TEX i Linux – integracja a integralność dwu światów

Maciej Jan Głowacki
ul. Abrahama 20/15 81-352 Gdynia
gmj@amg.gda.pl

Pracę zgłosił: Tomasz Przechlewski

Abstract

Linux, since its early days, contains TEX software (more precisely: Thomas Esser's tetex compilation). Nevertheless, the embedding of TEXin Linux environment is far from perfect. The article describes the advantages and drawbacks of the present state of the art and proposes remedies to some of problems. Several methods and aspects of software installation and configuration are considered. Among others, the following issues are discussed: the possibility of a selective installation, the consistency with standards and customs, the problems of splitting the software into packages, and the sharing of font resources.

Założenia wstępne

Wypowiadając się krytycznie na jakiś temat, można usłyszeć porady w rodzaju: „zamiast krytykować – zrób lepiej". Jednak ci sami doradcy zapewne oburzyliby się, gdyby jakiś lekarz przystąpił do kuracji nie postawiwszy wpierw diagnozy. Krytyka jakiegoś rozwiązania (nie będąca oczywiście krytyką Autora), szczególnie gdy jest konstruktywna, stanowi – tak sądzę – konieczną diagnozę stanu a nawet sugestię co do kierunku terapii. Trzeba przejść przez ten etap przed rozpoczęciem modyfikacji istniejącego lub budowaniem alternatywnego rozwiązania.

Omawiając styk tytułowych dwu światów należy pamiętać o istniejącej i istotnej różnicy między nimi: system operacyjny wciąż pozostaje niepomijalną podstawą działania jakiegokolwiek programu, choćby największego, najbardziej rozbudowanego. To **są** dwa integralne światy i **jest** postępująca integracja między nimi, ale na zasadzie **osadzenia** jednego w drugim.

Ważne jest także uzmysłowienie sobie różnicy między użytkownikiem programu a administratorem systemu operacyjnego: pierwszy – z definicji – używa oprogramowania, które przygotował (zainstalował, skonfigurował, zaktualizował) ten drugi. Co prawda w przypadku tzw. stacji roboczej jest to zazwyczaj ten sam człowiek i zwykle jest on nazywany po prostu użytkownikiem, ale nawet w tym przypadku wykonuje on dwie różne funkcje, wymagające różnych umiejętności, innych zakresów posiadanej wiedzy itd. W systemach unixowych dodatkowa różnica polega na tym, że prace administracyjne wykonuje się z reguły będąc głównym użytkownikiem czyli jako root, a użytkowe będąc dowolnym innym użytkownikiem – prac użytkowych wręcz nie powinno się wykonywać z prawami root-a choćby ze względów bezpieczeństwa.

Jeszcze jedna uwaga: poniższą analizę kieruję głównie do tych, którzy pracują z TEX-em na Linuxie, a którzy być może chcieliby w miarę swoich możliwości uczynić to środowisko pracy jeszcze lepszym niż jest obecnie. Zakładam ponadto, że odbiorcami tego tekstu mogą być także osoby, które choć trochę słyszały zarówno o TEX-u, jak i o Linuxie i których nie ma potrzeby przekonywać do tych produktów. Stosunkowo najmniejszą korzyść odniosą zapewne Ci, którzy chcieliby do Linuxa przenieść wszystkie swoje ulubione zabawki z dotychczas używanego środowiska, lub dla których Linux wciąż pozostaje jedynie darmową namiastką Powszechnie Uznawanego Systemu.

Specyfika Linuxa z punktu widzenia użytkownika TEX-a

Linux jest systemem operacyjnym, który w prawie każdej wersji dystrybucyjnej, a na pewno w każdej popularnej, zawiera wśród pakietów proponowanych do instalacji także pakiety z oprogramowaniem TEX-owym. Jest to cecha nie występująca w systemach innych niż unixowe. Dla potencjalnego użytkownika TEX-a oznacza to, iż jeśli chce zainstalować np. na swoim komputerze najprościej jak się da jakikolwiek system, tak żeby mieć też od razu działającego w tym kontekście TEX-a, to Linux jest **jedyną** możliwością (zależnie od dystrybucji może być potrzebna najwyżej wstępna konfiguracja). Nie trzeba wtedy szukać źródła oprogramowania TEX-owego w odpowiedniej wersji, czy martwić się o jego pomyślną instalację i prawidłową koegzystencję z systemem operacyjnym. Jest to rozwiązanie dojrzałe i należy je uznać za w pełni funkcjonalne.

