LaTeXu są znaki specjalne:

`~ # $ % ^ & _ { } \`

Ich wprowadzenie (poza `\`) do tekstu jest możliwe poprzez zastosowanie przed nimi znaku `\`.

czyli `\# \ $ \% \^ \& _ \{ \}`

Znak `\` wprowadza się instrukcją `\backslash`

b)

Polskie zwyczaje różnią się nieco od tych panujących w angielskojęzycznych tekstach, jeśli chodzi o wcięcia akapitowe w pierwszym akapicie po tytule: u nas się takie wcięcie robi, a „u nich” nie. Dlatego należy w preambule dołączyć pakiet `indentfirst`:

```
\usepackage{indentfirst}
```

c)

LaTeX sam decyduje, w którym miejscu złamać wiersz, sam też dobiera odstępy między wyrazami i, jak już wspomniałem, sam potrafi dokonać przeniesienia wyrazu. Jeżeli ręcznie chcemy dokonać złamania wiersza, robimy to za pomocą instrukcji `\\` lub `\newline`. Jeżeli dodatkowo chcemy _zakazać_ złamanie strony to używamy instrukcji `*`.

Nową stronę wstawiamy instrukcją `\newpage`.

Instrukcja:

`\linebreak[s]` oznacza zachętę do złamania wiersza

`\nolinebreak[s]` oznacza niezgodę na złamanie wiersza

`\pagebreak[s]` oznacza zachętę do złamania strony

`\nopagebreak[s]` oznacza niezgodę na złamanie strony

We wszystkich przypadkach `s` jest argumentem opcjonalnym o dopuszczalnej wartości z przedziału 0-4 oznaczającym stopień (siłę) tej zachęty/niezgody. W przypadku pominięcia tego argumentu brana jest wartość domyślną równa 4 (czyli bezwarunkowy nakaz/zakaz).

d)

Czasami podczas kompilacji dostajemy ostrzeżenie typu:

```
- Underfull \hbox (badness 1635) in paragraph at lines 6-7
```

Komunikat ten oznacza, że przy wyrównywaniu tekstu do prawego marginesu powstały zbyt duże odstępy między wyrazami. Liczba występująca po słowie „badness” określa jak bardzo złe są powstałe odstępy: maksymalna wartość to 10000.

```
- Underfull \vbox (badness 1270) has occurred while \output is active [9]
```

Jest to komunikat o podobnym znaczeniu jak ten pierwszy, z tym, że odnosi się do zbyt dużej ilości pustego miejsca pozostawianego u dołu strony (np. po złamaniu strony poleceniem

\pagebreak możemy zostawić pół strony pustej). Także tutaj liczba po słowie badness to ocena niepoprawności składu, o maksymalnej wartości równej 10000 (czyli wówczas wg LaTeXa jest najbardziej beznadziejnie). Liczba podana w [] oznacza numer strony, na której wystąpił problem

- Overfull \hbox (3.17558pt too wide) in paragraph at lines 15–16

Komunikat ten oznacza, że z powodu problemów z poprawnym złożeniem linii, tekst wystaje na prawy margines o 3,17558 punktów drukarskich (1pt = 0,376mm). Jeżeli chcemy dokładnie, łatwo i szybko zlokalizować miejsce wystąpienie tego błędu należy dopisać opcje draft w poleceniu \documentclass

```
\documentclass[12pt,a4paper,twoside,fleqn,draft]{mwrep}
```

a wówczas na prawym marginesie pojawi się czarny kwadracik oznaczający miejsce wystąpienia błędu. Należy jednak pamiętać, że draft oznacza szkic i w takim trybie nie zobaczymy rysunków, tylko ich ramki (piszę na wszelki wypadek, żeby się ktoś nie nastraszył).

Zazwyczaj nie zwraca się uwagi na powyższe błędy, chyba, że dokument wygląda na tyle źle, że nie jest przez nas do przyjęcia.

e)

Znak tyldy ~ w kodzie źródłowym oznacza niełamliwą spację. Używamy tego, gdy nie dopuszczamy złamania wiersza wewnątrz jakiegoś wyrażenia, np. „godz.~17.00”

Zakaz przenoszenia wyrazów pomiędzy wierszami oznaczamy umieszczając tekst wewnątrz polecenia \mbox{tekst}. Można to wykorzystać np. w sytuacji, gdy nasz akapit został zakończony słowem przeniesionym do nowego wiersza. Wygląda to dość brzydko, gdyż w ostatniej linii jest np. tylko jedna sylaba:

W dalszej części pracy opisana powyżej struktura nazywana będzie czwór-
ką.

