

Programy do przetwarzania tekstów koreańskich i polskich

Michał Piskorski

Instytut Orientalistyczny UW

Sekcja Koreanistyki

ul. Bednarska 2/4, Warszawa

11 grudnia 1999

1 Edytory w praktyce orientalisty

Jeszcze kilka lat temu podstawowym narzędziem do zapisu tekstów, które było w posiadaniu orientalisty, był ołówek lub długopis. Maszyny do pisania pozwalały zaledwie na zapis tłumaczenia wersji oryginalnej. Niejednokrotnie nawet korzystanie z transkrypcji języka źródłowego wymagało ręcznej korekty tych znaków, które zawierały dodatkowe akcenty czy ogonki.

Pojawienie się elektronicznych maszyn do pisania było początkiem przełomu, ale dopiero swobodna możliwość komputerowego generowania znaków dowolnego kształtu na ekranie, ich obróbka i przenoszenie końcowego rezultatu na papier za pomocą drukarek dały początek nowemu zjawisku, które możemy określić jako komputerowe przetwarzanie tekstu.

Dla orientalisty otworzyła się droga do nowoczesnej pracy nad tekstem pisanym. Wydawać by się mogło, że los długopisu został przesądzony. Niestety, rzeczywistość kreślona przez firmy produkujące oprogramowanie dla komputerów okazała się być niesprzyjająca dla tych, którzy w swojej pracy chcieli wykroczyć poza granice jednego języka, ewentualnie pewnego, ale ograniczonego przez twórcę oprogramowania, ich zbioru.

W pracy nad językiem koreańskim, choć dotyczy to również i innych języków orientalnych, często napotykanym problemem jest brak możliwości komputerowej obróbki tego języka w taki sposób, że równocześnie z tekstem oryginalnym można wprowadzać opisy lub tłumaczenia w języku polskim. Również konieczne niekiedy umieszczenie znaków fonetycznych lub przyjętej transkrypcji może okazać się niewykonalne.

Na rynku oprogramowania komercyjnego możemy wprawdzie znaleźć edytory, które w mniejszym lub większym stopniu mogą zaspokoić potrzeby orientalistów, jednak z takim oprogramowaniem łączą się zasadniczo dwie cechy. Po pierwsze są to często drogie programy, a przy zamiarze instalacji na kilka komputerów należy liczyć się ze zwielokrotnieniem kosztów. Po drugie zaś tylko producent może rozszerzyć ich możliwości edycyjne, a także wskazać platformę sprzętową oraz wersję językową systemu operacyjnego na jakich mogą być uruchamiane — nie mają więc charakteru otwartego.

Edytorem, który natomiast śmiało można polecić każdemu koreaniście jest Emacs. Nie jest to wyrób komercyjny, a jednak posiada możliwości, które stawiają go w czołówce oprogramowania o podobnym charakterze, zostawiając w tyle większość programów komercyjnych. Jeżeli w pracy konieczna jest edycja tekstów tak orientalnych jak i europejskich, a owa wielojęzyczność musi być zapewniona w jednym dokumencie, to Emacs jest w stanie poradzić sobie nawet z najtrudniejszym zadaniem.

2 Oprogramowanie wolnodostępne i Ogólna Licencja Publiczna GNU

Zanim bliżej poznamy edytor Emacs, warto na chwilę zatrzymać się przy idei oprogramowania wolnodostępnego¹ (*free software*), którego przedstawicielem jest właśnie Emacs. To chyba dzięki tej idei Emacs odniósł tak wielki sukces i rozwinął się do programu, który jest nie tylko wszechstronnym edytorem tekstów w językach naturalnych, ale również — czy może przede wszystkim — platformą do pisania programów w różnych językach programowania oraz do komunikacji z Internetem, a liczba różnorodnych modułów rozszerzających jego zakres zastosowania jest wręcz trudna do dokładnego oszacowania.

Koncepcja oprogramowania wolnodostępnego sięga początku lat 80-tych, kiedy to młody amerykański naukowiec, Richard M. Stallman, powołał do życia nieformalny ruch GNU Project, którego celem było utworzenie darmowej wersji systemu Unix. Punktem wyjścia było określenie takiej licencji na oprogramowanie, żeby wszyscy, którzy je otrzymają mieli prawo do uruchamiania, modyfikowania, redystrybucji i wnoszenia własnych dodatków. Efektem tych prac było utworzenie Ogólnej Licencji Publicznej GNU (*GNU General Public Licence*). Licencja ta posiada cechy zupełnie kontrastujące z typowymi licencjami programów komercyjnych, które po pierwsze służą

¹Nie ma obecnie oficjalnie przyjętego czy jednoznacznie tłumaczonego zestawu terminów, którym posługuję się w tym artykule. Tłumaczenie większości terminów dotyczących licencji i rodzajów oprogramowania jest zgodna z [2].

zapewnieniu dochodów bezpośrednio z tytułu ochrony praw autorskich od każdego licencjobiorcy, a po drugie nie pozwalają na żadne modyfikacje.

Każde oprogramowanie posiada tzw. kod źródłowy², który w przypadku oprogramowania komercyjnego jest obwarowany licencjami i patentami przez wytwórców, a co najważniejsze jest niedostępny dla użytkownika. Wynika z tego, że producenci oprogramowania komercyjnego ograniczają dostęp do kodu źródłowego w celu uzyskiwania dochodów ze skompilowanych programów komputerowych. W takim przypadku nie ma żadnej możliwości, aby inna osoba lub firma mogła samodzielnie, bez łamania praw licencyjnych czy autorskich modyfikować lub poprawiać takie oprogramowanie.

