

Władcy okienek

Po uruchomieniu X Window większość użytkowników Linuksa rozkłada bezradnie ręce. Jednak dostosowanie środowiska graficznego do własnych potrzeb, a nawet radykalna zmiana jego wyglądu, nie jest rzeczą trudną.

System X Window zawiera protokół określający sposób, według którego programy mogą korzystać z konsoli graficznej. Jest on oparty na komendach wysyłanych siecią, więc nie ma znaczenia, czy aplikacje korzystają z konsoli znajdującej się w innym budynku, czy monitora podłączonego bezpośrednio do komputera. Programy wysyłają wszystkie polecenia pod adres znajdujący się w zmiennej środowiskowej **DISPLAY**, zdefiniowanej na komputerze, na którym uruchamiany jest program. Jej wartość składa się z dwóch części przedzielonych dwukropkiem: `<internetowy adres komputera>:<numer monitora>`. W przypadku konsoli komputera lokalnego zmienna ta ma wartość `localhost:0` lub po prostu `:0`.

Przyjmowaniem oraz wykonywaniem komend wysyłanych przez programy zajmuje się tzw. X-Server; znajduje się on w pliku `/usr/X11R6/bin/X`. Aplikacja ta czeka, aż jakiś program (uruchomiony

lokalnie lub na innym komputerze) połączy się z portem 6000, po czym sprawdza, czy ma on prawo do wyświetlania (więcej informacji na ten temat można uzyskać, wykonując polecenie `man xhost`). Jeśli tak, X-Server przyjmuje dalsze polecenia, zwykle są to żądania przydzielenia aplikacji pewnego obszaru ekranu – okna – oraz komendy rysujące jego zawartość. To, że serwer X-ów komunikuje się z aplikacjami za pomocą sieci, oznacza, że na jednej konsoli mogą być wyświetlane okna programów uruchomionych na wielu różnych komputerach połączonych siecią.

Menedżer okien

Podstawowym pojęciem w protokole używanym w X Window jest okno, czyli prostokątny obszar kontrolowany przez jedną aplikację. Zdefiniowane są tam również komendy określające operacje na nich, takie jak przesuwanie czy zmiana ich wielkości. Jednak system X Window nie precyzuje, jak są one wykonywane, co oznacza, że

żadna aplikacja nie robi tego samodzielnie – po prostu wysyła odpowiednie polecenie, a jego realizacją zajmuje się menedżer okien. Jest to program, który „przechwytuje” pojawiające się na konsoli okienka i dodaje do nich elementy interfejsu – tytuły, ikonki i inne. Aplikacja ta interpretuje także kliknięcia myszą różnych części ramek. Przykładowo, gdy użytkownik wybierze przycisk Minimalizuj, menedżer wyśle serwerowi X polecenia schowania okienka i narysowania ikonki.

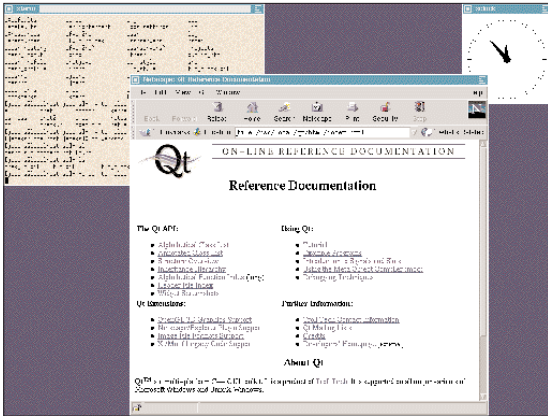
Menedżer okien określa mechanizmy, za pomocą których użytkownik może manipulować okienkami. Większość z nich obsługuje również specjalne menu, które pojawiają się, gdy użytkownik kliknie tzw. root window, czyli pulpit.

Oddzielenie funkcjonalności serwera X od interfejsu użytkownika pozwala na łatwą jego zmianę. Menedżerów okien jest bardzo wiele i każdy znajdzie tu coś, co mu się spodoba; należy jednak pamiętać, że bogate graficznie menedżery zajmują więcej pamięci i mocy procesora.

3, 2, 1, 0, start!

