Narzędzia

Używanie TeX-a w systemie UNIX

Adam Dawidziuk

TeX, jak wiadomo, nie jest zaopatrzony w standardowy interfejs użytkownika, co daje dużą swobodę wyboru metod przygotowywania dokumentów źródłowych. O ile format plików wejściowych i sposób ich przetwarzania przez TeX-a pozostaje z założenia niezmienny, to narzędzia wspomagające wprowadzanie treści dokumentów we właściwym formacie są nieustannie rozwijane. W artykule jest opisany jeden z możliwych sposobów używania TeX-a w systemie UNIX, ze szczególnym uwzględnieniem pakietu AUC TeX.

Przetwarzanie danych

Brak narzuconego i okienkowego interfejsu użytkownika powoduje, że TeX bywa postrzegany jako program przestarzały i trudny. Przeciętny użytkownik komputera zwykle nie jest w stanie zrozumieć sposobu działania TeX-a, a jeszcze trudniej dostrzec mu korzyści, jakie można osiągnąć przygotowując dokumenty w ten właśnie sposób. Pierwszą i najpoważniejszą barierą pojęciową jest fakt, że TeX nie jest systemem kompletnym, ale wymaga korzystania z wielu dodatkowych narzędzi wspomagających pracę. Wprawdzie mało kto wprowadza treść dokumentu z klawiatury w chwili, gdy TeX pokaże na ekranie dwie gwiazdki, ale na pomysł stworzenia sobie wygodnego i efektywnego środowiska pracy też niestety wpada nieliczne grono użytkowników.

Działanie TeX-a można sprowadzić do przetwarzania plików. TeX czyta dokument źródłowy, zwykle z pliku o nazwie z przyrostkiem `.tex`, i tworzy obraz kolejnych stron dokumentu w pliku o nazwie kończącej się napisem `.dvi`. Zadaniem użytkownika jest przygotowanie plików wejściowych i uruchomienie ciągu przetworzeń, które doprowadzą do pożądanego wyniku. Zarówno przygotowanie danych, jak i ich przetwarzanie można zorganizować w zasadzie w dowolny sposób. Istnieją dwie istotnie różne metody tworzenia plików z danymi wejściowymi: generowanie i wpisywanie.

Generowanie danych wejściowych do TeX-a to zagadnienie bardzo rozległe i mające wiele praktycznych zastosowań. Sam proces przygotowania danych, ich obróbka i metody generowania mogą być bardzo różne. Wszystko zależy od celu przetwarzania i postawionych wymagań. TeX jest doskonałym narzędziem do formatowania wydruków z baz danych (np. słowniki, sprawozdania) pozwalającym na uzyskanie wysokiej jakości typograficznej przy całkowitej automatyzacji procesu składania tekstu. Często użytkownik (autor tekstu) nie ma bezpośredniego kontaktu z TeX-em i nie musi znać specyficznego dla TeX-a systemu oznaczeń tekstu. Jednak jest to temat znacznie wykraczający poza ramy tego artykułu.

Większość użytkowników zadowala się zwyczajnym wpisywaniem tekstu. W takim wypadku najważniejszy jest ergonomiczny i dobrze opanowany edytor tekstów. Zwykle każdy ma swój ulubiony edytor, w którym wszystko mu wychodzi najłatwiej. I prawdopodobnie to będzie ten edytor, którego należy użyć tworząc pliki źródłowe do przetwarzania TeX-em. Jeśli jednak zamierzamy pisać dużo, to nie powinniśmy eksperymentować z edytorami zbyt prostymi. Czas poświęcony na poznanie inteligentnego, konfigurowalnego edytora zwróci się już po kilku dniach pisania. Z edytorów dostępnych w systemie UNIX doskonały jest `vim`. Można go „nauczyć” wielu pożytecznych skrótów, a jego ostatnie wersje potrafią rozpoznawać składnię i zwiększać czytelność plików źródłowych przez stosowanie kolorów. Lepszy jest tylko `emacs` (tak twierdzą użytkownicy Emacsa). Niestety, zarówno `vim` jak i Emacs u niektórych wywołują reakcje alergiczne, a na to rady nie ma.

Zwolennicy okienek MS nie powinni czuć się pokrzywdzeni (a przynajmniej nie z powodu braku edytora). Z rzeczy dobrych godny polecenia jest WinEdit, który może bardzo dużo i składnię pliku TeX-owego zna na tyle dobrze, że wyróżnia kolorami odpowiednie fragmenty. Jest łatwo konfigurowalny w zakresie wystarczającym do efektywnej pracy. Między innymi umożliwia powiązanie z polami menu dowolnych skryptów np. przetwarzających pliki źródłowe. Daje się używać bez zbyt częstego sięgania do myszy. Jedną z jego głównych zalet jest sprawdzanie na bieżąco składni wprowadzanych słów – te których brak w słowniku są wyróżniane. Do podstawowych wad należy konieczność konfigurowania właściwie wszystkiego od początku oraz to, że, w przeciwieństwie do vima i Emacsa nie jest to program bezpłatny. Cena jest jednak umiarkowana.