Wówczas najlepiej jest zapisać tekst źródłowy „W dalszej części pracy opisana powyżej struktura nazywana będzie \mbox{czwórką}.”, a w rezultacie odległości między słowami się lekko zmniejszą i całe słowo zostanie wstawione w tym samym wierszu:

W dalszej części pracy opisana powyżej struktura nazywana będzie czwórką.

6. Części, rozdziały, podrozdziały i spis treści

Najczęściej nasz dokument składa się z jakichś części, rozdziałów itd. W standardowych klasach LaTeXa mamy następujące instrukcje dla klasy article:

```
\section{...}           \paragraph{...}
\subsection{...}        \subparagraph{...}
\subsubsection{...}     \appendix{...}
```

Dla klas book i report dostępne są dodatkowo(!) instrukcje

```
\part{...}             oraz          \chapter{...}
```

W miejsce kropek wpisuje się tytuł danej jednostki.

Aby utworzyć spis treści, należy wstawić instrukcję
`\tableofcontents`

Aby LaTeX wstawił poprawne numery stron w spisie treści **kompilację należy wykonać dwukrotnie**.

Jeżeli chcemy, aby jakiś tytuł nie został uwzględniony w numeracji ani w spisie treści należy przed {tytułem} wpisać znak *, np. `\section*{mój ukryty tytuł}`.

Istnieje również możliwość nadania innego tytułu dla spisu treści, a innego w treści dokumentu. Najczęściej robi się to, gdy tytuł rozdziału/sekcji jest bardzo długi (normy wymagają, by w spisie treści, nie zamieszczać długich tytułów). Stosuje się wówczas następującą składnię:

```
\section[tytuł do spisu treści]{tytuł wstawiany w treści dokumentu}
```

W przypadku korzystania z klas Marcina Wolińskiego (mwcls), mamy nieco więcej opcji:

- dwa opcjonalne tytuły

```
\section[param1][param2]{tytuł}
```

gdzie param1 to skrótowy tytuł, który pojawi się w paginie, a param2 zawiera tytuł, który pojawi się w spisie treści.

- wersja z * działa w ten sposób, że numer nie jest nadawany, ale tytuł jest umieszczany w spisie treści

- zablokowane jest dzielenie wyrazów w obrębie tytułów.

Szczegółowy opis klas mwcls znajduje się w [MWCLS].

7. Rysunki i spis rysunków

Opiszę tutaj, jak szybko wstawić rysunek. Wszelkie niuanse opisane są w [MYSZKA 1], [MYSZKA 2], [88minut] oraz na grupach newsowych.

Na początku musimy doinstalować pakiet graphicx i dołączyć go w preambule:

```
\usepackage{graphicx}
```

Formatem graficznym zalecanym dla LaTeXa jest EPS (w [MYSZKA 2] są wymienione zalety tego formatu).

Rysunek wstawiamy w następujący sposób:

```
\begin{figure}[!ht]
  \centering
  \includegraphics{images/kwadrat_przyklad.eps}
  \caption{Przykład kwadratu}
  \label{kwadrat_przyklad}
\end{figure}
```

Parametry ! h t mówią o umiejscowieniu rysunku w tekście. Składem zajmuje się LaTeX i to on decyduje jak najlepiej umieścić rysunek w tekście. Możemy jednak dać LaTeXowi pewne wskazówki:

h – dokładnie w miejscu wstawienia instrukcji (należy z tego korzystać w przypadku niedużych rysunków) (here)

t – na górze strony (top)

b – na dole strony (bottom)

p – na specjalnej stronie zawierającej rysunki i tabele (page)

! – ignorując parametry dotyczące umieszczania rysunków (np. maksymalna liczba rysunków na stronie).