Ponieważ oprogramowanie wolnodostępne (tj. dostępne w postaci „open source” – ogólnie dostępnego kodu źródłowego) może być dowolnie modyfikowane w ramach Ogólnej Licencji Publicznej GNU, to chronione są prawa autorskie, ale oprogramowanie nie jest źródłem dochodów z tego tytułu. Źródłem takim staje zasadniczo doradztwo i obsługa klienta.

Organizacją, która patronuje rozwojowi oprogramowania wolnodostępnego jest Fundacja Oprogramowania Wolnodostępnego (*Free Software Foundation*), która jest zapleczem i centrum koordynacyjnym dla prac nad oprogramowaniem przechodzącym ciągle usprawnienia zaproponowane przez programistów na całym świecie. W ten sposób różnego rodzaju oprogramowanie objęte Ogólną Licencją Publiczną GNU jest dostępne bez ograniczeń i nieodpłatnie w sieci Internet.

3 Inne zalety edytora Emacs

Ponieważ Emacs jest dostępny w postaci kodu źródłowego, możliwe jest skompilowanie tego kodu do postaci wykonywalnej na różnych komputerach pracujących pod kontrolą różnych systemów operacyjnych. Zachowane zostają przy tym wszystkie funkcje edytora, które nie są bezpośrednio uzależnione od danego systemu operacyjnego, w związku z czym praca na Emacsie zasadniczo nie różni się na komputerach typu PC z systemem Windows 95, 98, ME, NT, 2000, podobnie rzecz ma się z Linuxem czy dowolnym Unixem, jak i z systemami na komputerach firmy Apple. Gotowe wersje uruchamialne (skompilowane) można znaleźć w internecie. Szczególnie godnym polecenia dla orientalistów pracujących na komputerach pod kontrolą Windows 9x, nt

²Kod źródłowy to w uproszczeniu program napisany przy użyciu jakiegoś języka programowania wyższego poziomu, np. C, Perła itp. Tak zapisany program nie może być jednak uruchomiony bezpośrednio na komputerze. Wersja wykonywalna programu jest otrzymywana przez kompilację kodu źródłowego. Jednakże otrzymany w ten sposób zapis binarny staje się praktycznie nieczytelny dla programisty.

jest płyta WNPTW (Wybrane Narzędzia do Przetwarzania Tekstów Wielojęzycznych) stworzona w Zakładzie Zastosowań Informatycznych IOUW przez prof. UW, dr hab. Janusza S. Bienia, która została przygotowana specjalnie w celu ułatwienia edycji tekstów orientalnych.

Pewnego rodzaju konsekwencją jednakowego działania Emacsa na różnych platformach systemowych jest jednakowy sposób zapisu danych bez względu na rodzaj tej platformy. Co więcej, w przeciwieństwie do wielu programów komercyjnych, nie ma problemu z odczytywaniem dokumentów tworzonych pod starszymi wersjami Emacsa. Tak więc jeżeli np. jeden z użytkowników pracuje na komputerze z systemem operacyjnym Windows 95 i przekazuje swój dokument stworzony pod Emacsem innemu użytkownikowi Emacsa, który pracuje na komputerze pod kontrolą Linuxa, to dokument będzie wyglądał tak samo na obu urządzeniach. Jest to niezwykle ważne np. przy wspólnej pracy nad tworzeniem jakiejś publikacji lub jej późniejszej korekcie.

Ogromna większość współczesnych programów komercyjnych ma bardzo duże wymagania sprzętowe w odniesieniu do mocy obliczeniowej i pamięci RAM. Na twardym dysku musi być również zarezerwowana coraz rozleglejsza ilość miejsca na nawet najskromniejszą opcję instalacyjną, gdyż użytkownik w zasadzie ma bardzo ograniczony dostęp do indywidualnego wyboru potrzebnych mu narzędzi. Emacs jest programem, który powstawał w czasie, gdy użytkownik pracował zwykle na terminalu ze zdalnym połączeniem do serwera, na którym uruchomiony był Emacs. Ten rodzaj pracy w połączeniu z dość ograniczoną przepustowością połączeń sieciowych w tamtych czasach wymusił na projektantach daleko posuniętą dbałość o maksymalne wykorzystanie potencjału ówczesnych urządzeń komputerowych. Korzyści z tego tytułu są zauważalne również i dziś. Do stosunkowo wygodnej pracy z Emacsem nie musimy dokonywać kosztownego zakupu najnowszych komputerów, możemy pracować na kilkuletnich, dużo wolniejszych maszynach. Nawet na takich „antykach” jak komputery z procesorem 386 i 8MB pamięci RAM możliwa jest jeszcze praca z Emacsem³. Należy zauważyć, że nie jest to małe znaczące atut, gdyż - przynajmniej w sytuacji koreanistów - większość komputerów to właśnie przestarzałe urządzenia z procesorem 486. Podstawowa dystrybucja Emacsa, mimo że wymaga dość dużego wolnego obszaru pamięci na twardym dysku, może być znacznie okrojona, gdy nie zamierzamy korzystać np. z rozszerzeń wspomagających programowanie. Z drugiej strony wielkość tej dystrybucji i tak nie jest współmierna z dystrybucjami wielu edytorów komercyjnych, których na dodatek zwykle nie da się skutecznie

³ograniczeniem przy wyświetlaniu odpowiednich fontów może być jednak zainstalowany system operacyjny

odchudzić.