Przewidując trwały związek z TEX-em warto usprawnić przetwarzanie pliku źródłowego do postaci wynikowej. Nie ma wielkiego problemu, jeśli przetwarzaniu podlega jeden plik, a formatem wyjściowym jest dvi. Już jednak konwersja pliku dvi do PostScriptu oznacza konieczność uruchomienia dodatkowego programu. Jeśli jeszcze dołożymy do tego tworzone podczas pisania rysunki (w szczególności rysunki generowane, np. METAPOST-em) to liczba programów, które trzeba uruchomić w ściśle określonym porządku, zniechęca do robienia tego ręcznie. Łatwo się pomylić. Dobrym pomysłem jest napisanie skryptu, który wykonuje wszystkie operacje. Przetwarzanie za każdym razem wszystkich plików źródłowych nie zawsze jest konieczne, a znacznie wydłuża czas oczekiwania na wynik. W takiej sytuacji najwygodniej użyć programu make.

Doświadczony użytkownik TEX-a zwykle ma własny zestaw programów, które czynią życie łatwiejszym. Często w skład tego zestawu wchodzi dodatkowo jakiś edytor strumieniowy np. sed, czy częściej awk lub perl (najbardziej wszechstronny). W przypadku umieszczania w dokumentach ilustracji lub zdjęć należy uwzględnić specjalizowany programy do przetwarzania obrazków, itd. Podstawową zaletą TEX jest łatwość zintegrowania go z dowolnym używanym systemem.

Każdy może więc używać TEX dokładnie tak jak lubi i tak jak mu wygodnie. Trudno jest w jednym krótkim tekście omówić wszystkie typowe przypadki. Ograniczę się do przykładu szczególnego – systemu UNIX. Przyczyn tego wyboru mógłbym wymienić wiele. Przede wszystkim cały UNIX zbudowany jest z programów wykonujących dobrze określonym funkcje, takich „klocków”, z których można zbudować dowolnie skomplikowane systemy przetwarzania informacji. Nie posiadający interfejsu użytkownika, przetwarzający pliki w inne pliki i konfigurowalny TEX łatwo daje się dopasować do potrzeb użytkownika UNIXa. Oczywiście istnieje mnóstwo różnych i dobrych sposobów pracy z TEX-em. Zarówno podany przykład edytora (Emacs z AUC TEX-em), jak i techniki przetwarzania dokumentów noszą wyraźne piętno moich osobistych preferencji.

AUC TEX

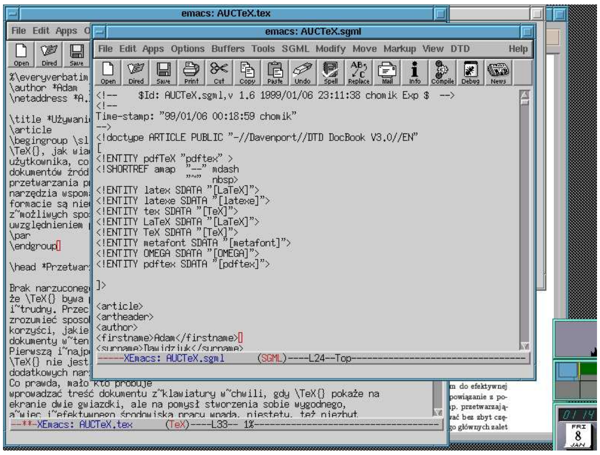
AUC TEX to dostępne w edytorze emacs zintegrowane środowisko do tworzenia i przetwarzania dokumentów TEX-owych i LATEX-owych. Możliwości AUC TEX-a są ogromne, na szczęście nie trzeba wiedzieć wszystkiego, aby z niego korzystać. AUC TEX-a można oswajać „krok po kroku”, podobnie jak i samego Emacsa.

Emacs. Edytor emacs ma długą historię w systemie UNIX. W dużej części został napisany w języku Lisp (dokładniej w dialekcie Emacs Lisp). Jest to język funkcyjny służący przede wszystkim do porozumiewania się z Emacsem. Lisp bywał chętnie stosowany w tworzeniu tzw. „systemów sztucznej inteligencji”. Nie dziwi więc naprawdę nieprzeciętna inteligencja edytora Emacs jeśli uświadomimy sobie, że jest to największy z popularnych programów napisanych w Lispie. A możliwości i umiejętności Emacsa są chyba niezgłębione (strzyże, śpiewa, tańczy, gotuje, krawaty wiąże. . .).