Parametry mają priorytet idąc od lewej strony. Czyli `[\!ht]` oznacza, że jeżeli można to umieścić rysunek dokładnie w miejscu gdzie pojawiła się instrukcja. Ale jeżeli nie będzie to możliwe (np. koniec strony), to wstaw rysunek na górze następnej strony.

W przypadku nie podania parametrów umiejscowienia rysunku, domyślne ustawienia to `[tbp]`.

Może się jednak zdarzyć, że chcemy wstawić rysunki, które nie są w formacie EPS, a nie mamy czasu i/lub ochoty by je przerabiać. Można wstawić rysunki w innym formacie niż EPS, ale tak naprawdę, w czasie kompilacji wykonana zostanie konwersja. Wstawienie rysunku do dokumentu wygląda identycznie jak w przypadku pliku EPS – po prostu podajemy odpowiednią nazwę pliku. Dodatkowo musimy jednak utworzyć plik tekstowy o nazwie identycznej jak nazwa pliku z rysunkiem i nadajemy mu rozszerzenie `bb`. Wewnątrz pliku wpisujemy:

```
%%BoundingBox: 0 0 x y
```

gdzie `x` to szerokość rysunku w pikselach, a `y` to wysokość.

Odwołanie do rysunku w tekście wykonujemy wstawiając

```
\ref{label}
```

gdzie `label` to nadana rysunkowi etykieta.

Spis rysunków wykonuje się poleceniem `\listoffigures`. W spisie umieszczony zostanie tytuł rysunku podany w `\caption`.

Jeśli chodzi o program graficzny to polecam Paint Shop Pro. Oferuje naprawdę wiele możliwości, operuje na warstwach, obsługuje wiele formatów graficznych, w tym także EPS. Wersja shareware pozwala na 60 dni użytkowania za darmo.

8. Proste tabele i spis tabel

Tabele w LaTeX to tak szeroki temat, że aby go całkowicie opisać powstałaby niezłych rozmiarów broszurka. Świadczyć o tym może np. liczba pakietów pomocnych w tworzeniu tabel. Ich krótki opis możemy znaleźć pod adresem http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/macro_t.html

Pokażę, jak zrobić najprostszą tabelę

```
\begin{table}[\!ht]
  \begin{center}
    \caption{Oto nasza pierwsza tabelka}
    \label{tabela_przyklad}
    tu są definicje tabel
  \end{center}
\end{table}
```

Widać tu dużo analogii i podobieństw do środowiska `figure`, więc nie będę tu ponownie opisywać wspólnych rzeczy. Omówię za to teraz fragment „tu są definicje tabel”. W tym miejscu wstawiamy środowisko `tabular`:

```
\begin{tabular}[pozycja]{kolumny}
  treść wierszy
\end{tabular}
```

pozycja określa pozycję w pionie: domyślnie jest środek, dostępne argumenty to t (top) i b (bottom)

kolumny definiują liczbę kolumn i ich wyrównanie w poziomie. Robi się to wstawiając litery l, c lub r. Każda litera oznacza jedną kolumnę z tekstem wyrównanym do lewej strony, do środka lub do prawej strony. Litery można oddzielić znakiem | który wskazuje, że komórki mają być oddzielone pionową linią. Można też wstawić p{szerokość} zamiast wymienionych literek – wówczas zawartość komórki wstawiona jest jako prostokąt z wyjustowanym tekstem.

treść wierszy ma postać wart 1 & wart 2 & ... & wart n \\

Linie poziomą wstawiamy instrukcją \hline

Komórki rozciągające się na kilka kolumn wstawiamy instrukcją

\multicolumn{liczba_kolumn}{formatowanie}{tekst}

gdzie formatowanie oznacza położenie w poziomie (l, c, r)

Spis tabel wykonuje się poleceniem \listoftables.