Pod Linuksem system X Window uruchamiany jest komendą `startx` lub przez program `xdm`. Ten drugi ma tę zaletę, że środowisko graficzne inicjowane jest zaraz po starcie systemu i tam przebiega logowanie do systemu. W przypadku `startx` identyfikacja użytkownika ma miejsce w trybie tekstowym, po czym uruchamia on X Window ręcznie. `xdm` jest startowany, gdy wybrany jest piąty poziom operacyjny (runlevel 5). Można go ustawić, edytując plik `/etc/inittab` i zamieniając liniijkę zawierającą `id:3:initdefault:` na `id:5:initdefault:`. Po ponownym starcie systemu użytkownika przywita konsola graficzna.

Po zainicjowaniu X Window przez komendę `startx` uruchamiany jest skrypt `/etc/X11/xinit/xinitrc`. Jego zadaniem jest ustawienie pewnych właściwości interfejsu, takich jak np. układu klawiatury (więcej informacji na ten temat można uzyskać, wykonując polecenie `man xmodmap`) oraz domyślnych parametrów niektórych aplikacji – tzw. resources (pomoc na ten temat jest dostępna po wykonaniu komendy `man xrdp`). Następnie skrypt stara się uruchomić menedżera okien według specjalnego schematu. Najpierw sprawdza, czy istnieje plik `$HOME/.Xclients`, czyli zbiór o nazwie `.Xclients` w katalogu domowym użytkownika (uwaga: plik ten zaczyna się od kropki, więc nie jest wyświetlany przez komendę



System X Window uruchomiony z opisanym niżej plikiem .Xclients (menedżer twm)

1s. Aby sprawdzić, czy istnieje, trzeba użyć polecenia `ls -a`). Jeśli plik został znaleziony, jest traktowany jako skrypt, który inicjuje wybrany przez użytkownika menedżer okien. Jeżeli opisywanego wyżej zbioru nie ma, wykonany zostaje `/etc/X11/xinit/Xclients`. Jest to skrypt, który stara się sprawdzić, jakie menedżery okien znajdują się w systemie i uruchomić któryś z nich.

Gdy X Window jest uruchamiane za pomocą `xdm`, skryptem inicjującym jest `/etc/X11/xdm/Xsession`. Z punktu widzenia użytkownika nie ma jednak różnicy, gdyż skrypt ten zachowuje się tak samo jak `/etc/X11/xinit/xinitrc`.

Użytkownik może wybrać menedżer okien, edytując plik `.Xclients`, znajdujący się w jego domowym katalogu. Uwaga: ponieważ plik ten jest skryptem, musi mieć nadane prawa do wykonywania. Można to zrobić komendą `chmod u+x ~/.Xclients`. Administrator może wybrać domyślny menedżer okien, umieszczając odpowiednie polecenie w pliku `/etc/X11/xinit/Xclients`, który jest wykonywany w przypadku, gdy użytkownik nie wybrał własnego menedżera.

Oba opisane pliki mają następującą właściwość: zakończenie wykonywania zawartego w nich skryptu jest dla systemu X Window sygnałem, że sesja graficzna dobiegła końca. Nie należy więc uruchamiać menedżera okien w tle (za pomocą znaku `&` umieszczonego na końcu polecenia), gdyż wówczas skrypt i co za tym idzie X Window zakończą się natychmiast. Poprawny plik `.Xclients` wygląda na przykład tak:

```
#!/bin/sh
exec /usr/X11R6/bin/twm
```

Skrypt ten uruchamia bardzo prostego menedżera okien o nazwie `twm`, który domyślnie znajduje się w wielu dystrybucjach Linuksa, w tym w Red Hacie 5.1. Następnie czeka, aż `twm` skończy pracę, po czym kończy się sam.

W pliku `.Xclients` można również umieścić wywołania aplikacji, które powinny być uruchamiane natychmiast po starcie systemu X Window. Odpowiednie wpisy należy umieścić przed komendą inicjującą menedżera okien i uruchomić je w tle za pomocą `&`. W przeciwnym wypadku

skrypt, zanim przejdzie do wykonywania następnej linijki, będzie czekał na zakończenie się sesji X Window.