Edytor ten bywa niezbyt lubiany przez początkujących użytkowników ze względu na sposób wydawania poleceń i niezbyt zrozumiałą terminologię stosowaną w dokumentacji i komunikatach. Koncepcja porozumiewania się z edytorem jest rzeczywiście mocno egzotyczna, ale wbrew pozorom ma głęboki sens. Kluczem do sukcesu jest znalezienie klawisza `Meta` na klawiaturze. Wciśnięcie tego klawisza ze znakiem `x` (oznaczane jako `Meta-x` lub `M-x`) rozpoczyna wprowadzanie poleceń. W klawiaturze komputera PC zwykle klawiszem `Meta` jest `Alt`. Jeśli jednak `Alt` został przedefiniowany (np. w celu wprowadzania polskich znaków) to `M-x` można wprowadzić sekwencyjnie wciskając najpierw klawisz `Esc`, a potem `x`. Tego typu problemy nie istnieją z klawiszem `Ctrl`. Podstawowymi chwytami są `Ctrl-x` i `Ctrl-c`, opisywane często jako `C-x` i `C-c`.

Mocną stroną Emacsa jest dokumentacja (`C-h`). Funkcje przypisane klawiszom zmieniają się w zależności od trybu pracy (*mode*), co zawsze można sprawdzić wpisując `C-h m`. Wiedząc, że Emacs może otwierać wiele plików jednocześnie, ale każdy w innym buforze (*buffer*), że w środowisku okienkowym może otwierać wiele okien zwanych ramkami (*frames*) i że każda ramka może być podzielona na wiele „okien” (*window*) – por. rys. 1 – można bez problemu zrozumieć język dokumentacji.

Istnieje wiele odmian Emacsa. Na potrzeby tego artykułu założymy sobie, że dysponujemy zainstalowaną, współczesną wersją XEmacsa. Założenie to jest uzasadnione, ponieważ normą są już komputery, na których działające okienka graficzne nie



Dwie ramki edytora XEmacs, w jednej wersja TEX-owa tego tekstu, w drugiej – zapisana w formacie DocBook

powodują zatrzymania systemu. A XEmacs dobrze wygląda w okienkach. Poza tym w XEmacsie AUC TEX jest domyślnie zainstalowany, co pozwala skrócić ten tekst o rozdział poświęcony instalacji. Dodatkowo nie będę rozwijać problemu konfiguracji, którą w XEmacsie można wykonać za pomocą myszy. Wszystkie ustawienia AUC TEX-a wykonuje się wybierając myszą z menu Options Customize Emacs Wp Tex AUC Tex, a potem już według uznania. W to samo miejsce można trafić wciskając `M-x customize`, tyle że po drodze jest więcej objaśnień i trudniej się zagubić.

Jeśli ktoś nie dysponuje XEmacsem lub nie lubi myszy, to będzie musiał wpisywać odpowiednie zaklęcia do pliku `.emacs`, który znajduje się w katalogu domowym. Pomocne może być odszukanie pliku `tex.el` i `tex-site.el`, które leżą gdzieś z innymi plikami Emacsa (ale w różnych katalogach – przyda się program `find`). Można w nich podejrzeć, jak są zadawane wartości początkowe list i zmiennych i je zmodyfikować. Nie należy przy tym zmieniać pliku `tex.el` (ani jego skompilowanej wersji `tex.elc`) lecz kopiować co trzeba do pliku `.emacs`. Jeśli zmiana ma dotyczyć wszystkich użytkowników należy odpowiedni fragment pliku `tex.el` skopiować do pliku `tex-site.el` i tam pozamieniać wszystko to, co wymaga zmiany.

I jeszcze jedna uwaga dla początkujących: w Emacsie można zapisać plik z bufora na dysk wciskając `C-x C-s`, a pracę z Emacsem zakończyć sekwencją `C-x C-c`. Emacs nie jest jednak małym programem, więc szanując swój czas lepiej nie otwierać go i nie zamykać bez wyraźnej potrzeby, zwłaszcza, że raz otwarty mocno nie przeszkadza.

Do efektywnego wykorzystania Emacsa na potrzeby języka polskiego jest potrzebna polska klawiatura (każdy robi to jak lubi, jedna z klasycznych metod w systemie XWindow sprowadza się do wykorzystania `xmodmap`) oraz polski słownik do programu `ispell`. Zagadnienia te są dokładnie omówione w tekstach poświęconych polskim „ogonkom”.