A teraz przykład z życia wzięty: kod, wynik i omówienie.

```
\begin{table}[htp]
\caption{Przykładowa tabela}
\label{tabela_przyklad}
\begin{center}

\begin{tabular}[t]{|r|r|r|}
\multicolumn{1}{c}{n} & \multicolumn{1}{c}{C\raisebox{-0.5ex}{n}} & 
\multicolumn{1}{c}{Czas [sek]} \\ \hline
1 & 4 & 0.00 \\ \hline
2 & 12 & 0.00 \\ \hline
3 & 36 & 0.00 \\ \hline
4 & 100 & 0.00 \\ \hline
5 & 284 & 0.00 \\ \hline
6 & 780 & 0.00 \\ \hline
7 & 2172 & 0.00 \\ \hline
8 & 5916 & 0.00 \\ \hline
9 & 16268 & 0.00 \\ \hline
10 & 44100 & 0.00 \\ \hline
11 & 120292 & 0.01 \\ \hline
12 & 324932 & 0.01 \\ \hline
13 & 881500 & 0.04 \\ \hline
14 & 2374444 & 0.10 \\ \hline
\multicolumn{1}{c}{} & \multicolumn{1}{c}{\parbox{1.23in}{\ }} & 
\multicolumn{1}{c}{} \\ \hline
\end{tabular}

\hspace{0.1in}

\begin{tabular}[t]{|r|r|r|}
\multicolumn{1}{c}{n} & \multicolumn{1}{c}{C\raisebox{-0.5ex}{n}} & 
\multicolumn{1}{c}{Czas [sek]} \\ \hline
15 & 6416596 & 0.27 \\ \hline
16 & 17245332 & 0.72 \\ \hline
17 & 46466676 & 1.93 \\ \hline
18 & 124658732 & 5.19 \\ \hline
19 & 335116620 & 13.93 \\ \hline
20 & 897697164 & 37.43 \\ \hline
21 & 2408806028 & 100.44 \\ \hline
22 & 6444560484 & 269.36 \\ \hline
\end{tabular}

\end{center}
\end{table}
```

```

23 & 17266613812 & 721.57 \\ \hline
24 & 46146397316 & 1956.87 \\ \hline
25 & 123481354908 & 5172.83 \\ \hline
26 & 329712786220 & 13822.45 \\ \hline
27 & 881317491628 & 37172.49 \\ \hline
\multicolumn{1}{c}{} & \multicolumn{1}{c}{\parbox{1.23in}{\ }} & \\
\multicolumn{1}{c}{} & \\
\end{tabular}

\end{center}
\end{table}

```

Wynik:

Tabela 2.1. Przykładowa tabela

n	C _n	Czas [sek]	n	C _n	Czas [sek]
1	4	0.00	15	6416596	0.27
2	12	0.00	16	17245332	0.72
3	36	0.00	17	46466676	1.93
4	100	0.00	18	124658732	5.19
5	284	0.00	19	335116620	13.93
6	780	0.00	20	897697164	37.43
7	2172	0.00	21	2408806028	100.44
8	5916	0.00	22	6444560484	269.36
9	16268	0.00	23	17266613812	721.57
10	44100	0.00	24	46146397316	1956.87
11	120292	0.01	25	123481354908	5172.83
12	324932	0.01	26	329712786220	13822.45
13	881500	0.04	27	881317491628	37172.49
14	2374444	0.10			

Omówienie:

Po pierwsze widzimy, że mamy dwa środowiska `tabular` oddzielone odstępem poziomym `\hspace{0.1in}`. Konstrukcja obydwóch `tabular` jest identyczna.

```
\begin{tabular}[t]{|r|r|r|}
```

Określamy, że będą trzy kolumny, wszystkie wyrównane do prawej, z pionowymi liniami po bokach – chcemy, żeby tabelka była typu siatka.

```

\multicolumn{1}{c}{n} & \multicolumn{1}{c}{C\raisebox{-0.5ex}{n}} & \\
\multicolumn{1}{c}{Czas [sek]} & \\
\hline

```

To jest nagłówek. Ponieważ chcemy, żeby ten wiersz zachowywał się inaczej niż standardowa komórka tabeli (wyśrodkowanie zamiast wyrównanie do prawej), stosujemy `multicolumn`, co pozwala nam na zmianę wyrównania.

```
1 & 4 & 0.00 \\ \hline
```

A to są już zwykłe wiersze tabeli...

```

\multicolumn{1}{c}{} & \multicolumn{1}{c}{\parbox{1.23in}{\ }} & \\
\multicolumn{1}{c}{} & \\

```

To jest pewien trick zastosowany po to, żeby określić szerokość kolumny pozostawiając jednocześnie wyrównanie tekstu do prawej strony w komórkach z danymi (chodzi o środkową kolumnę). Nie wiem czy można to osiągnąć jakoś inaczej (nie korzystając z dodatkowych pakietów) – mi się nie udało, więc wymyśliłem taki sposób.