Cechami Linuxa które mogą się okazać bardzo pomocne użytkownikowi TEX-a są jego elastyczność i łatwość konfiguracji. Są dystrybucje duże i małe, trywialne w instalacji i używaniu lub skomplikowane, ale za to znacznie bardziej konfigurowalne. Poszczególne składniki środowiska są zazwyczaj dostępne w kilku alternatywnych zestawach, co może być początkowo pewnym utrudnieniem, ale po dokonaniu właściwego wyboru znacznie wzrasta komfort pracy. Nade wszystko jednak cenna jest możliwość dotarcia do każdego zakątka systemu. Pliki konfiguracyjne, z reguły w postaci tekstowej, łatwej do bezpośredniej edycji, są zgromadzone w jednym katalogu (`/etc`). Nie ma tajemniczych i niedostępnych rejestrów. Wszelkie stosowane polecenia i formaty plików posiadają swoje opisy (`man`, `info`). Hierarchiczny system plików jest jednorodny i w pełni dostępny, a jednocześnie bezpieczny (z zachowaniem zadeklarowanych praw dostępu).

Jest jeszcze jeden aspekt godny uwagi TEX-nika przy porównywaniu systemów operacyjnych. Wiadomo, że pełne środowisko służące do pracy z TEX-em to nie tylko to, co jest dostarczane w dystrybucjach czy pakietach TEX-owych, ale także sporo oprogramowania narzędziowego i dodatkowego, jak edytory, programy do obróbki grafiki, konwertery, rozmaite narzędzia do automatyzacji powtarzalnych czynności, programy zarządzające wersjami plików itd. Tak się składa, że spośród tych gatunków większość oprogramowania cenionego przez TEX-ników wywodzi się z Linuxa, a nawet ze starszych systemów unixowych. Wiele takich programów wtórnie zostało zaadoptowanych do innych, niezwykle popularnych systemów i bywają tam wykorzystywane przez co światlejszych użytkowników, ale warto być świadomym faktu, że naturalnym, rodzimym (czy, jak mawiają moderniści: „natywnym") środowiskiem dla większej części tego oprogramowania pozostaje wciąż Linux i Unix.

Nie chcę tu wzniecać polemiki o wyższości jednego systemu nad drugim, gdyż jest to temat zbyt szeroki i wieloaspektowy. Natomiast zakładając, że istotnym aspektem przydatności systemu pozostaje możliwość wszechstronnej pracy z TEX-em, stawiam tezę, iż **Linux jest obecnie znacznie lepszy od alternatywnych systemów** i to dla każdego, także początkującego użytkownika, zarówno korzystającego z zasobów serwera na stacji klienckiej, jak i pracującego na autonomicznej stacji roboczej. Należy jednak mieć świadomość, że zarówno administracja Linuxa jak i produktywne wykorzystanie możliwości tego środowiska wymagają zmiany mentalności i podejścia bardziej świadomego niż w przypadku niezwykle popularnych, konkurencyjnych odpowiedników.

Alternatywne pakiety instalacyjne

Instalacja oprogramowania jest jak wiadomo warunkiem koniecznym, by móc wykorzystywać jakikolwiek program. Co więcej, instalacja musi przebiegać i zakończyć się prawidłowo. Nie jest to zadanie trywialne: współczesne oprogramowanie jest budowane wielowarstwowo, także z wykorzystaniem bibliotek współdzielonych. Mnóstwo elementów rozmaitego autorstwa musi ze sobą precyzyjnie współpracować. Prawidłową realizację tego zadania umożliwia dostarczanie i instalowanie oprogramowania w postaci pakietów.