Uruchomienie AUC TEX-a. Po uruchomieniu edytora Emacs utworzenie nowego pliku jest możliwe po wciśnięciu `C-x C-f` i podaniu nazwy. Podając nazwę pliku z przyrostkiem `.tex` zmusimy edytor do wejścia w tryb `LaTeX-mode`. Informacja o tym powinna pojawić się w wierszu trybu pracy (*Mode Line*) w dolnej części okna jako napis `LaTeX`. Można też wymusić wejście w tryb `LaTeX` wywołując polecenie `LaTeX-mode` (`M-x LaTeX-mode`). Do rozwijania nazw można użyć znaku klawisza tabulacji (`Tab`).

Jeśli powyższe działania nie prowadzą do uruchomienia AUC TEX-a, to albo nie jest on zainstalowany, albo w pliku `.emacs` brakuje wiersza

```
(require 'tex-site)
```

Brak zainstalowanego AUC TEX-a może prowadzić do pomyłek ponieważ Emacs wchodzi w takiej sytuacji w dużo prostszy tryb wspomagania LATEX-a, a wszystko działa inaczej (w szczególności „klawiszologia” jest inna).

Słowa kluczowe LATEX-a. Przy tworzeniu pliku LATEX-owego pewne oznaczenia tekstu pojawiają się częściej niż inne. Na szczególną uwagę zasługują mechanizmy AUC TEX-a ułatwiające wpisywanie tych właśnie często występujących słów kluczowych. Najczęściej wprowadza się do tekstu oznaczenia środowisk i ich nazwy, „zwykłe” polecenia TEX-owe, śródtytuły oraz polecenia zmiany kroju pisma.

Najważniejszą konstrukcją LATEX-a są środowiska, czyli pary `\begin{*} \end{*}`. Do wprowadzania środowisk służy polecenie `LaTeX-enviroment` domyślnie przypisane do sekwencji `C-c C-e`. Po wydaniu tego polecenia AUC TEX pyta o nazwę środowiska i ewentualne dodatkowe dane. Przy pomocy klawisza `Tab` lub spacji można rozwijać częściowo wpisane nazwy. AUC TEX zna mnóstwo standardowych środowisk, poza tym potrafi uczyć się nowych analizując włączane pakiety i nagłówek pliku.

Otwarte środowisko można zamknąć poleceniem `LaTeX-close-enviroment` (`C-c ]`). Zdecydowanie jednak najbezpieczniejsze jest wprowadzanie kompletnych środowisk przy użyciu `C-c C-e`.

Poza środowiskami w pliku LATEX-owym często pojawiają się tradycyjne polecenia TEX-a, czyli słowa rozpoczynające się znakiem `\`. Zapamiętanie wszystkich poleceń jest trudne, a przy ich wprowadzaniu łatwo o błąd. Dlatego należy korzystać z polecenia `LaTeX-complete-symbol`. Jeśli pisane słowo rozpoczyna się znakiem `\` to wystarczy wcisnąć `M-Tab` aby je zakończyć, lub uzyskać listę możliwych zakończeń.

Inny sposób wpisywania makr TEX-owych polega na wykorzystaniu polecenia `TeX-insert-macro` (`C-c C-m`). Aby użyć jeszcze innej metody należy np. w pliku `.emacs` ustawić różną od `nil` wartość zmiennej `TeX-electric-escape`

```
(setq TeX-electric-escape t)
```

Wiąże to znak `\` z funkcją `TeX-electric-macro`, co umożliwia uruchamianie mechanizmu rozwijania makr przez wpisanie tego znaku. Jest to metoda najszybsza.

AUC TEX domyślnie wstawia parę pustych nawiasów `{}` po wpisaniu makra, co zabezpiecza przed „zniknięciem” spację znajdującą się za tym makrem. Jeśli nawiasy nie są mile widziane należy zmiennej `TeX-insert-braces` przypisać wartość `nil`.

Inaczej oznacza się elementy listy `\item`. Służy do tego polecenie `LaTeX-insert-item` (`M-Return`), którego działanie zależy od kontekstu.

Kolejnym często występującym elementem tekstu są śródtytuły, oznaczane słowami kluczowymi `\chapter`, `\section` itd. Można ułatwić sobie życie korzystając z polecenia `LaTeX-section` (`C-c C-s`). Wywołanie tego polecenia spowoduje wstawienie oznaczenia śródtytułu, przy czym domyślnie AUC TEX zapyta o tytuł i etykietę związaną z tym śródtytułem (argument polecenia `\label`). Domyślny poziom śródtytułu można zmienić wpisując ręcznie właściwe słowo kluczowe (z rozwijaniem) lub podając argument (`C-u`) przed wywołaniem polecenia `LaTeX-section`. Jeśli argument jest ujemny, to poziom zostanie zmieniony w dół o jeden.