9. Tworzenie bibliografii

Tworzenie bibliografii w LaTeX jest bardzo wygodne.

```
\begin{thebibliography}{szerokosc}
  \bibitem{etykieta1}opis1
  \bibitem{etykieta2}opis2
\end{thebibliography}
```

Posługując się etykietą możemy łatwo odwołać się do jakiejś pozycji w bibliografii. W miejscu, w którym takie odwołanie ma być zrobione wpisujemy

```
\cite{etykieta}
```

Ułatwia nam to pisanie dokumentu w ten sposób, że nadawanie etykiet jest niezależne od symbolu (liczby) pojawiającej się w odwołaniu. Jeżeli po jakimś czasie wzbogacimy ułożoną alfabetycznie bibliografię, nie musimy zmieniać numerów we wszystkich odwołaniach w dokumencie.

Domyślnym formatem tworzenia odwołań jest zapis [x], gdzie x to kolejne liczby począwszy od 1. Numeracja kolejnych pozycji bibliografii wykonywana jest automatycznie.

szerokosc jest dowolnym napisem, którego liczba znaków określa maksymalną liczbę znaków w odwołaniach. Jest to potrzebne do prawidłowego wyświetlenia bibliografii. Jeżeli wpisujemy `\begin{thebibliography}{1}`, a nasza bibliografia ma ponad 9 pozycji, wówczas będzie ona błędnie wyrównana (przykład po lewej stronie)

[1] pozycja nr 1	[1] pozycja nr 1
[2] pozycja nr 2	[2] pozycja nr 2
[3] pozycja nr 3	[3] pozycja nr 3
[4] pozycja nr 4	[4] pozycja nr 4
[5] pozycja nr 5	[5] pozycja nr 5
[6] pozycja nr 6	[6] pozycja nr 6
[7] pozycja nr 7	[7] pozycja nr 7
[8] pozycja nr 8	[8] pozycja nr 8
[9] pozycja nr 9	[9] pozycja nr 9
[10] pozycja nr 10	[10] pozycja nr 10
[11] pozycja nr 11	[11] pozycja nr 11
[12] pozycja nr 12	[12] pozycja nr 12

natomiast jeżeli wpisujemy `\begin{thebibliography}{11}` wyrównanie będzie poprawne (przykład po prawej stronie)

Odwołania nie muszą być robione automatycznie. Własne odwołania podajemy w nawiasach [] przed etykietą:

```
\begin{thebibliography}{szerokosc}
  \bibitem[nazw rok]{etykieta1}opis1
  \bibitem[nazw rok]{etykieta2}opis2
\end{thebibliography}
```

Jeżeli w niektórych pozycjach podamy własne napisy odwołań, a w niektórych nie podamy odwołań w [], to dla tych bez [] LaTeX stworzy sam o postaci [liczba], a numeracja będzie

opuszczać pozycje z nazwanymi odwołaniami, czyli zapis po lewej da wynik taki, jak po prawej stronie poniżej:

<code>\begin{thebibliography}{123}</code>	<code>[cos]</code> pozycja 1
<code>\bibitem[cos]{etykieta1}pozycja 1</code>	<code>[1]</code> pozycja 2
<code>\bibitem{etykieta2}pozycja 2</code>	<code>[2]</code> pozycja 3
<code>\bibitem{etykieta3}pozycja 3</code>	<code>[kot]</code> pozycja 4
<code>\bibitem[kot]{etykieta4}pozycja 4</code>	<code>[3]</code> pozycja 5
<code>\bibitem{etykieta5}pozycja 5</code>	
<code>\end{thebibliography}</code>	

Aby bibliografia oraz odwołania do niej były wykonane prawidłowo, należy wykonać co najmniej **trzy kompilacje dokumentu**.