Oto przykładowy plik `.Xclients` (efekt jego wywołania znajduje się na ilustracji powyżej):

```
#!/bin/sh
xsetroot -solid darkblue &
# Ciemnoniebieskie tło
# /usr/X11R6/lib/X11/rgb.txt)
xterm -geometry +0+0 &
# xterm w lewym-górnym rogu
xclock -geometry -0+0 &
# zegar w prawym-górnym rogu
netscape &
exec /usr/X11R6/bin/twm
# menedżer okien
```

Aby dostosować menedżera okien do swoich potrzeb, najprościej jest skopiować domyślny plik (patrz tabela Zestawienie plików konfiguracyjnych) do katalogu domowego, nadać mu odpowiednią nazwę i tam dokonać jego edycji. Na początek warto przeczytać o konfiguracji `twm`, pierwszym z opisywanych tutaj menedżerów okien – jest on najprostszy i jego opis zawiera pojęcia dotyczące pozostałych, jak np. sposób wybierania fontów w systemie X Window.

Nieraz zdarza się, że po wprowadzeniu pewnych zmian w pliku inicjującym i ponownym uruchomieniu X Window menedżer się zawiesi. Należy wówczas zamknąć sesję kombinacją klawiszy `[Ctrl]+[Alt]+[BkSp]`. Uwaga: `[BkSp]` i `[Del]` to dwa różne klawisze – jeśli się pomylimy, zamiast zamknąć sesję X Window, zresetujemy komputer! Po wyjściu z trybu graficznego należy odszukać komunikaty o przyczynach błędnego

działania programu. Jeśli X Window został uruchomiony za pomocą `startx`, pojawiają się one na konsoli tekstowej. Jeśli używamy `xdm`, informacji o błędach należy szukać w pliku `.xsession-errors`, znajdującym się w katalogu domowym.

Małe jest piękne

Najprostszym i najmniejszym z opisanych tutaj menedżerów jest `twm` – przy pięciu otwartych okienkach zajmuje w pamięci RAM zaledwie 1,5 MB, nadaje się więc do komputerów wyposażonych w małą ilość pamięci operacyjnej. Jego poważnym ograniczeniem jest brak wirtualnych pulpitów (istnieje rozszerzona wersja tego programu o nazwie `ctwm`, która jest wzbogacona tą opcją).

Plik konfiguracyjny `twm` składa się z wielu części, z których każda definiuje inne właściwości interfejsu, jak skróty klawiaturowe, czynności przypisane kliknięciom, menu i inne.

Pierwsza sekcja określa niektóre elementy `twm`, jak np. fonty używane w tytułach okien. Polecenia definiujące rodzaj używanych czcionek mają specjalny format i można je wpisać ręcznie, ale dużo łatwiej posłużyć się programem `xfontsel`. Ta aplikacja pozwala na podanie parametrów fontu, po czym wyświetla, jaka czcionka, ze znajdujących się w systemie, spełnia te wymagania.

Druga część pliku konfiguracyjnego dotyczy ustawień kolorów. Zawiera się ona pomiędzy klamrami poprzedzonymi słowem `Color`. Można tu określić barwy wszystkich elementów `twm`, np. ramek (`BorderColor`), tytułów (`TitleBackground`) itd. Dozwolone jest podawanie ▶ 308

podstawy

Polecenia twm

f.beep – dźwięk ostrzegawczy
f.exec „nazwa” – uruchom program „nazwa”. Przy uruchamianiu zewnętrznych programów za pomocą `f.exec` należy pamiętać, by pracowały one w tle. Robimy to, dodając do nazwy aplikacji znak `&`
f.delete – zamknij wybrane okno
f.iconify – zminimalizuj wybrane okno
f.menu „nazwa” – wyświetl menu „nazwa”
f.quit – zakończ pracę (zazwyczaj powoduje to koniec sesji X)

predefiniowanych nazw, np.: **DarkBlue**, lub określanie nasycenia kolorów składowych zgodnie z modelem RGB: **rgb:x/y/z**, gdzie **x**, **y** i **z** odpowiadają nasyceniu barwy czerwonej, zielonej i niebieskiej. Wszystkie zdefiniowane wcześniej nazwy kolorów można znaleźć w pliku **usr/X11R6/lib/X11/rgb.txt**.

Kolejna sekcja pliku konfiguracyjnego dotyczy makr. Są to polecenia, które definiują komendy w dalszej części przypisane przyciskom myszy lub kombinacjom klawiszy. Każda taka linia ma składnię:

```
Function nazwa-makra { polecenie1
    polecenie2 polecenie3 ... }
```

Wszystkie polecenia mają postać **f.nazwa-polecenia**. Niektóre z nich znajdują się w ramce poniżej; opis wszystkich można otrzymać, wykonując komendę **man twm**.