I tak, jeśli ostatnio wstawiony był śródtytuł na poziomie `\subsection` to będzie to aktualny poziom domyślny. Sekwencja `C-c C-s` oznacza polecenie wpisania śródtytułu na tym właśnie poziomie. Podanie `C-u C-c C-s` to wpisanie `\subsubsection`, a `C-u -1 C-c C-s` spowoduje wstawienie `\section`. Przedrostki nazw w etykietach `\label` są kontrolowane zmienną `LaTeX-section-label`. Jeśli wartością tej zmiennej jest `nil`, to odnośniki nie będą wstawiane.

AUC TEX udostępnia również silny mechanizm zmiany kroju pisma. Jego podstawą jest polecenie `TeX-font` (skrót `C-c C-f`), po którym podaje się argument określający pożądaną zmianę kroju pisma. I tak `C-e` powoduje wpisanie `\emph{}`, `C-i` – `\textit{}`, `C-t` – `\texttt{}` itd, włącznie z usuwaniem ustawienia kroju pisma `C-d`. Lista możliwości jest wypisywana po podaniu spacji jako argumentu:

```
Font list:

     C-b        \textbf{ }
     C-c        \textsc{ }
     C-e          \emph{ }
     C-f        \textsf{ }
     tab        \textit{ }
  return        \textmd{ }
     C-n    \textnormal{ }
     C-r        \textrm{ }
     C-s        \textsl{ }
     C-t        \texttt{ }
     C-u        \textup{ }
     C-d     -- delete font
```

Jeśli polecenie `TeX-font` jest poprzedzone argumentem (`C-u`), to zostanie zmieniony krój pisma w miejscu, w którym znajduje się kursor. Przykładowo jeżeli kursor jest umieszczony na tekście będącym argumentem polecenia `\textit{}`, to sekwencja `C-u C-c C-f C-e` zmieni polecenie `\textit` na `\emph`.

Inne elementy tekstu. W treści dokumentu pojawiają się nie tylko słowa kluczowe, ale także znaki specjalne: cudzysłów, znak dolara $, nawiasy klamrowe `{}` oraz znak komentarza %.

W TEX-u nie używamy znaków `"` na oznaczenie cudzysłowu. W tekstach pisanych po angielsku używamy znaku ` `` ` jako cudzysłowu otwierającego i znaku `''` jako cudzysłowu zamykającego. Jeśli użytkownik wpisuje `"` to AUC TEX zadba, aby w tekście pojawiło się to co trzeba. W przypadku języka polskiego (i spolszczonego LATEX-a) cudzysłów otwierający oznacza się w tekście dwoma przecinkami `,,`. Aby polecenie `TeX-insert-quote` wywoływane klawiszem `"` zadziałało prawidłowo należy zmiennej `TeX-open-quote` przypisać wartość `,,` (czyli dwa przecinki).

Znaki dolara $ w pliku TEX-owym obejmują tekst wzoru matematycznego, a w podwójne dolary $$ jest ujęty wzór wystawiony. W celu lepszej kontroli otwierania i zamykania wzorów wprowadzenie znaku dolara powoduje wywołanie polecenia `TeX-insert-dollar`. Polecenie to, oprócz wprowadzenie do pliku znaku $ sprawdza stan sparowania tych znaków. Dzięki temu dokładnie wiadomo, gdzie się zaczyna i gdzie kończy środowisko matematyczne, a przy okazji nie da się wzoru rozpoczętego dwoma dolarami zamknąć tylko jednym.

Polecenie `TeX-insert-braces` pozwala wprowadzać nawiasy klamrowe parami. Skrót klawiaturowy jest intuicyjny: `C-c {`.

Przy wprowadzaniu i usuwaniu znaku komentarza na początku wierszy przydaje się polecenie `TeX-comment-regin` (`C-c ;`). Umożliwia ono wstawienie znaku komentarza na początku każdego wiersza zaznaczonego fragmentu tekstu (*region*) czyli obszaru od punktu oznaczonego przez `C-spacja` do bieżącej pozycji kursora (lub zaznaczonego myszą). Podobnie działa polecenie `TeX-comment-paragraph` (`C-c %`) dla akapitu. Aby usunąć znak komentarza wpisujemy `C-u - C-c ;` lub `C-u - C-c %`.

Matematyka. W przypadku wprowadzania dużej liczby wzorów matematycznych warto mieć pod ręką mechanizm przyspieszający wpisywanie słów takich jak `\upsilon` czy `\rightarrow`. Mechanizm taki uruchamia się (i wyłącza) w AUC TEX-u poleceniem `LaTeX-math-mode` (`C-c ~`). W trybie matematycznym po wprowadzeniu znaku „`‘`” klawiatura zmienia znaczenie i zamiast zwykłych znaków pojedynczym klawiszem można wprowadzać symbole matematyczne. I tak (w trybie matematycznym) sekwencja klawiszy `‘d` spowoduje wpisanie `\delta`, `‘G` – `\Gamma`, a `‘cN` – `{\cal N}`.