Mówiąc w pewnym uproszczeniu pakiet – z punktu widzenia systemu operacyjnego – to taki plik, który zawiera jeden lub więcej programów, ewentualnie bibliotekę procedur (funkcji) wraz z koniecznymi dodatkami (pliki konfiguracyjne, opisy, programy pomocnicze wykorzystywane tylko w tym kontekście, informacje o sobie i zależnościach od innych pakietów itd). Tak więc oprogramowanie jest dzielone na mniejsze, lecz w miarę samodzielne i niezależne części (choćby po to, by można było instalować i uaktualniać tylko te pakiety, które mają być wykorzystywane).

Prawie wszystkie wypracowane w ostatnich latach formaty pakietów posiadają tę istotną cechę: spełniają w trakcie instalacji wszystkie istniejące zależności między różnymi elementami składowymi instalowanego oprogramowania. Nie jest to łatwe w sytuacji, gdy nowe wersje niektórych programów powstają co kilka tygodni, a zależności z wersji na wersję mogą ulegać zmianom. Pod adresem `http://www.kitenet.net/~joey/pkg-comp/` można znaleźć porównanie różnych formatów pakietów używanych w systemach Linux i Unix.

Potrzeba spełniania zależności nie tylko wymusiła ich uwzględnienie przy projektowaniu formatów; zaowocowała także dodatkową funkcjonalnością – dzięki temu można bezpiecznie przeprowadzać deinstalację, grupować zadania instalacyjne, weryfikować integralność całego systemu itd. W konsekwencji coraz większą popularność zdobywają dystrybucje Linuxa, które umożliwiają automatyczną lub półautomatyczną aktualizację całego zainstalowanego oprogramowania (lub tylko wybranych pakietów) przez sieć.

Pakiety RPM. Najbardziej rozpowszechnionym formatem jest RPM czyli RedHat Package Manager albo RPM Package Manager. Format ten stosuje ponad dwadzieścia różnych wersji systemów unixowych (`http://www.rpm.org/platforms/`) i większość dystrybucji linuxowych (67 spośród 119 wymienionych na `http://distrowatch.com/stats.p hp?section=packagemanagement`).

Pakiety DEB. Drugim co do popularności formatem pakietów jest rodzimy format dystrybucji Debian, DEB. Z zestawienia podanego powyżej wynika, iż jest on używany w 16/119 dystrybucjach.

Pakiety TGZ. Najprostszym, a jednocześnie ciągle popularnym formatem pakietów jest TGZ czyli `tar.gz` (pliki zarchiwizowane `tarem` i spakowane `gzipem`). Format ten jest używany między innymi w najstarszej dystrybucji, Slackware, którą w początkach, czyli od 1991 roku rozwijał twórca jądra i idei Linuxa, Linus Torvalds. Także do tego formatu istnieją narzędzia, ułatwiające instalację i aktualizację ze sprawdzaniem zależności międzypakietowych.

Konwersja formatów. Każdy z powyższych formatów (istnieją jeszcze inne, nie wymienione tutaj, o mniejszym znaczeniu) ma swoje zalety i niedoskonałości, gdyż nie stworzono jak dotąd uniwersalnego formatu pozbawionego wszelkich wad. Istnieje jednak możliwość konwersji dowolnego pakietu z jednego formatu na drugi przy pomocy programu `alien` (`http://kitenet.net/programs/alien/`).

Ignorowanie zależności. Niestety, część oprogramowania dla Linuxa, między innymi także `tetex`, nie jest oferowana przez Autorów w postaci umożliwiającej sprawdzanie zależności w trakcie instalacji lub aktualizacji. Najczęściej jest to niepodzielna „paczka” którą można zainstalować „jako taką”, np. jakimś dołączonym instalatorem, lecz ryzykując konflikt wersji z już zainstalowanym oprogramowaniem lub brak składnika, którego istnienie w systemie było domniemane. W takich sytuacjach twórcy poszczególnych dystrybucji Linuxa, chcąc włączyć taką paczkę do swojej oferty, przygotowują pakiety we własnym zakresie. Jest to jednak sytuacja niezdrowa, gdyż w każdej dystrybucji inaczej wygląda selekcja materiału, inaczej dokonywany jest podział, a w konsekwencji inaczej odczytywane są zależności.