Makra mogą być wywoływane za pomocą myszy lub klawiatury. W zależności od wybranego przycisku, położenia kursora oraz ewentualnego naciśnięcia dodatkowego klawisza kliknięcie może dawać różne skutki. Menedżer zakłada, że urządzenie wyposażone jest w trzy przyciski, do których odwołujemy się przez nazwy **Button1**, **Button2** i **Button3**. Linijka opisująca zachowanie myszki wygląda następująco:

```
ButtonN = stan-klawiatury : położenie1|
    położenie2|... : polecenie
```

Atrybut **stan-klawiatury** opisuje fakt naciśnięcia jednego z klawiszy: [Shift], [Alt] lub [Ctrl]. Odpowiadające im wartości to: **shift** bądź **s**, **meta** lub **m** oraz **control** bądź **c**. Przedostatnia grupa parametrów to lista miejsc, których kliknięcie ma

wywołać wybrane polecenia. Wśród nich mogą być między innymi **root** (kliknięcie w pulpit), **title** (wybranie tytułu okna) oraz **icon**, odnoszące się do ikonki zminimalizowanego okna. Atrybut **polecenie** może być nazwą zdefiniowanego wcześniej makra lub polecenia (f.nazwa). Przydatną komendą jest **f.menu "nazwa-menu"**, która pozwala na przypisanie kliknięciu zdefiniowanego w dalszej części pliku menu.

Podobnie jak kliknięciom, tak i kombinacjom klawiszy możemy przypisać polecenie lub makro. Linijka definiująca skrót klawiaturowy wygląda prawie tak samo jak opisywana przed chwilą – jedyną różnicą jest to, że pierwszy parametr jest nazwą klawisza, a nie przycisku myszy. Klawisze mają intuicyjne nazwy, np.: **F5**, **Left**, **Tab** itd.

Jeśli w którejś z opisywanych wcześniej sekcji użyliśmy polecenia **f.menu "nazwa-menu"**, w tej części pliku musimy zdefiniować menu, do którego wcześniej się odwoływaliśmy. Definicja taka wygląda następująco:

```
menu "nazwa-menu"
{
    "nazwa1" polecenie1
    "nazwa2" polecenie2...
}
```

Atrybut **nazwaN** jest tekstem, który pojawia się w n-tej pozycji menu, a **polecenieN** poleceniem, które zostanie wywołane, jeśli zostanie ona wybrana. Podobnie jak w poprzednich częściach pliku konfiguracyjnego, w miejscu **polecenieN** można podać nazwę makra lub polecenia (**f.beep**, **f.delete** itp.).

Jak Windows

Opisany przed chwilą menedżer jest bardzo prostą aplikacją. Chociaż jego możliwości konfiguracyjne są duże, to interfejs tego programu wydaje się bardzo ubogi szczególnie, jeśli używaliśmy wcześniej Windows 95. Menedżerem, który stara się upodobnić do „okienek” Microsoftu, jest **fwm2**. Środowisko to ma wiele udogodnień, np. znany użytkownikom Windows pasek zadań oraz odpowiednik menu Start, specjalny panel z przyciskami do uruchamiania aplikacji, kolorowe ikonki, wirtualne pulpity, a także ramki o „trójwymiarowym” wyglądzie. Jest nieco większy niż **twm** – przy pięciu otwartych



W wielu menedżerach okien możemy tworzyć własne menu z przydatnymi poleceniami

okienkach razem z modułami (pasek zadań, panel uruchomieniowy i inne) zajmują 3,5 MB. **fwm2** to domyślny menedżer okien w Red Hacie 5.x.

Podobnie jak było to w przypadku poprzedniego menedżera, plik konfiguracyjny **fwm2** jest podzielony na części. To środowisko działa jednak na trochę innej zasadzie, więc poszczególne części zbioru nie odpowiadają dokładnie sekcjom z pliku **twm**.