Pożyteczne bywa podanie argumentu (`C-u`) przed symbolem matematycznym. Sekwencja klawiszy `C-u ‘e` spowoduje wpisanie do bufora `$\epsilon$` – co znacznie przyśpiesza wstawianie pojedynczych symboli w bloku tekstu.

Bez możliwości definiowania własnych skrótów przydatność trybu matematycznego byłaby znacznie ograniczona. Można to robić modyfikując na prywatne potrzeby listę `LaTeX-math-list`. Nie sprawi też problemu zamiana klawisza `‘` na jakiś poręczniejszy (jeśli ktoś musi) – należy zmienić wartość zmiennej `LaTeX-math-abberv-prefix`.

Własne definicje. Powróćmy do rozwijania słów kluczowych. Często się zdarza, że definiujemy własne makra. AUC TEX doskonale nas rozumie. Jeśli wpiszemy definicję LATEX-ową (`\renewcommand`) to AUC TEX ją rozpozna i się jej „nauczy” – będzie ona dostępna poprzez mechanizm rozwijania początków wyrazów. Możliwe jest również rozpoznawanie definicji TEX-a wprowadzanych jako `\def`.

AUC TEX rozpoznaje makrodefinicje zapisane w pliku podczas parsowania dokumentu po wydaniu polecenia `TeX-normal-mode` (`C-c C-n`). Jeśli więc wpiszemy w pliku nową definicję, należy wcisnąć `C-c C-n` i nie musimy jej dokładnie pamiętać. Informacje o makrodefinicjach zapisywane są w pliku `./auto/nazwa.el`.

Można wymusić parsowanie dokumentu natychmiast po jego wczytaniu oraz umieszczanie informacji o nim w katalogu auto przy każdym zapisie na dysk np. przez wpisanie do pliku `.emacs` następujących wierszy:

```
; parsowanie przy wczytywaniu:
(setq TeX-parse-self t)
; parsowanie przy zapisie:
(setq TeX-auto-save t)
```

Zmianę pliku `./auto/nazwa.el` powoduje również wstawienie do tekstu dokumentu nazwy etykiety `\label`. Wstawiając odnośnik do tej etykiety (`\ref`, `\pageref`) można etykietę wybrać z listy etykiet pokazywanej przez mechanizm rozwijania (`Tab`). I to jest naprawdę cenna właściwość.

Sposób parsowania dokumentu można konfigurować (po szczegóły odsyłam do dokumentacji). Na przykład jeżeli mamy słaby komputer i chcemy, aby rozpoznawane były tylko etykiety musimy zadbać o następujące ustawienia (znów plik `.emacs`):

```
(setq-default
  TeX-auto-regexp-list
  'LaTeX-auto-label-regexp-list)
```

Przetwarzanie z wnętrza AUC TEX-a. Na uwagę zasługuje możliwość uruchamiania TEX-a i oglądania plików .dvi bez konieczności opuszczania Emacsa. Służy do tego polecenie przypisane do sekwencji `C-c C-c`, czyli `TeX-command-master`. Po wydaniu tego polecenia AUC TEX zapyta jakie działania ma podjąć. Jeśli w pliku były dokonywane jakieś zmiany, to zaproponuje jego przetworzenie TEX-em. Po przetworzeniu i kolejnym wciśnięciu `C-c C-c` będzie chciał uruchomić przeglądarkę `.dvi`. Jeżeli nie wystąpią żadne błędy w przetwarzaniu, to w okienku obok obejrzymy wynik składu.

Standardowe polecenia przetwarzania pliku źródłowego można łatwo rozszerzyć o nasze ulubione:

mex, platex czy pdftex. Najłatwiej przez menu XEmacsa. W razie potrzeby można skopiować odpowiedni fragment z pliku `tex.el` (czy wręcz `tex.elc`). Prawdopodobnie warto zmienioną listę `TeX-command-list` wstawić do pliku `tex-site.el` – niech się cieszą wszyscy użytkownicy.

Naprawdę niezwykle użyteczna (i trudna do zaimplementowania poza edytorem) jest możliwość przetworzenia tylko fragmentu pliku źródłowego. W przypadku składania skomplikowanego dokumentu można TEX-ować tylko fragment nad którym pracujemy, co zaoszczędza mnóstwo czasu. Służy do tego polecenie `TeX-command-region` przypisane do sekwencji `C-c C-r`. Wszystko działa tak samo jak w przypadku `TeX-command-master`, z jednym wyjątkiem – AUC TEX buduje z nagłówka i zaznaczonego fragmentu tekstu nowy plik, który przetwarza TEX-em. Kolejne `C-c C-r` umożliwia obejrzenie właściwego pliku `.dvi`. Trzeba się tylko pilnować, żeby nie definiować żadnych makr globalnych poza nagłówkiem. W plikach LATEX-owych końcem nagłówka jest wiersz zawierający napis `\begin{document}`. W plikach TEX-owych jest to natomiast `%**end of header`. Znacznik końca nagłówka można zdefiniować jako wyrażenie regularne określając wartość zmiennej `TeX-header-end`.