10. Pisanie długich dokumentów

W przypadku pisania długich dokumentów ułatwieniem może być podzielenie go na kilka plików. W LaTeX istnieje bardzo prosty sposób dołączenia plików. Służy do tego instrukcja

`\include{plik}`

gdzie `plik` jest nazwą pliku (sama nazwa, bez rozszerzenia) .tex dołączanego do dokumentu. Instrukcja ta działa na zasadzie „wkopiowania” zawartości pliku. W miejscu wklejenia pliku zostanie utworzona nowa strona w dokumencie. Jeżeli nie chcemy, by nowa strona była tworzona, należy użyć instrukcji

`\input{plik}`

Dodatkową pomocną instrukcją jest

`\includeonly{lista plików}`

którą umieszcza się w preambule. Lista plików określa, które instrukcje `\include` mają zostać wykonane (nie dotyczy to instrukcji `\input`, które zawsze są wykonywane). Wykorzystuje się to np. wtedy, jeżeli chcemy przekazać komuś plik DVI obejmujący tylko fragment naszego dokumentu, a nie chcemy się namęczyć wykommentowując instrukcje `\include` – zmieniamy tylko listę plików w instrukcji `\includeonly`. Lista ta ma postać `{plik1,plik2,plik3}` – bez spacji i innych odstępów.

Patrząc się z praktycznego punktu widzenia, w sytuacji, gdy pracujemy z kilkoma plikami jednocześnie, wygodniejszym edytorem jest EditPlus w porównaniu z Scite.

11. PDF

Z MiKTeX to nic trudnego! Wraz z dystrybucją dostępne jest narzędzie pdfLaTeX, które z pliku `nazwa.tex` tworzy plik `nazwa.pdf`. W sieci znajduje się wiele materiałów dotyczących tworzenia PDF w LaTeX – najobszerniejsze opracowanie to pozycja [PDF 2]. W wyniku użycia `pdflatex` dostarczanego razem z MiKTeXem dostajemy plik PDF o naprawdę wysokiej jakości, wraz z zachowaniem kolorów (także w rysunkach). Możemy więc zdefiniować sobie nowe narzędzie w EditPlusie i teraz naciskając np. CTRL+1 robimy DVI, a CTRL+2 robimy PDF.

12. Co dalej?

Na początek polecałbym zapoznanie się z następującymi „pomocami naukowymi” (w takiej właśnie kolejności):

- „Nie za krótkie wprowadzenie do systemu LaTeX2e” (tłumaczenie i opracowanie wersji polskiej: Janusz Gołdasz, Ryszard Kubiak, Tomasz Przechlewski) dostępne pod adresem <http://www.gust.org.pl/doc.html>
- „bardzo krótkie FAQ grup pl.comp.dtp.tex oraz pl.comp.dtp.tex.gust” zebrane przez Grzegorza Sapijasko dostępne pod adresem http://www.sapijasko.net/pl_tex_faq.pdf
- Woliński M., „MWCLS – Moje własne klasy dokumentów dla LaTeX2e. Podręcznik użytkownika”
- Myszka W., „Włączanie grafik do tekstów w LATEX2e” dostępne pod adresem <http://www.immt.pwr.wroc.pl/~myszka/grafika/grafika.pdf>
- <http://www.giss.nasa.gov/latex/ltx-2.html>

Jeżeli kogoś interesuje wprowadzenia kolorów do swojego dokumentu, to polecam zapoznać się z artykułem Macewicz W., „LaTeX na kolorowo”, Zeszyt 19 GUST

Następnie, już jako uzupełnienie, polecam przeczytać:

- Peter Flynn „A beginner’s introduction to typesetting with LaTeX”
- Książka Wojciecha Myszk i Ewarysta Rafajłowicza, „LaTeX. Podręcznik użytkownika” ISBN 83-85190-60-0 Warszawa, wydaw. PLJ 1992
- <http://www.ia.pw.edu.pl/~wujek/tex/>
- http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/macro_t.html
- <http://www.fuw.edu.pl/~pliszka/hints/LaTeX.html>
- pozostałe pozycje literatury, których tutaj nie wymieniłem

Niektóre z dokumentów oraz linki do stron dostępne są na mojej stronie WWW.