Zwykle orientalista w swojej pracy naukowej tworzy dokumenty, w których posługuje się językiem ojczystym, językiem swojej specjalności oraz jego zapisem transkrypcyjnym czy fonetycznym. Niemniej w wielu przypadkach konieczne jest rozszerzenie dostępnych narzędzi o możliwość edytowania innych alfabetów niż te, których zwykle używa. Często jednak edytory tekstu, nawet te bardzo rozbudowane, posiadają ograniczony zbiór edytowanych znaków, który niestety nie można w prosty sposób rozszerzać. Ograniczenia, które się tu pojawiają są dwojakiego rodzaju. Po pierwsze licencja, która umożliwia legalne korzystanie z danego oprogramowania z reguły nie pozwala na własne jego modyfikacje. Dotyczy to szczególnie oprogramowania komercyjnego, ale również i programów z licencją typu *shareware*⁴. Po drugie, nawet w przypadku gdy istnieje techniczna możliwość zainstalowania i wykorzystania dodatkowych fontów, należy zauważyć, że ich przeważająca większość jest typu komercyjnego i może być legalnie używana dopiero po wykupieniu licencji na ich używanie, a która równocześnie nie pozwala na swobodne modyfikacje tych fontów (np. w celu dorobienia znaków diakrytycznych). W przypadku Emacsa, ze względu na to, że jest dystrybuowany na zasadach wspomnianej wcześniej Ogólnej Licencji Publicznej GNU, nie występuje ograniczenie modyfikacji oprogramowania. Z tego względu liczna grupa entuzjastów tego programu opracowała szereg uzupełnień pozwalających na pracę z praktycznie dowolnym językiem naturalnym. Jeżeli okaże się, że w danej instalacji Emacsa nie możemy posługiwać się określonym językiem, to z dużym prawdopodobieństwem konieczne dodatki znajdziemy w Internecie na jednym z archiwów *oprogramowania publicznego*⁵. W skrajnym przypadku, gdyby dane rozszerzenie nie było opracowane, zawsze można zlecić taką pracę programiście. Również zdobycie odpowiednich fontów nie powinno być sprawą skomplikowaną. W wyżej wspomnianych archiwach oprogramowania publicznego można znaleźć zestaw bezpłatnych fontów, tzw. *International Fonts*, które współpracują z Emacsem.⁶

⁴*Shareware* to rodzaj licencji na oprogramowanie, która pozwala na jego bezpłatne korzystanie do celów testowych przez określony w licencji okres czasu, po którym konieczne jest wykupienie licencji do dalszego legalnego użytkowania

⁵czym jest oprogramowanie publiczne

⁶*International Fonts* znajdują się m.in. na serwerze ftp [sunsite.icm.edu.pl](ftp://sunsite.icm.edu.pl) w katalogu `/pub/gnu/intlfonts`. Na zestaw ten składają się fonty typu bdf, znane z X-window w systemach typu unix.

4 Emacs i metody wejściowe

Emacs, jako edytor tekstowy, umożliwia posługiwanie się międzynarodowym zestawem znaków, w skład których wchodzi nie tylko języki europejskie, ale również najróżniejsze alfabety spotykane na całym świecie. Możemy więc pisać po polsku, po francusku, rosyjsku, ale również i w języku japońskim czy koreańskim. Możliwość ta nie jest ograniczona do oddzielnych dokumentów, możemy w dowolnym momencie „przesiąć się” z jednego języka na inny, np. z polskiego na koreański.

W niektórych edytorach istnieje możliwość pisania w językach innych niż podstawowy język edytora, ale bywa to okupione istotnymi ograniczeniami i niewygodą we wprowadzaniu dodatkowych znaków. Jest to dość częste zjawisko, kiedy korzystamy z edytorów nie przystosowanych do pracy z językiem polskim. Niekiedy znaki diakrytyczne mogą być wprowadzane tylko przez podanie stosownego kodu lub wybór z tabeli. Ale nawet jeśli znaki narodowe dostępne są bezpośrednio z klawiatury to zdarza się, że ich układ nie jest zgodny z przyjętym ogólnie standardem. Niektóre programy umożliwiają natomiast zdefiniowanie własnego układu klawiatury, chociaż nie zawsze pozwalają przy tym na wykorzystanie układów rozszerzonych, jakich wymaga np. powszechnie przyjęta *polska klawiatura maszynistki*, która do wprowadzania polskich znaków diakrytycznych wykorzystuje tzw. *prawy Alt*. Z takim właśnie ograniczeniem spotykają się koreańscy użytkownicy korzystający ze stosunkowo popularnego programu *Hangul Word Processor*⁷.

W przypadku edytora Emacs wprowadzanie z klawiatury znaków narodowych oraz innych znaków specjalnych, często charakterystycznych dla typografii stosowanej w danym kraju, jak np. używane w Korei odmiany nawiasów [『 』] lub [【 】], ściśle łączy się z koncepcją tzw. metod wejściowych (*input methods*). Można powiedzieć, że metoda wejściowa to pewien program, którego zadaniem jest przechwycenie informacji o wciśniętych klawiszach na klawiaturze, jej przetworzenie zgodnie z definicją danej metody i przesłanie wyniku do programu aplikacji, czyli w naszym przypadku do edytora. Z punktu widzenia użytkownika oznacza to, że pisanie w danym języku jest możliwe po ustawieniu metody wejściowej dla tego języka. Wielojęzyczna praca w edytorze Emacs polega więc na włączeniu metody wejściowej odpowiedniej dla danego języka i przełączenia na inną w chwili, gdy następuje zmiana edytowanego języka.