Zaszłości historyczne związane z TEX-em

Generowanie formatów, będące wynikiem zastosowania technologii i uwzględnienia ograniczeń istniejących ćwierć wieku temu, nie jest obecnie nigdzie poza TEX-em wykorzystywane. W sytuacji gdy nawet jądro Linuxa najczęściej jest dystrybuowane w postaci skompilowanej, należy generowanie docelowych formatów TEX-owych uznać za anachronizm. Również „zaszywanie” w programie takich funkcji jak wzorce przenoszenia wyrazów jest zbędnym usztywnieniem.

Niektóre założenia dystrybucji tetex

Pakietyzacja zasobów TEX-owych. Jak już wspomniałem, dystrybucja `tetex` jest udostępniana w postaci niepodzielonej na pakiety. Podział którego dokonują twórcy poszczególnych dystrybucji Linuxa należy uznać za nieudolny, trudno jednak wymagać więcej bez dobrej wiedzy o tych zasobach.

Rozmiary pakietów – na przykładzie kilku popularnych dystrybucji linuxowych – wyglądają następująco: tylko około 1% pakietów (ilościowo) przekracza rozmiar 10 MB (do 30 MB). W tym kontekście zasoby TEX-owe prezentują się niekorzystnie – w przejrzanych przeze mnie dystrybucjach pakiety zawierające `tetex` znajdują się w ścisłej czołówce największych.

Większym problemem jest fakt, że istniejące podziały na pakiety nie odpowiadają podstawowym funkcjom: na przykład nie ma pakietu `plain-tex`, który zawierałby niezbędne, „ortodoksyjne” minimum konieczne do przetworzenia pliku TEX-owego i tylko to. Brak pakietów „narodowych”, które zawierałyby dorobek lokalnych społeczności specyficzny dla danego języka, jak choćby fonty, pakiety makr TEX-owych itd. Brak wyodrębnionych pakietów zawierających całą dostępną dokumentację dodatkową z podziałem na języki.

TDS. Od dziesięciu lat TDS (TEX Directory Structure) jest lansowany jako obowiązujący standard zasobów TEX-owych na wszystkich platformach systemów operacyjnych (`http://www.tug.org/tds/`). O ile jednak jakaś ustalona struktura istniejących zasobów była potrzebna, to już wymuszanie takiej samej struktury na różnych systemach operacyjnych może budzić kontrowersje. Jednakże istniejąca struktura jest przykładem przerostu formy nad treścią: w samym tylko podkatalogu `fonts` jest 636 podkatalogów na pięciu poziomach, a w jednym tylko `fonts/tfm/public` są 132 podkatalogi na jednym poziomie (dane z TL7).

Wiadomo, że każde założenie jednego, nawet pustego katalogu to zajęcie jednego bloku (nie wchodząc w szczegóły zależne od stosowanego systemu plików jest to 0,5–4 KB). Jeśli nawet w przepastnych zasobach CTAN tak głęboki podział mógłby być uzasadniony, to w normalnym, użytkowym systemie plików jest labiryntem utrudniającym odnalezienie czegokolwiek. Ilość plików fontowych potrzebnych w znakomitej większości przypadków jest rzędu kilkuset, a więc nie byłoby bałaganu nawet gdyby wszystkie znalazły się w jednym katalogu.

Niejednorodność i niekompletność dokumentacji. Niestety ciągle pozostaje sporo programów i skryptów, które nie mają regularnej dokumentacji (`man`, `info`). To samo dotyczy ważniejszych plików konfiguracyjnych, które w Linuxie zwyczajowo mają swoje opisy `man`. Kuriozalnym przykładem jest osiem skryptów `altpdf*` (każdy po 13877 bajtów), o takiej samej zawartości i bez jakiejkolwiek dokumentacji. Niektóre opisy programów czy plików konfiguracyjnych owszem istnieją, ale w nietypowej formie lub dziwnym miejscu. Trzeba jednak przyznać, że w tej dziedzinie następuje stopniowa poprawa. Jeśli jednak potraktować TEX jako język programowania, to pozostaje tylko pomarzyć o takich plikach pomocy jakie ma np. język Tcl/Tk (zob. podrozdział **Przykład pozytywny** na następnej stronie). Na razie można się uczyć TEX-a tylko z TEXbook-a i innych książek.