13. FAQ

1. Co to jest Babel?

Babel to pakiet LaTeX opracowany przez Johannes Braamsa umożliwiający dostosowanie dokumentów do wybranego języka, np. polskiego. Dużo informacji na temat tego pakietu można znaleźć w [88minut]. Obecnie nie zaleca się używanie tego pakietu ze względu na pojawienie się pakietu ‘polski’ wchodzący w skład zestawu latex. Pakiet polski oferuje znacznie lepszą polonizację dokumentów niż babel, jest też łatwiejszy w stosowaniu. Także PDFy generowane z babla nie wyglądają najlepiej.

2. Jak kodować dokumenty znakami ISO-8859-2

Jeden sposób to użycie programu konwertującego. Jeżeli często modyfikujemy nasz plik, często musimy wykonać konwersję, a to jest niewygodne.

Drugi sposób to zainstalowanie w systemie klawiatury ISO-8859-2. Aby to wykonać należy pobrać ze strony <http://windows.online.pl/klawiat.htm> pliki konfiguracyjne (inne dla Windows 9x/Me, inne dla Windows 2000, XP, NT). Szczegółowa instrukcja instalacji klawiatury jest razem z plikami konfiguracyjnymi. Po zainstalowaniu klawiatura ISO dostępna jest tak, jak dowolny inny język w systemie Windows. Wygodne przełączanie się

między zainstalowanymi klawiaturami dostępne jest pod skrótem lewy Alt+Shift (to ustawienie można sobie zmienić). Ikona klawiatury (w win xp jest to pasek języka) można także wyświetlić sobie w systray. Dostęp do ustawień jest poprzez Panel Sterowania – Opcje regionalne i językowe – Języki – Szczegóły.

3. Jak można łatwo eksportować kody źródłowe programów do LaTeX'a?

Umożliwia to edytor SciTE. Plik – Eksport – Jako LaTeX. Podajemy nazwę pliku wyjściowego (nazwa.tex) i gotowe. Otrzymany plik możemy teraz skompilować.

4. Jak nauczyć SciTE języka polskiego?

Pierwsza rzecz to spolszczenie menu. Należy pobrać ze strony <http://www.tkaleta.republika.pl/> plik zawierający polskie napisy dla programu. Autor udostępnia wersje CP1250 i Latin2. Na 99% będzie potrzebna wersja CP1250 (to kodowanie nie ma nic wspólnego z kodowaniem dokumentów), gdyż nie sądzę żeby ktoś używał kodowania Latin2 dla wyświetlania menu w Windows. Następnie ściągnięty plik należy zapisać w katalogu SciTE pod nazwą `locale.properties`.

Druga sprawa to spowodowanie, żeby dokumenty zawierające polskie znaki były poprawnie wyświetlane na ekranie. W zasadzie można by się tym nie przejmować i wiedzieć, że ± ¶ Ł oznaczają ą ś ź, ale problem zaczyna się, jeżeli używamy eksportu dokumentu do LaTeX'a czy PDF – wyeksportowane zostaną takie krzaczki i to już nam nie zawsze odpowiada.

a) jeżeli używamy kodowania CP1250

to sprawa jest prosta. W pliku SciTEGlobal.properties znajdującym się w katalogu z SciTE należy wpisać linijki (wiersz numer około 190):

```
code.page=1250
```

```
character.set=238
```

i wykomentować znakiem # inne tego typu ustawienia.

b) jeżeli używamy kodowania Latin2

należy najpierw pobrać ze strony <http://windows.online.pl/klawiat.htm> ekranową czcionkę ISO i ją zainstalować w systemie (prawoklik -> instaluj). Teraz czcionka ta jest dostępna tak, jak wszystkie pozostałe czcionki w systemie: Arial, Times, Verdana itd., pod nazwą „EFN PolskieStrony 2000 10pt”. Ważne jest, że czcionka ta została tak zaprojektowana, że dobrze wygląda tylko w wielkości 10pt ☺.