Wygląd okien **fwm2** jest określany przez style. Określają one takie atrybuty okna, jak wygląd i zachowanie ramki, ikonkę pojawiającą się na pasku zadań po zminimalizowaniu zadania, czy okno powinno być zawsze na wierzchu i inne. Linia definiująca styl wygląda następująco:

```
Style "nazwa" opcja1, opcja2, ...
```

Różnym rodzajom okien można przypisać różne style – gdy na pulpicie pojawia się nowa aplikacja, **fwm2**, na podstawie tekstu znajdującego się w tytule okna, stara się dopasować do niego styl. Jego nazwa może zawierać znak *, więc styl **Fwm*** będzie przypisany każdemu oknu, którego nazwa zaczyna się od **Fwm**. Jeśli jakiemuś tekstowi zawartemu w tytule odpowiada więcej niż jeden styl, użyte atrybuty są sumą opcji wszystkich odpowiadających definicji. Ustawienia zawarte w stylu * dotyczą wszystkich okienek. Opcji, których można używać, jest bardzo wiele. Kilka z nich znajduje się w ramce obok.

Podobnie jak w przypadku **twm**, **fwm2** pozwala na definiowanie makr, które są uruchamiane kliknięciem myszy lub naciśnięciem kombinacji klawiszy. Dodatkowo w każdym poleceniu należy określić, jaka akcja myszki je wywołuje. Do wyboru mamy **M** (wykonanie ruchu ► 311

podstawy

Polecenia fwm2

Exec „nazwa” – uruchom program „nazwa”

Delete – zamknij okno

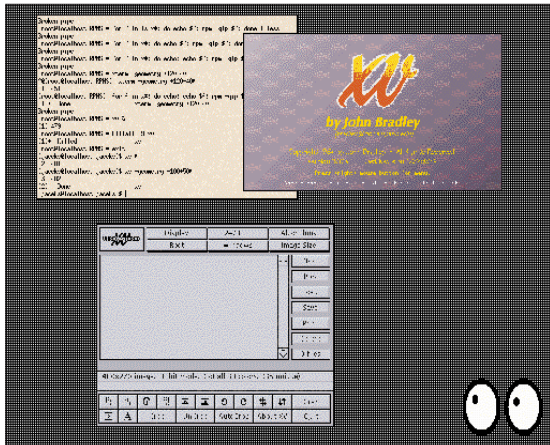
Maximize – powiększ okno do rozmiaru pulpitu

Function „nazwa” – wykonaj zdefiniowaną poprzednio funkcję

Quit – zakończ pracę

Popup „nazwa” – wyświetl menu „nazwa”

Module „nazwa” – uruchom moduł „nazwa”



X-Server uruchomiony bez menedżera okien wygląda bardzo nieciekawie i nie pozwala na przesuwanie okienek

kursorem), C (pojedyncze) i D (podwójne kliknięcie myszką). Jeśli użyjemy I, polecenie zostanie wykonane natychmiast.

Użytkownik może tworzyć makra o dowolnych nazwach, ale dwie z nich są zarezerwowane. **InitFunction** jest wywoływane bezpośrednio po pierwszym uruchomieniu menedżera okien, a **ReStartFunction** w przypadku jego kolejnych startów. Pierwsze powinno uruchamiać aplikacje, które mają się od razu pojawiać na pulpicie – efekt jest podobny do wpisania ich do pliku **.Xclients**. Składnia definicji makra wygląda następująco:

```
AddToFunc "nazwa"
+ "kiedy1" polecenie1
+ "kiedy2" polecenie2
+ "kiedy3" polecenie3
...
```

Parametr **kiedyN** określa, kiedy dana komenda powinna zostać wykonana (I, M, C lub D). Polecenia znajdujące się w miejscu **polecenieN** są podobne do poznanych już w **twm**. Niektóre z nich znajdują się w ramce obok.

Przykładowe makro **InitFunction** może więc wyglądać tak:

```
AddToFunc "InitFunction"
+ "I" Exec xv -root -quit chip.bmp &
# tło pulpitu
+ "I" Module FvwmButtons
# okienko z przyciskami
# do uruchamiania programów
+ "I" Module FvwmTaskBar
# pasek zadań
+ "I" Exec xterm &
# okienko xterm
```

W przeciwieństwie do poprzedniego menedżera definicja makra w **fwm2** jest dwustopniowa. Rozpoczyna się od polecenia określającego kolory i tryb działania. Oto jego schemat:

```
MenuStyle kolor-tekstu
kolor-tła kolor-cienia
font tryb
```

Opcja **kolor-cienia** określa, jaką barwą będą wyświetlane nieaktywne pozycje menu. Jako tryb można wpisać **fwm** lub **mwm**. Wpisanie pierwszej z nich spowoduje, że po kliknięciu w pozycję prowadzącą do kolejnego menu poprzednie zniknie. Wartość **mwm** da taki efekt, że menu będą działały tak jak w Windows: kolejne podmenu będą rozwijały się bez kliknięcia, a wszystkie znikną dopiero po wybraniu pozycji końcowej.