Cenną zaletą takiego rozwiązania jest niezależność metody wejściowej od programu, który wykorzystuje kierowaną do niego informację. Ograniczony

⁷*Hangul Word Processor* to komercyjny edytor firmy Hansoft z Korei. Dla koreanisty pracującego w Polsce jego główną zaletą jest możliwość instalacji w polskojęzycznym środowisku Windows.

jedynie liczbą stosowanych konwencji wprowadzania tekstu w różnych krajach zestaw metod wejściowych może być udostępniany w postaci biblioteki dołączanej do edytora. Zestaw metod wejściowych współpracujących z edytorem Emacs nosi nazwę *LEIM* od *Library of Emacs Input Methods* i jest dostępny na serwerach z archiwami oprogramowania publicznego. Dzięki mechanizmowi metod wejściowych możliwe było zachowanie przyjętych w danym kraju czy alfabecie sposobów wprowadzania znaków. Często jest kilka — to użytkownik decyduje, która forma jest dla niego najwygodniejsza przez podanie stosownej metody wejściowej, gdyż w ramach danego języka może funkcjonować kilka takich metod, co ma miejsce np. w języku koreańskim. Niekiedy zaś taka sama metoda wejściowa może służyć do posługiwania się kilkoma językami, jak np. metody wejściowe z rodziny latin-2, dzięki której dostępne są narodowe znaki polskie, ale również i kilku innych krajów słowiańskich. Lista metod wejściowych dostępnych w bibliotece *LEIM* jest podana w załączniku.

Należy tu dodać, że tradycyjną metodą wprowadzania polskich znaków diakrytycznych było korzystanie z metod wejściowych latin-2-prefix oraz latin-2-postfix, w których znaki narodowe, takie jak np. polskie [ą] czy [ź] tworzone są przez zestawienie litery podstawowej i „ogonka”, odpowiednio prefiksowo lub postfiksowo, a więc dla latin-2-postfix [ą] otrzymujemy przez kolejne wciśnięcie [a], a następnie [,], natomiast [ź] wprowadzamy przez kolejne naciśnięcie [z] i [']. Z uwagi na upowszechnienie się metody wprowadzania polskich diakrytyków w sposób stosowany w MS Windows, czyli za pomocą tzw. klawiatury maszynistki, mechanizm ten zaimplementowano również w Emacsie. Aby wybrać polskie znaki należy jednocześnie wciśnąć prawy klawisz Alt oraz tę literę z klawiatury, która ma otrzymać „ogonek”.

Zależnie od struktury danego języka, metody wejściowe można podzielić na kilka kategorii, które różnią stopniem skomplikowania i mechanizmem otrzymywania znaków danego alfabetu.

W najprostszej postaci znakom danego alfabetu są przypisane znaki dostępne na klawiaturze. Oznacza to, że wybór określonego znaku dokonywany jest przez wciśnięcie z góry zdefiniowanego klawisza standardowej klawiatury ASCII. Tak tworzone są metody wejściowe dla języków greckiego i rosyjskiego. Przypisanie może być dokonane na bazie podobieństw fonetycznych (np. w metodzie wejściowej cyrillic-yawerty), ale zdarza się często, że jest poddyktowane konwencjami utrwalonymi historycznie w dobie kształtowania się narodowych układów klawiatury i wtedy powiązania ze znakami ASCII są dość przypadkowe (np. cyrillic-jcuken). W przypadku alfabetów, w których liczba liter jest większa niż liczba liter na standardowej klawiaturze ASCII, pewnym znakom muszą być przypisane klawisze, które w ASCII służą do wprowadzania znaków nieliterowych.

Kolejną z kategorii jest kompozycja. Tutaj znaki narodowe, takie jak np. polskie [ą] czy [ź] tworzone są przez zestawienie litery podstawowej i odpowiedniego „ogonka”, jak we wspomnianej już wyżej metodzie wejściowej latin-2-postfix. Dużym ułatwieniem w przypadku tego typu metod wejściowych jest stosowanie mechanizmu podpowiedzi w oknie linii stanu edytora. Jeżeli np. wciśniemy literę [a] otrzymujemy informację, że istnieje możliwość kompozycji w połączeniu z następującymi znakami: [’ ’ ’ , ^]. Gdy naciśnięty zostanie przecinek, [a] zamienia się w [ą]. Pewna niedogodność wynikająca ze stosowania tej metody wejściowej może być zauważona, gdy para znaków wykorzystywana jako kombinacja ma wystąpić samodzielnie, np. gdy po słowie kończącym się na [e] ma wystąpić przecinek jako znak interpunkcyjny. Niedogodności tej nie ma w metodzie prefiksowej, ale tu trzeba pamiętać jaki prefiks służy do otrzymania właściwego „ogonka”, np. kombinacja [’] i [e] daje [é], a [‘] i [e] spowoduje powstanie [ē].

Jeszcze bardziej skomplikowaną kategorię tworzą metody wejściowe dla alfabetów sylabicznych, takich jak koreański hangŭl. W alfabecie koreańskim posługujemy się literami, które zostają następnie złożone w sylaby. Słowo „złożone” jest tu chyba najbardziej odpowiednie, gdyż nie chodzi tu tylko o formalne zdefiniowanie sylaby, ale również i konstrukcję jej układu graficznego. Układ ten podlega jasno określonym prawom, ale co jest najistotniejsze dla obróbki tekstu, nie jest on linearny. Oznacza to, że poszczególne litery tworzące sylabę mogą być rozmieszczone zarówno jedna za drugą, jaki i jedna nad drugą, a także obie te kombinacje mogą występować równocześnie.