Fonty w TEX-u. W strukturze TDS istnieje spory i niepotrzebny balast – na TL7 na przykład podkatalog `fonts` liczy sobie blisko 188 MB (to rozmiar łączny plików, bez uwzględnienia rozmiaru katalogów), podczas gdy przykładowo dla użytkownika, który używa tylko wszystkich polskich fontów, wystarczy... poniżej 6 MB! Do tego należy dodać, że ciągle brakuje w dystrybucji `tetex` jakiegokolwiek narzędzia do instalacji poszczególnych fontów – a przecież istotne jest, by każdy font był instalowany świadomie i to nie tylko z powodów licencyjnych. Dotychczas możliwa jest albo „ręczna” instalacja, choć dla początkującego TEX-nika nie jest to zajęcie proste, albo instalacja domyślna, która zarówno poprzez pakiety linuxowe, jak i za pomocą instalatora z TEXLive kopiuje całe drzewo katalogów z fontami, także tymi, które nigdy nie będą wykorzystane.

Ciągle jeszcze w zestawach instalacyjnych znajdują się fonty bitmapowe, choć od chwili, gdy zaczęto w sposób dojrzały wykorzystywać fonty obwiedniowe, nie widać żadnego uzasadnienia dla stosowania bitmap. Domyślne zachowanie TEX-a, który – jeśli nie zostały jawnie zdefiniowane wykorzystywane fonty – stosuje bitmapowe wersje fontów `cm`, także jest w obecnej sytuacji anachronizmem.

Cicha podmiana. W lipcu 2002 roku na liście dyskusyjnej GUST-u zgłoszony został problem niemożności uruchomienia dialogowego skryptu `texconfig`. Okazało się, że problem powstał z powodu cichej podmiany: Linux zawierał program `dialog`, podczas gdy `tetex` z dystrybucji TEXLive instalował go po cichu w nieco okrojonej wersji o nazwie `tcdialog`. Co gorsza: oba programy korzystały z tego samego pliku konfiguracyjnego, ale `tcdialog` miał manual o nazwie `tcdialog`, a zawartości opisującej `dialog`, lecz znów nieco okrojonej. Natomiast skrypt `texconfig` z dystrybucji linuxowej korzystał z programu `dialog`, a ten dystrybuowany z TEXLive – ze „swojego”, podmienionego odpowiednika `tcdialog`. Taka sytuacja powodowała potencjalny konflikt także między innymi skryptami korzystającymi z programu `dialog`: tymi z dystrybucji Linuxa a tymi z TEXLive. Tego rodzaju praktyki nie tylko utrudniają życie adminom, ale są objawem fatalnych nawyków programistycznych, a w efekcie sytuacja przypomina tę, znaną z innych systemów: „zainstalowałem sobie jakiś program i teraz inne programy przestały mi działać”.