W przypadku wystąpienia błędów w czasie przetwarzania można oglądać je kolejno przy pomocy polecenia `TeX-next-error` (`` C-c ` ``). AUC TEX opatruje błędy TEX-a dodatkowym komentarzem, co może pomóc początkującym użytkownikom.

I więcej. . . . Zatrzymajmy się jeszcze nad tym, co widzimy na ekranie. Emacs (w szczególności XEmacs) potrafi inteligentnie zaznaczać odpowiednie fragmenty tekstu kolorami. Konfigurowanie kolorów w XEmacsie jest proste (myszą w sekcji *faces* – dlatego warto używać XEmacsa). Jeśli kolorowanie nie włączyło się z własnej woli można je wymusić poleceniem `font-lock-mode`.

Nawet bez kolorowania treść dokumentu można wprowadzić w niezwykle czytelnej formie dzięki zaawansowanym poleceniom formatowania tekstu. W wolnej chwili warto wypróbować polecenia:

`LaTeX-fill-paragraph` (`C-c C-q C-p` lub `M-q`),
`LaTeX-fill-enviroment` (`C-c C-q C-e`),
`LaTeX-fill-section` (`C-c C-q C-s`) oraz
`LaTeX-fill-region` (`C-c C-q C-r` lub `M-g`).

Ograniczona objętość artykułu nie pozwala omówić wszystkich możliwości AUC TEX-a. Nie zmieściły się tematy obszerne, takie jak konfigurowanie działania AUC TEX-a i liczne szczegółowe, np. praca z dokumentami hierarchicznymi. Wszystko można doczytać w dokumentacji. Czytając trzeba pamiętać, że AUC TEX jest częścią edytora Emacs i przez cały czas można korzystać ze wszystkiego, co Emacs oferuje. Przykładem jest niezależny (ale wspierany przez AUC TEX-a) tryb „ukrywania” fragmentów tekstu, włączany/wyłączany poleceniem `outline-minor-mode`. Pojawiające się dodatkowe pozycje menu (i polecenia) pozwalają wyświetlić zamiast rozdziału, podrozdziału itd. tylko fragment pierwszego wiersza z tytułem (bez zmiany tekstu). Możliwe jest np. pisanie czwartego rozdziału trzy wiersze od końca nagłówka, albo poruszanie się po tekście tak, jakby składał się on tylko ze śródtytułów.

Przetwarzanie zewnętrzne

W całej zabawie z Emacsem najmilsze jest to, że wcale nie trzeba go używać. Tekst można przygotować korzystając z jakiegokolwiek edytora (np. . . .Emacsa), a pliki źródłowe przetwarzać zewnętrznie. Przypuśćmy, że kolega podesłał nam 46 felietonów, które napisał w ostatnim roku. Wszystkie, to kompletne pliki LATEX-owe. W takiej sytuacji TEX-owanie z wnętrza edytora zajmie zbyt dużo czasu. Najszybszą metodą będzie umieszczenie wszystkich plików w jednym katalogu i napisanie w terminalu mniej więcej czegoś takiego:

```
~$ for i in *.tex; do latex $i; done
```

Nie zawsze jednak jest tak prosto. Często otrzymanie wyniku końcowego wymaga wielu przetworzeń. Wpisywanie ciągu tych samych poleceń po każdej modyfikacji pliku bywa męczące. Dlatego mądrzy ludzie stworzyli program `make` (lub `gnumake`). Wszystkie polecenia wystarczy wpisać raz ale do pliku `Makefile` w bieżącym katalogu. W najprostszym przypadku plik ten będzie wyglądał tak:

```
all: referat.ps

%.dvi: %.tex
        latex $<

%.ps: %.dvi
        dvips $<

%.pdf: %.tex
        pdflatex $<
```

Z zawartości pliku wynika, że pracujemy nad referatem (tworzymy plik `referat.tex`) i że używamy przeglądarki plików PS. Teraz całe TEX-owanie sprowadza się do uruchomienia w terminalu programu `make`. Program ten domyślnie wykona pierwszą regułę z pliku Makefile (`all: referat.ps` – zamiast nazwy `all` można wstawić dowolne słowo, np. `Psia-Buda`) i dowie się, że jesteśmy zainteresowani plikiem `.ps`. Z trzeciej reguły program wywnioskuje, że pliki `.ps` robi się z plików `.dvi` poleceniem `dvips`. Nie ma jednak pliku `.dvi`, ale o tym jak go zrobić mówi druga reguła. A plik referat.tex jest na dysku.