Przy edycji znaków alfabetu hangŭl mamy do dyspozycji dwie metody wejściowe. Pierwsza z nich o nazwie korean-hangul obsługuje najczęściej spotykany w Korei układ klawiatury 한글 2벌식 (*hangŭl i pŏlshik*), który jest zgodny z normą KS X 5002:1992. W układzie tym literom alfabetu przypisane są określone klawisze literowe klawiatury standardowej, ale spółgłoski rozpoczynające i kończące sylabę nie są rozróżniane. Zapisując po koreańsku słowo „Korea” — 한국 (*hanguk*), wpisujemy kolejno ciąg znaków z klawiatury: **gksrnr**, w którym poszczególnym literom odpowiadają kolejno znaki alfabetu hangŭl: **ㅎ ㄱ ㅌ ㄴ ㄱ ㄹ**. Podobnie jak w przypadku opisywanych wcześniej metod latin-2, w trakcie pisania dostępny jest system podpowiedzi, wskazujący możliwe dopełnienia. System ten jest tu jednak bardziej rozbudowany, gdyż wciśnięcie klawisza [TAB] otwiera dodatkową ramkę z pełną listą dopełnień wraz z ich graficzną interpretacją w alfabecie *hangŭl*. Drugą z metod wejściowych do pisania w hangŭlu jest korean-hangul3. W metodzie tej wykorzystywany jest zupełnie inny układ klawiatury, określany jako 한글 3벌식 (*hangŭl sam pŏlshik*). W tym przypadku rozróżniane są trzy pozycje w sylabie koreańskiej: 초성 (*ch’osŏng*) — spółgłoska rozpoczynająca, 중성 (*chungsŏng*) — samogłoska oraz 종성 (*chongsŏng*) — spółgłoska lub spół-

głoski kończące sylabę. Zapis słowa „Korea” teraz będzie wymagał innego ciągu znaków z klawiatury: `mfskbx`. Ze względu na rozróżnienie spółgłosek rozpoczynających i kończących sylabę, większej liczbie znaków alfabetu koreańskiego muszą być przypisane klawisze standardowej klawiatury, wykraczając poza klawisze literowe. Również dla tej metody wejściowej funkcjonuje system dopełnień możliwych kombinacji liter w celu utworzenia poprawnej sylaby koreańskiej.

Jeżeli nie dysponujemy klawiaturą z naniesionym układem znaków koreańskim w dowolnej z wymienionych postaci wybór odpowiednich znaków może wydawać się dość kłopotliwy. Jednak i w tym przypadku możemy otrzymać stosowną odpowiedź. Po wybraniu żądanej metody wejściowej należy zastosować komendę `Describe-Input-Method` i zaakceptować metodę podaną jako *default*. W nowo otwartej ramce otrzymamy wyczerpującą informację tak o układzie klawiatury, jak i dodatkowych możliwościach danej metody.

Najbardziej złożone metody dotyczą języków japońskiego i chińskiego, a także koreańskiego w przypadku, gdy wprowadzamy ideogramy chińskie. Rozpatrzmy metody wejściowe dla języka koreańskiego. Możliwe są następujące wybory: `korean-hanja`, `korean-hanja3` oraz `korean-hanja-jis`. Metoda wejściowa `korean-hanja` służy do wprowadzania ideogramów poprzez konwersję znaków alfabetu `hangul` zapisywanych w konwencji metody `korean-hangul`, natomiast `korean-hanja3`, analogicznie, w konwencji metody `korean-hangul3`. W trakcie wpisywania kolejnych znaków z klawiatury również i tutaj funkcjonuje system dostępnych dopełnień (również pełna lista wraz z konwersją na odpowiednie ideogramy chińskie — za pomocą klawisza [TAB]). W momencie gdy kombinacja liter utworzy sylabę, dla której w wewnętrznym słowniku skojarzone są ideogramy chińskie, ideogramy te zostają wyświetlone wraz z możliwością wyboru jednego ze znaków. Kolejne modyfikacje sylaby powodują wyświetlenie aktualnego zestawu dostępnych ideogramów chińskich. Prześledźmy to na przykładzie pierwszej sylaby słowa `한국` (*hanguk*) — „Korea” dla metody `korean-hanja`. Po wciśnięciu [g] otrzymujemy jedynie listę dostępnych dopełnień, gdyż spółgłoska `ㅎ` (*h*) nie tworzy jeszcze żadnej sylaby. Wprowadzenie samogłoski `ㅏ` (*a*) klawiszem [k] powoduje umieszczenie informacji o możliwych dopełnieniach i równocześnie listę ideogramów chińskich dla sylaby `하` (*ha*): 下, 何, 厦, 夏, 廈, 晷, 河, 瑕, 荷, 蝦, 賀, 遐, 霞, 鰕. Wpisanie kolejnej spółgłoski `ㄴ` (*n*) — klawisz `s` — spowoduje wyświetlenie innego zestawu ideogramów, tym razem odpowiadających sylabie `한` (*han*): 閑, 閒, 限, 韓. Ponieważ poszukiwaliśmy ostatniego z tej listy, ustawiamy na nim kursor, co automatycznie spowoduje, że analogiczny ideogram pojawi się w ramce edytora. Kontynuując edycję dokonujemy akceptacji wybranego znaku.