Brak systematyki programów. Jeśli początkujący TEX-nik zapyta zaawansowanego jakie programy są mu potrzebne do najprostszej a sprawnej realizacji składu, to się dowie, że TEX, dvips, jakiś sensowny edytor tekstowy (uchowaj Boże od jakiegoś „procesora tekstu”), ghostscript, może jeszcze coś. Tymczasem w pakietach linuxowych czy na TEXLive jest tych programów o dwa rzędy wielkości więcej (na TL7 jest 235 programów i skryptów w katalogu `/bin/i386-linux` zajmujących ponad 11 MB) i nie istnieje jakiś podzbiór typu „pakiet-minimum”. To jest sytuacja bardzo niezdrowa, gdyż wybieranie w takim mnóstwie tego, co potrzebne, tego czego początkujący (i nie tylko) chciałby używać, to zadanie trudne, jeśli nie najtrudniejsze. Ta ilość jest jednak sztucznie zawyżona, gdyż dla dla niektórych programów bywa po kilka czy kilkanaście „skrypcików”, które różnią się jedynie parametrami wywołania programu, podczas gdy wystarczyłoby napisać krótką dokumentację do takiego programu o użyciu tychże parametrów, lub obudować go jakimś GUI (*Graphical User Interface*). Pół biedy, jeśliby to wszystko było dobrze skatalogowane tematycznie. Niestety nie jest. Trudno wybrać to, co ważne z katalogu alfabetycznego zawierającego setki pozycji. A wśród różnych dystrybucji Linuxa właśnie jednym z istotnych czynników świadczących o ich jakości pozostaje odpowiednio przeprowadzona systematyka zasobów – owszem, wymaga to wiele pracy od osób przygotowujących pakiety, ale oszczędza sporo czasu wszystkim użytkownikom.

Instalator z TEXLive. W świetle tego, co napisałem powyżej o znaczeniu pakietów instalacyjnych w Linuxie, propozycja stosowania innego, znacznie gorszego instalatora, proponującego wybór instalacji „mało – dużo – wszystko”, z innymi ustawieniami, niemożnością przeprowadzania sensownej aktualizacji czy deinstalacji wybranych elementów, będącego w pełnym konflikcie z resztą systemu, nie ma najmniejszego sensu.

Biblioteka kpathsea. Początki tego pomysłu sięgają bodaj roku 1990. Wówczas dzięki tej bibliotece (dokładniej: dzięki jej „wszyciu” do podstawowych programów) udawało się łatwiej (szybciej) znajdować pliki fontowe. Dorobiono też mechanizm, który potrafił „w locie” generować odpowiednie fonty bitmapowe o odpowiednim rozmiarze i zadanej rozdzielczości (miało to znaczenie w czasach, gdy system operacyjny DOS...). Niestety ta spora maszyneria jest daleka od doskonałości: jest trudna w konfiguracji a bywa uciążliwa w stosowaniu, gdyż wystarczy mała pomyłka w nazwie fontu, a `kpathsea` natychmiast inteligentnie zaczyna poszukiwania pliku o tej nazwie wśród źródeł metafontowych, próbuje jakichś podmian, po to, by po dłuższej chwili zakomunikować, że i tak nie może pomóc. W dodatku mechanizm jest nieelastyczny, a więc na przykład nie bardzo daje się zaprząc do poszukiwania innych typów plików, co skutecznie „zabezpiecza” go przed szerszym wykorzystaniem. W kontekście faktu, iż fonty bitmapowe w zastosowaniach TEX-owych odchodzą w przeszłość, także `kpathsea` staje się niepotrzebnym balastem.

Tzw. lokalizacja. Oczywiście chodzi o uwzględnienie języka innego niż angielski w działaniach danego programu, w oparciu o aktualną wartość zmiennych systemowych. W przypadku pakietów `tetex` brakuje tego rodzaju wrażliwości (np. domyślnego włączenia `MeXa` lub polskich wzorców dzielenia wyrazów, czy instalacja pakietów z polskimi fontami lub dokumentacją dodatkową w języku polskim).

Konfiguracja. Po prawidłowym zainstalowaniu pakietów `tetex` w większości dystrybucji konieczne jest wykonanie kilku funkcji za pomocą skryptu `texconfig`. Konieczność pamiętania o tych działaniach jest zbędnym obciążeniem.

Od strony Linuxa

Dezintegracja zasobów fontowych. Niestety, Linux też nie jest ideałem systemu operacyjnego. Jedną z poważnych (choć – jak mniemam – nielicznych) wad tego systemu jest praktycznie nieokiełznana dowolność w zakresie korzystania z fontów, a co za tym idzie – niemożność skoordynowanego zarządzania instalacją i dostępnością fontów. Tak więc nie ma np. możliwości bezpośredniego użycia Antykwy Półtawskiego w programie GIMP lub choćby w terminalu, mimo zainstalowania go dla TEX-a. Jasne, że wszystko można sobie ręcznie dodłubać, ale każdy program wymaga innego ustawiania, szuka fontów w innych miejscach (a więc wymaga zazwyczaj zwielokrotniania plików lub komplikacji dowiązaniami), wiele programów jeszcze nie reaguje właściwie na ustawienia zmiennych `locale` itd.