Następnie w pliku SciTEGlobal.properties znajdującym się w katalogu z SciTE należy odszukać następujący fragment (wiersze numer około 270):

```
if PLAT_WIN
```

```
font.base=font:Verdana,size:10
```

```
font.small=font:Verdana,size:8
```

```
font.comment=font:Comic Sans MS,size:9
```

```
font.code.comment.box=$(font.comment)
```

```
font.code.comment.line=$(font.comment)
```

```
font.code.comment.doc=$(font.comment)
```

```
font.text=font:Times New Roman,size:11
```

```
font.text.comment=font:Verdana,size:9
```

```
font.embedded.base=font:Verdana,size:9
```

```
font.embedded.comment=font:Comic Sans MS,size:8
```

```
font.monospace=font:Courier New,size:10
```

```
font.vbs=font:Lucida Sans Unicode,size:10
```

Są to ustawienia używanych czcionek. W zasadzie wystarczy zastąpić pierwszą linię na
`font.base=font:EFN PolskieStrony 2000 10pt,size:10`

Jednak może okazać się to niewystarczające (jeżeli np. używamy polskich liter w komentarzach). Wówczas trzeba w odpowiednich pozostałych liniach zamienić nazwę czcionki na `EFN PolskieStrony 2000 10pt` i ustawić rozmiar na 10.

Aby ustawienia były widoczne należy ponownie uruchomić SciTE.

5. Jak nauczyć EditPlus języka polskiego?

Najpierw należy pobrać ze strony <http://windows.online.pl/klawiat.htm> **ekranowa** czcionkę ISO i ją zainstalować w systemie (prawoklik -> instaluj). Teraz czcionka ta jest dostępna tak jak wszystkie pozostałe czcionki w systemie: Arial, Times, Verdana itd., pod nazwą „EFN PolskieStrony 2000 10pt”. Ważne jest, że czcionka ta została tak zaprojektowana, że dobrze wygląda tylko w wielkości 10pt ☺.

Następnie w ustawieniach programu (Dokument – Permanent Settings – General – Fonts) wybieramy czcionkę „EFN PolskieStrony 2000 10pt” do wyświetlania zawartości dokumentu. Uwaga: program umożliwia wybranie innej czcionki dla wyświetlania na ekranie, inną dla wydruku i jeszcze inną dla innych wyjść.

14. Literatura

- [88minut] „Nie za krótkie wprowadzenie do systemu LaTeX2e” (tłumaczenie i opracowanie wersji polskiej: Janusz Gołdasz, Ryszard Kubiak, Tomasz Przechlewski) <http://www.gust.org.pl/doc.html>
- [FAQ] bardzo krótkie FAQ grup `pl.comp.dtp.tex` oraz `pl.comp.dtp.tex.gust` zebrane przez Grzegorza Sapijaszko, http://www.sapijaszko.net/pl_tex_faq.pdf
- [LATEX_WWW] http://www.cs.stir.ac.uk/guides/latex/tableofcontents2_1.html
- [HINTS] <http://www.fuw.edu.pl/~pliszka/hints/LaTeX.html>
- [NASA] <http://www.giss.nasa.gov/latex/ltx-2.html>
- [GUST] <http://www.gust.org.pl/doc.html>
- [WUJEK] <http://www.ia.pw.edu.pl/~wujek/tex/>
- [WUJEK_M] http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/macro_t.html
- [TUG] <http://www.tug.org/begin.html>
- [KOLOR] Macewicz W., „LaTeX na kolorowo”, Zeszyt 19 GUST
- [MYSZKA 1] Myszka W., „Tego nie znajdziesz w FAQ: (La)TeXa grafika”, Zeszyt 19 GUST
- [MYSZKA 2] Myszka W., „Włączanie grafik do tekstów w LATEX2e”, <http://www.immt.pwr.wroc.pl/~myszka/grafika>
- [MYSZKA 3] Myszka W., Rafajłowicz E., „LaTeX. Podręcznik użytkownika” ISBN 83-85190-60-0 Warszawa, wydaw. PLJ 1992
- [FLYNN] Peter Flynn „A beginner’s introduction to typesetting with LaTeX”
- [PDF 1] Sapijaszko G., „O lepszą jakość PDF-ów”, Zeszyt 19 GUST
- [PDF 2] Sapijaszko G., „Tworzenie dokumentów pdf przy pomocy LATEX-a”
- [MWCLS] Woliński M., „MWCLS – Moje własne klasy dokumentów dla LaTeX2e. Podręcznik użytkownika”