Metoda wejściowa `korean-hanja-jis` funkcjonuje identycznie jak `korean-`

hanja, jednakże tym razem wyświetlane są ideogramy w typografii japońskiej oraz możliwy jest wybór znaków uproszczonych. Porównajmy jeszcze raz słowo „Korea” — 한국 *hanguk* w obu metodach. W metodzie korean-hanja otrzymujemy: 韓國, natomiast w metodzie korean-hanja-jis: 韓國, przy czym dostępna jest również wersja uproszczona: 韩国.

Ostatnią z metod wejściowych dla języka koreańskiego jest korean-symbol. Dzięki niej możemy wybierać znaki specjalne, często niedostępne przy pomocy pozostałych metod. Należy podać kategorię, do której należy poszukiwany znak, a następnie wybrać go z wyświetlonej na ekranie listy. Poniżej przedstawiony jest pełny wykaz kategorii z przykładowymi znakami:

```

【(】 괏호열기 【arrow】 화살 【sex】 ♂ ♀ 【index】 첨자 【accent】 악센트
【)】 괏호닫기 【music】 음악 【dot】 점 【quote】 따옴표 【xtext】 § ※ ¶ i ð
【Unit】 ° Å ¢ °F 【math】 수학기호 【pic】 상형문자 【line】 선문자
【unit】 단위 【frac】 분수 【textline】 - — || \ ~
【wn】 (췎) 【ks】 Ⓜ 【No】 No. 【Co.】 Co. 【dag】 † 【ddag】 ‡ 【percent】 %
【am】 am 【pm】 pm 【TM】 TM 【Tel】 Tel 【won】 ₩ 【yen】 ¥ 【pound】 £
【Eng】 A B C... 【enum】 0 1 2... 【Russ】 Б В Г... 【Greek】 Α Β Γ...
【eng】 a b c... 【easc】 영어ASCII 【russ】 а б в... 【greek】 α β γ...
【Rom】 I II III... 【Scan】 Đ đ Ħ ħ... 【hira】 あ い い 【rom】 i ii iii...
【scan】 đ đ ħ ħ... 【kata】 ア イ イ
【ojaso】 ㉠~㉡ 【pjaso】 ㉠~㉡ 【oeng】 ㉠~㉡ 【peng】 (a)~(z)
【ogana】 ㉠~㉡ 【pgana】 ㉠~㉡ 【onum】 ①~⑮ 【pnum】 (1)~(15)
【자소】 2별식 + ㄴ(S) △(t_) ○(D) ㅁ(DD) ㅂ(aD) ㅅ(_d) ㅇ(̄G) ·(uk)

```

Po tym krótkim opisie metod wejściowych należy nadmienić, że po zainstalowaniu Emacsa w wersji podstawowej nie jest dostępna żadna z nich. Aby uzyskać możliwość korzystania z metod wejściowych, konieczne jest dołączenie do Emacsa biblioteki *LEIM*. Aby zaś wprowadzone znaki były widoczne na ekranie monitora, należy zainstalować odpowiedni zestaw fontów, np. wspomniany wcześniej pakiet *International Fonts*, gdyż Emacs w podstawowej dystrybucji nie ma dołączonych fontów do edycji tekstów wielojęzycznych. Trzeba je więc zainstalować samodzielnie. Jednakże na płycie WNPTW Emacs jest od razu przygotowany do pracy wielojęzycznej i posiada zainstalowane odpowiednie zestawy fontów.

5 Środowiska językowe

Do edycji tekstów wielojęzycznych wystarczy w zasadzie wybranie odpowiedniej metody wejściowej. Jednakże przyjęcie środowiska językowego (*language environment*) jest o tyle istotne, że nadaje ono Emacsowi pewne wartości domyślne, które niejako preferują wybrany przez to środowisko język, co w

większości przypadków znacznie ułatwia użytkownikowi współpracę z edytorem, np. uwalniając go od ręcznego wprowadzania niezbędnych informacji o sposobie kodowania znaków przy odczycie lub zapisie plików lub poprzez zdefiniowanie skrótów klawiaturowych pomaga szybko uzyskać znaki specjalne, typowe dla danego języka. Nie musimy dokładnie pamiętać, jakie parametry są ustawiane poprzez wybór środowiska językowego. Żądanie opisu danego środowiska spowoduje wyświetlenie możliwych skrótów klawiaturowych, przykładowego tekstu, dostępnych metod wejściowych oraz domyślnego kodowania wykorzystywanego do odczytywania lub zapisu plików.

Weźmy pod uwagę środowisko językowe dla języka koreańskiego. Przy próbie wyboru metody wejściowej, otrzymujemy informację, że domyślnie wybraną będzie *korean-hangul*. Dzięki zdefiniowanym skrótom klawiaturowym istnieje bardzo prosty mechanizm wyboru wprowadzanych znaków: liter alfabetu hangŭl, ideogramów chińskich lub znaków specjalnych. Klawisz funkcyjny [F9] służy do przełączania między metodą wejściową *korean-hangul* oraz *korean-hanja*, natomiast kombinacja [Ctrl-F9] — między *korean-hangul* i *korean-symbol*, przy czym działają one w identyczny sposób niezależnie od tego jaka została wprowadzona wcześniej metoda wejściowa. Jest to bardzo wygodna cecha, gdy powracamy do edycji tekstu koreańskiego po chwilowo wybranej innej metodzie wejściowej.