W dodatku nie zapowiada się na rychłą poprawę tej sytuacji. Słuszna i – jak mi się wydaje – bardzo potrzebna inicjatywa, Linux Font Project (`http://dreamer.nitro.dk/linux/lfp/`) sprawia wrażenie zarzuconej – najświeższa wiadomość pochodzi sprzed półtora roku.

Przykład pozytywny. Skryptowy język programowania Tcl/Tk jest co prawda zaimplementowany na większości platform, ale wygląda na to, że w Linuxie „czuje się” najlepiej. Wszystkie podstawowe biblioteki są zazwyczaj w domyślnych zestawach instalacyjnych; wszystko jest dobrze wtopione w stosowaną w Linuxie strukturę systemu plików określoną standardem FHS (Filesystem Hierarchy Standard, np. `http://www.pathname.com/fhs/`). Istotne dla początkującego jest, iż wszystkie podstawowe polecenia mają swoje opisy `man`. Naukę można praktycznie rozpocząć od wpisania w X-terminalu, udostępniającym pełne środowisko pracy, np. polecenia `wish` (jest to jedna z powłok). Pomoc można oczywiście znaleźć w wielu miejscach w sieci oraz na listach dyskusyjnych, ale praca i nauka początkującego programisty może się opierać **wyłącznie** na lokalnej stacji roboczej z zainstalowanym domyślnie Linuxem (żadnego doinstalowywania, konfigurowania, generowania, za to z dostępem do kontekstowych plików pomocy), co – przynajmniej w moich oczach – jest bliskie ideału (bo ideał jak wiadomo nie istnieje).

Postulaty poprawy integracji

Rozmiar dystrybucji. Fazy życia dojrzałego software'u wyglądają dość rozmaicie. Bywa, że program osiąga coraz większą dojrzałość, wręcz zbliża się do doskonałości, ale przestaje się dynamicznie rozwijać. Do takich programów można zaliczyć między innymi przeglądarkę internetową Galeon. Zdarza się także, zazwyczaj w dużych projektach, że mimo ciągłego, znacznego rozwoju, wzrostu dojrzałości i niezawodności, niektórzy użytkownicy i tak rezygnują na korzyść innych, np. mniejszych, alternatywnych projektów. Tak jest od pewnego czasu ze środowiskiem graficznym Gnome. Powód jest prosty. Gnome jest ciągle dobry, ale coraz bardziej się rozrasta. Zajmuje więcej pamięci, jest coraz wolniejszy, coraz trudniej w nim zachować np. jednolitość wyglądu czy konfiguracji. I faktycznie powstają projekty alternatywne, o takiej samej funkcjonalności, a pozbawione wielu wodotrysków, dzięki czemu są znacznie „lżejsze”, a w konsekwencji o wiele szybsze (choćby Xfce).

Nie życzę TEX-owi, żeby się pojawił jakiś konkurencyjny `neotex`, który zachowując w pełni samego TEX-a oraz utrzymując funkcjonalność oprogramowania TEX-owego potrzebną w 90% przypadków, miałby 10% obecnej „wagi” dystrybucji `tetex`, choć obawiam się, że jacyś szaleńcy wreszcie wykonają taki krok. Zamiana dystrybucyjnego krążka CD na DVD spowoduje jedynie, iż tym trudniej będzie cokolwiek znaleźć. Podstawowy postulat dotyczący rozmiaru to radykalna selekcja zarówno wśród programów („skrypciki”, obsługa fontów bitmapowych, programy rzadko używane) jak i fontów (bitmapy, fonty niekompletne). Oczywiście wszystko i tak pozostaje dostępne na CTAN-ie, chodzi tylko o „odchudzenie” dystrybucji `tetex`.