Po linii **MenuStyle** następują opisy wartości menu:

```
AddToMenu "nazwa-menu"
+ "nazwa1 %ikonka1.xpm%" polecenie1
+ "nazwa2 %ikonka2.xpm%" polecenie2
...
```

Podobnie jak w **twm**, **nazwaN** jest tekstem, który pojawi się na n-tej pozycji. Menu **fwm2** są jednak wzbogacone ikonkami, które można wybrać, wpisując ich nazwę pomiędzy znaki **%**. **polecenieN** może być jednym z poleceń (**Quit**, **Popup**, **Exec** itp.) lub identyfikatorem makra. Uwaga: nazwę ikonki należy oddzielić od tekstu spacjami, a nie tabulatorem.

Po zdefiniowaniu menu możemy sprawić, by pojawiało się ono po kliknięciu myszką. Składnia polecenia jest prawie identyczna z opisem menu w **twm**:

```
Mouse numer-przycisku gdzie
stan-klawiatury polecenie
```

Pierwszy parametr to liczba oznaczająca numer przycisku – 1, 2 lub 3, **gdzie** określa miejsce kliknięcia – niektóre dozwolone wartości to R (pulpit), T (tytuł okna), S (ramka okna z wyłączeniem narożników), I (ikonka zminimalizowanego okna) i F (róg ramki). Kolejny argument, **stan-klawiatury**, mówi o stanie klawiatury w chwili kliknięcia. **N** oznacza, że nie został wybrany żaden klawisz,

A odpowiada naciśnięciu dowolnego. Oprócz tych wartości można użyć jeszcze **C**, **S** i **M**, które oznaczają kolejno fakt wybrania klawiszy [Ctrl], [Shift] i [Alt]. Ostatni parametr, **polecenie**, jest nazwą polecenia lub jednego ze zdefiniowanych wcześniej makr.

Domyślne ustawienia **fwm2**, z którymi rozpoczynamy prace po instalacji systemu Red Hat 5.1, sprawiają, że rozmiar okienka można zmieniać, klikając narożnik ramki, natomiast wybranie jednego z brzegów powoduje przeniesienie całości. Większość użytkowników Windows jest jednak przyzwyczajona, że kliknięcie brzegu również pozwala na rozszerzenie okienka. Zachowanie ramek jest określone następującymi dwoma poleceniami w domyślnym pliku konfiguracyjnym:

```
Mouse 1 F A Function "Resize-or-Raise"
Mouse 1 TS A Function "Move-or-Raise"
```

Kliknięcie rogu ramki (parametr **F**) wywołuje funkcję **Resize-or-Raise**, która przesuwa okienko na wierzch, jeśli nastąpiło samo kliknięcie, i zmienia jego wielkość, jeśli następną czynnością jest przesunięcie kursora. Kliknięcie każdej innej części ramki lub tytułu (**T** i **S**) wywołuje funkcję **Move-or-Raise**, która działa podobnie, ale przemieszczanie kursora powoduje zmianę pozycji okna.

podstawy

Definiowanie stylów

HandleWidth x – szerokość ramki w pikselach

NoTitle – okno nie będzie miało paska z tytułem

StaysOnTop – zawsze na wierzchu

Sticky – okno będzie widoczne zawsze na wybranym pulpicie wirtualnym

Minicon icon.xpm – nazwa ikonki, która pojawi się na pasku zadań. Ikonka o tej nazwie musi się znajdować w folderze, którego nazwa znajduje się w zmiennej środowiskowej **PixmapPath** lub **IconPath**

BackColor kolor – kolor ramki. W środowisku **fwm2** kolory można definiować za pomocą nazw lub podając nasycenie barw składowych, według modelu RGB – **#rrggbb**, gdzie **rr**, **gg** i **bb** podane są w systemie szesnastkowym

Aby upodobnić zachowanie *fvwm2* do Windows, wystarczy przenieść **S** (kliknięcie w ramkę) do górnej linijki.