Emacs ma zdefiniowane następujące środowiska językowe⁸: *Chinese-BIG5*, *Chinese-CNS*, *Chinese-GB*, *Cyrillic-Alternativnyj*, *Cyrillic-ISO*, *Cyrillic-KOI8*, *Devanagari*, *English*, *Ethiopic*, *Greek*, *Hebrew*, *Japanese*, *Korean*, *Lao*, *Latin-1*, *Latin-2*, *Latin-3*, *Latin-4*, *Latin-5*, *Thai*, *Tibetan*, *Vietnamese*

6 Systemy kodowania

Z zapisem tekstów wielojęzycznych w Emacsie wiąże się jeszcze jedna kwestia. Jest to system kodowania. Dla różnych języków ukształtowały się pewne sposoby kodowania znaków, z których wybrane stały się dla danych języków jedynym obowiązującym standardem, a w innych funkcjonuje ich kilka równoległe (np. w Polsce w środowisku Windows (Dos) o prymat walczą ISO 8859-2 oraz CP1250 (852)). Dodatkową komplikacją jest sposób zapisu końca linii (RET/CR) odmienny w środowiskach DOS, Mac czy Unix. Jest to tzw. *end-of-line conversion*.

Emacs oferuje bardzo szeroki wachlarz systemów kodowania. Jest to możliwe, gdyż dla własnych potrzeb dokonuje konwersji z danego systemu kodowania na swój własny, wewnętrzny sposób czytania danych oraz konwersji na

⁸Nazwy języków celowo zostały podane w oryginale dla zachowania zgodności z terminami komend Emacs'a.

dany system kodowania przy zapisie danych. Określając środowisko językowe zostaje domyślnie przyjęty jeden z systemów kodowania, charakterystyczny dla danego języka⁹. Oczywiście zawsze można ustawić, dostosowany do konkretnych potrzeb, inny system kodowania. W szczególności, gdy posługujemy się w jednym dokumencie wieloma językami, może zajść konieczność jego zapisu w wewnętrznym kodowaniu Emacsa, czyli *emacs-mule*. Niemniej, o ile nie zachodzi konieczność ustawienia specjalnego kodowania, do pracy w danym języku wystarczy określić środowisko językowe oraz metodę wejściową, a system kodowania można pozostawić w sposób domyślnie przyjęty przez program.

W środowisku języka koreańskiego dostępne są następujące systemy kodowania:

korean-iso-8bit-with-esc — obejmuje zestaw znaków koreańskich KS X 1001¹⁰. Jest podobny do *korean-iso-8bit*, ale może obsługiwać każdy z zestawów znaków poprzez tzw. *ISO's escape sequences*.

korean-iso-8bit, **euc-kr**, **euc-korea** — obejmuje zestaw znaków koreańskich KS X 1001. Jest to kodowanie EUC dla koreańskiego zestawu znaków bazowane na standardzie ISO 2022 (oznaczenie MIME: *EUC-KR*).

iso-2022-kr, **korean-iso-7bit-lock** — obejmuje zestaw znaków koreańskich KS X 1001. Jest to 7-bitowe kodowanie ISO 2022 dla koreańskiego zestawu znaków (oznaczenie MIME: *ISO-2022-KR*).

iso-2022-7bit-lock-ss2, **iso-2022-cjk** — obejmuje zestawy znaków: koreański KS X 1001, chiński (kontynentalny) GB 2312, chiński (Tajwan) CNS 11643-1÷7. Jest to kodowanie będące połączeniem *ISO-2022-JP*, *ISO-2022-KR* i *ISO-2022-CN*.

iso-2022-jp-2 — obejmuje zestawy znaków: japoński JIS X 0208-1978, JIS X 0208, łaciński z JIS X 0201, japoński JIS X 0212, katakana z JIS X 0201, chiński GB 2312, koreański KS X 1001, łaciński ISO 8859-1, grecki ISO 8859-7). Jest to 7-bitowe kodowanie ISO2022 dla CJK, Latin-1 oraz greckiego (oznaczenie MIME: *ISO-2022-JP-2*).

⁹Właściwie jest to lista systemów kodowania z zachowaniem ustalonego priorytetu użycia.

¹⁰Zestaw znaków określonych normą KS X 1001:1992 (dawniej KS C 5601) obejmuje 2350 możliwych układów samogłosek i spółgłosek w alfabecie hangŭl, 4888 ideogramów chińskich oraz 986 symboli.

7 Podsumowanie

Ze względu na wręcz ogromne możliwości edycyjne Emacs może być zaliczony do grona najbardziej wszechstronnych programów tego typu. Dla orientalisty niezaprzeczalnym jego atutem jest opisana wyżej wielojęzyczność, atrybut, którego nie posiada wiele dostępnych edytorów. Wręcz nasuwa się pytanie — dlaczego tak uniwersalny program nie jest dostatecznie popularny?

Po pierwsze Emacs jako program niekomercyjny nie jest objęty działaniami marketingowymi czy akcjami reklamowymi. Często przypadek lub uporcewywe poszukiwania pomagają odkryć to potężne narzędzie, szczególnie, gdy użytkownik nigdy nie miał okazji pracować na komputerze z systemem Unix czy Linux.

Edytory należące do oprogramowania wolnodostępnego mają również i swoje wady. Ponieważ nie powstają jako oprogramowanie komercyjne, ale są dziełem współpracy wielu niezależnych programistów i wynikiem doświadczeń ich użytkowników na całym świecie, z których każdy może mieć wpływ na modyfikację bądź dopisanie kodu źródłowego, główny nacisk nie jest położony na prostotę ich wykorzystania, lecz funkcjonalność. Kupując program znanych potęg software'owych, jesteśmy często prowadzeni za rękę w spotkaniu z komputerem, choć rzadko możemy się z tej opieki uwolnić. Oprogramowanie, do którego należy również Emacs, niewątpliwie jest trudniejsze do opanowania, wymaga pewnej wiedzy i doświadczenia w pracy z komputerem. Emacs nie posiada również tak bogatej listy publikacji i podręczników jak znane edytory komercyjne. Z drugiej strony Emacs jest stale ulepszany i modernizowany. Zmiany, które zachodzą, zbliżają go coraz bardziej do użytkownika, który nie tracąc czasu na poznawanie komputerowych tajemnic, potrzebuje wydajnych narzędzi do swojej pracy.