Systematyka. Należy przemyśleć i opracować systematykę udostępnianych programów.

Fonty. W odniesieniu do fontów proponuję najpierw usunąć z dystrybucji fonty niekompletne i pozostałe bitmapowe, a potem podzielić całą resztę na niezależne pakiety:
- zestaw konieczny do działania TEX-a (fonty `cm`),
- oddzielne zestawy fontów związanych z Omegą,
- zestawy narodowe (np. oddzielny pakiet zawierający wszystkie polskie i spolszczone fonty),
- zestawy wielojęzyczne, specjalistyczne i inne.

Ewentualne dzielenie katalogu `fonty` na podkatalogi proponuję zostawić osobie instalującej lub ustawić w programie służącym do obsługi fontów jeśli taki trafi do dystrybucji.

Konieczne jest utworzenie programu do obsługi fontów. Należy przewidzieć w przyszłości możliwość integracji zasobów fontowych w ramach całego Linuxa.

Kpathsea. Cały mechanizm `kpathsea` służący generowaniu fontów bitmapowych powinien zostać usunięty i zastąpiony prostym mechanizmem udostępniania plików fontowych (ewentualnie biblioteka `kpathsea` mogłaby się znaleźć w oddzielnym pakiecie).

Formaty. Jeśli chodzi o generowanie formatów: należy umieszczać w dystrybucjach wygenerowane zestawy, albo generowanie wykonywać w ramach procedur postinstalacyjnych w sposób niewidoczny dla użytkownika. Niektóre mechanizmy (jak np. dzielenie wyrazów) powinny zostać wyodrębnione w postaci wymiennych, ładowalnych modułów bibliotecznych.

Postać dystrybucji. To wszystko, co zostałoby po dokonanej selekcji, powinno być rozsądnie podzielone na około 20 do 30 pakietów RPM lub DEB, prawidłowo wykorzystujących środowisko Linuxa (`locale`, struktura katalogów, pliki pomocy i `man`). Jaknajszybciej należy zapomnieć o „instalatorze” oferowanym ciągle w ramach TEXLive.

Uwagi końcowe

Problemy wynikające z braku pełnej integracji zasobów TEX-owych ze współczesnymi systemami operacyjnymi stosunkowo często pojawiają się na listach dyskusyjnych, nie tylko GUST-owej. Porady siłą rzeczy są próbą leczenia objawowego. Wielka masa dobrego oprogramowania po prostu dryfuje. Próby zainteresowania naszego środowiska tą sytuacją, które – być może nieudolnie – podejmowałem kilkakrotnie w ciągu ostatnich lat, szczególnie w kontekście interesującego mnie Linuxa, nie przyniosły żadnych rozstrzygnięć. Owszem, większość luminarzy tego środowiska wygłaszała publicznie tezy, będące w znacznej mierze podstawą sformułowanych tu zarzutów. Niestety, sytuacja nie uległa poprawie. Tak więc niniejszy artykuł może być interpretowany jako kolejna próba „wsadzenia kija w mrowisko”. Uprzedzając ewentualne zarzuty o „kalaniu własnego gniazda” (jak by nie było czuję się w kontekście Linuxa i TEX-a jak w domu rodzinnym) pozwolę sobie zacytować śp. Stefana Kisielewskiego: *Czy ten ptak gniazdo swoje kala, który kala, czy ten, co kalać innym nie pozwala?* Usiłowałem starannie omijać jakiekolwiek konotacje personalne, skupiając się na problemach, które mi doskwierają, a których sam nie rozwiążę. Niestety, na takie same problemy będą się nadziewały następne pokolenia dzielnych adeptów pięknego składu i poskramiaczy TEX-a. Diagnoza została więc postawiona. Możemy ją merytorycznie negować, modyfikować, przedstawić inną, albo po prostu przyjąwszy ją podjąć próbę leczenia. Ja w każdym razie współdryfowanie jako rozwiązanie wszystkich problemów zdecydowanie odrzucam.