Ostatnia reguła zostanie wykorzystana gdy wykonamy polecenia `make referat.pdf` – wtedy zostanie utworzony plik `referat.pdf`.

Najważniejszą cechą programu make jest to, że do przetwarzania dochodzi tylko wtedy, gdy program źródłowy był modyfikowany po ostatnim przetworzeniu (plik wynikowy jest starszy od pliku źródłowego). Dzięki temu można w terminalu napisać takie (lub podobne) zaklęcie:

```
~$ while [ TRUE ]; do make;
  sleep 5; done
```

Tak potraktowany interpretator poleceń co 5 sekund będzie uruchamiał program make, który przetworzy plik źródłowy tylko wtedy, gdy ten ostatni został zapisany na dysk po ostatnim przetworzeniu. Teraz wystarczy tylko myszą wskazać przeglądarkę, żeby zobaczyć, co „naskładaliśmy” do tej pory (przeglądarka powinna reagować na modyfikację przeglądanego pliku). Ta technika jest wrażliwa na zapisywanie pliku z błędem. Oczywiście, jeśli mamy dobry edytor, to wpisanie złego słowa kluczowego, czy nie domknięcie środowiska lub nawiasu nie jest proste.

Co się jednak stanie, gdy popełnimy błąd, na którym TEX się zatrzyma? Po pierwsze możemy przeczytać uważnie o co chodzi i w edytorze przeskoczyć do właściwego wiersza, a TEX-owi powiedzieć np. x. Jeżeli jednak ustawimy zmienną środowiska:

```
TEXEDIT='emacsclient +%d %s'
```

lub w przypadku XEmacsa:

```
TEXEDIT='enu +%d %s'
```

i jeżeli używamy Emacsa w którym powiemy `M-x server-start` (`gnuserv-start`), to możemy odpowiedzieć TEX-owi e (od *edit*). Wtedy Emacs wczyta plik z błędem i kursorem wskaże wiersz, co do którego TEX ma wątpliwości. Po zakończeniu poprawek należy zapisać plik (`C-x C-s`) i powiedzieć klientowi, że już po wszystkim (`C-x #`). Dla jasności: serwer to Emacs, a klient to TEX...

Korzystanie z emacsowego klienta i serwera pomaga w wielu trudnych sytuacjach. Wyobraźmy sobie książkę mającą 42 rozdziały, każdy zapisany w oddzielnym pliku. Jeśli w kilkunastu plikach zamiast `\ala` omyłkowo wpisaliśmy `\ola`, z tym że czasem `\ola` jest dobrze i ma zostać, to zanosi się na dłuższe poprawianie. A można z poziomu interpretatora poleceń shell napisać:

```
~$  gnuclient `grep -l \\ola *.tex`
```

Potem wystarczy wciskanie klawisza `C-s` i czasem `C-x e` (pierwszy raz oczywiście również `C-x (` i `C-x )`). Jeżeli jednak ktoś ma tendencje do zapominania o takich szczegółach jak `-l`, to niech lepiej próbuje się uczyć krok po kroku... Na razie. Po jakimś czasie może okazać się potrzebny dodatkowy wiersz w pliku `.emacs`:

```
(gnuserv-start)
```

Bardzo niechętnie używam możliwości przetwarzania pliku TEX-owego z wnętrza edytora. Nawet pisząc niewielki dokument warto (moim zdaniem) korzystać z programu make. Już kilka razy zapomniałem uruchomić dvips-a (mam drukarkę PS), w efekcie drukując nieaktualną wersję dokumentu. Dlatego mi wystarcza, gdy wewnątrz edytora można wykonać tylko jedno polecenie – make.

Podsumowanie

Współczesne narzędzia umożliwiają wygodne i bezbolesne korzystanie z dobrodziejstw oferowanych przez TEX-a. We współczesnym sprzęcie karta graficzna i monitor pozwalają na pracę w systemie okienkowym (np. XWindow); można także na bieżąco oglądać w sąsiednim oknie gotowy dokument. W przypadku korzystania z XEmacsa (lub podobnego edytora) pracować z LATEX-em mogą nawet ci użytkownicy, którzy nie zdejmują ręki z myszy.

Mam nadzieję, że udało mi się pokazać, jak duże możliwości stoją przed kimś, kto chce wygodnie i efektywnie używać TEX-a. Opisałem tylko najprostsze warianty konfiguracji środowiska pracy. Istnieje wiele sposobów i sposobików, o których nie wspomniałem i jak sądzę, jeszcze więcej takich, o których nigdy nie słyszałem. Wszystkim życzę miłego TEX-owania.

◇ Adam Dawidziuk
A.Dawidziuk@GUST.org.pl