Wydaje się jednak, że największą barierą w poznawaniu nowego, w tym również programów komputerowych są stare przyzwyczajenia, które nie pozwalają nam rozstać się z wykorzystywanymi dotychczas narzędziami. Mam jednak nadzieję, że ten krótki opis stanie się zachętą do poznania edytora Emacs i innych niezwykle pomocnych narzędzi, należących do oprogramowania wolnodostępnego, jak program do profesjonalnego składu tekstów — TeX. Zapowiadana jest również kolejna, przełomowa wersja edytora Emacs, która z pewnością przyniesie szereg nowości i udogodnień. Na zakończenie pragnę jeszcze raz zachęcić do samodzielnych prób z Emacsem, wykorzystując dostosowaną do potrzeb orientalistów, wspomnianą wcześniej płytę WNPTW.

8 Bibliografia

1. Cameron Debra, Rosenblatt Bill, Raymond Eric, „Learning GNU Emacs” II wyd., O’Reilly, 1996
2. Deuch L. Peter, „Prawo i oprogramowanie”, tłum. Ryszard Kubiak, zeszyt 11 GUST, grudzień 1998
3. Finseth C.A., „The Craft of Text Editing. Emacs for the Modern World”, Springer-Verlag, 1991
4. Lunde Ken, „CJKV Information Processing, O’Reilly, 1999
5. Pedersen Jesper, „Sams Teach Yourself Emacs in 24 Hours”, Sams, 1999

9 Dodatek

Na zakończenie przedstawiam pełną listę dostępnych metod wejściowych z biblioteki *leim*:

british	catalan-prefix
chinese-b5-quick	chinese-b5-tsangchi
chinese-cns-quick	chinese-cns-tsangchi
chinese-py-punct	chinese-py-punct-b5
cyrillic-beylorussian	cyrillic-jcuken
cyrillic-jis-russian	cyrillic-macedonian
cyrillic-serbian	cyrillic-translit
cyrillic-translit-bulgarian	cyrillic-ukrainian
cyrillic-yawerty	czech
czech-prog-1	czech-prog-2
czech-prog-3	czech-qwerty
danish-alt-postfix	danish-keyboard
danish-postfix	devanagari-hindi-transliteration
devanagari-itrans	devanagari-keyboard-a
devanagari-transliteration	english-dvorak
esperanto-alt-postfix	esperanto-postfix
esperanto-prefix	ethiopic
finish-keyboard	finnish-alt-postfix
finnish-postfix	french-alt-postfix
french-azerty	french-keyboard
french-postfix	french-prefix
german	german-alt-postfix
german-postfix	german-prefix
greek	greek-jis
hebrew	icelandic-alt-postfix
icelandic-keyboard	icelandic-postfix
ipa	irish-prefix
italian-alt-postfix	italian-keyboard
italian-postfix	japanese
japanese-ascii	japanese-hankaku-kana
japanese-hiragana	japanese-katakana
japanese-zenkaku	korean-hangul
korean-hangul3	korean-hanja
korean-hanja-jis	korean-hanja3
korean-symbol	lao
lao-lrt	latin-1-alt-postfix
latin-1-postfix	latin-1-prefix
latin-2-alt-postfix	latin-2-postfix
latin-2-prefix	latin-3-alt-postfix
latin-3-postfix	latin-3-prefix
latin-4-alt-postfix	latin-4-postfix
latin-5-alt-postfix	latin-5-postfix
norwegian-alt-postfix	norwegian-keyboard
norwegian-postfix	portuguese-prefix

scandinavian-alt-postfix
slovak
slovak-prog-2
spanish-alt-postfix
spanish-postfix
swedish-alt-postfix
swedish-postfix
thai-pattachote
tibetan-wylie
turkish-postfix

scandinavian-postfix
slovak-prog-1
slovak-prog-3
spanish-keyboard
spanish-prefix
swedish-keyboard
thai-kesmanee
tibetan-tibkey
turkish-alt-postfix
vietnamese-viqr

10 Uwagi do wersji elektronicznej

Niniejszy artykuł stanowi tekst referatu wygłoszonego 11 grudnia 1999 roku na seminarium koreanistycznym zorganizowanym przez Zakład Japonistyki i Koreanistyki Instytutu Orientalistycznego Uniwersytetu Warszawskiego. Jest on złożony do druku w znajdującej się w przygotowaniu publikacji

Halina Ogarek-Czój, Romuald Huszcza (red.)

Studia Coreana Varsoviensia

Wydawnictwo Uniwersytetu Warszawskiego

Warszawa

s. 108–119

Wersja elektroniczna — w formacie Postscript (MP-sk99.ps) i PDF (MP-sk.pdf)
— jest dostępna pod adresem

http://www.orient.uw.edu.pl/~zzi/publikacje/MP-sk99.*

oraz (od maja 2003 r.) pod adresami

http://www.orient.uw.edu.pl/~jsbien/MP/MP-sk99.*

http://www.mimuw.edu.pl/~jsbien/MP/MP-sk99.*