Przypisanie polecenia kombinacjom klawiszy wygląda podobnie jak w przypadku kliknięcia, jednak zamiast przycisków myszki podajemy któryś z klawiszy:

Key nazwa-klawisz gdzie
stan-klawiatyry polecenie

Warto zauważyć, że również kombinacjom klawiszy można przyporządkować menu. Jeżeli na naszym pulpicie często panuje bałagan i trudno jest znaleźć miejsce nie zakryte przez żadne okno, menu z przydatnymi poleceniami można przyporządkować kombinacji klawiszy. Możemy także użyć klawisza z symbolem Windows, który najczęściej w Linuksie jest bezużyteczny. Pierwszą czynnością jest nadanie mu nazwy (na przykład *menu*), przez dodanie do funkcji *InitFunction* polecenia:

```
+ "I" Exec xmodmap -e 'keycode 115 = Menu'
```

Następnie przypisujemy klawiszowi menu menu-pulpit:

Key Menu A N Popap "menu-pulpit"

W *fvwm2* istnieją moduły, które pozwalają wzbogacić menedżera takimi elementami, jak pasek zadań, pager – czyli okno pokazujące miniatury wirtualne pulpity i zawarte w nich okna (*FvwmPager*), okienka z przyciskami (*FvwmButtons*) itd. Każda linijka opisująca moduł zaczyna się od gwiazdki oraz nazwy modułu.

Polecenie **FvwmTaskBarBack*

#c0c0c0 ustawia tło paska zadań (moduł *FvwmTaskBar*) na kolor *#c0c0c0* (jasnoszary). Dokładne opisy każdego modułu można uzyskać, wykonując komendę *man moduł*, gdzie *moduł* jest nazwą komponentu, o którym chcemy się czegoś dowiedzieć. Polecenie wyświetli nam opis wszystkich dostępnych opcji.

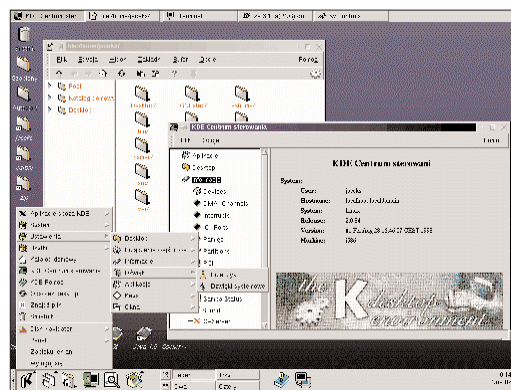
Najciekawszym modułem jest *FvwmButtons*. Jest to okienko zawierające przyciski uruchamiające często używane aplikacje. Możemy w nim umieszczać inne aplikacje, np. *xload*, który wyświetla wykres obciążenia systemu, czy też okienko *page-ra*. Dzięki temu są one zawsze widoczne i nie przeszkadzają nam na pulpicie.

Ład i spokój

Wadą systemu X Window jest niespójny interfejs. Przyciski oraz menu należące do menedżera okien wyglądają inaczej niż te w aplikacjach, a i programy różnią się wyglądem pomiędzy sobą. Jest to spowodowane faktem, że nie ma żadnej standardowej biblioteki definiującej elementy interfejsu użytkownika pod X Window; każda aplikacja robi to po swojemu.

Próbą rozwiązania tego problemu jest projekt KDE. KDE to coś więcej niż menedżer okien. Z założenia sam menedżer oraz wszystkie aplikacje napisane dla tego środowiska powinny używać biblioteki Qt, nadającej im jednolity wygląd. Ponadto KDE definiuje mechanizmy, dzięki którym różne aplikacje mogą ze sobą współpracować, także korzystając z techniki przeciągnij-upuść, znanej użytkownikom Windows. Aplikacje nie napisane specjalnie dla KDE będą działać również, nie będą jednak umiały korzystać ze specjalnie zaprojektowanych mechanizmów komunikacji.

Aplikacji KDE (a jest ich już wiele) nie konfiguruje się, edytując pliki tekstowe, ale za pomocą interfejsu graficznego. Parametry



Jako jedyny z istniejących menedżerów KDE umie komunikować się po polsku

wspólne dla całego środowiska, takie jak kolory i fonty, również można ustawić za pomocą przyjaznych menu. W pakiecie jest także przeglądarka plików (*kfm*), która, podobnie jak Eksplorator Windows, przy wybraniu jakiegoś dokumentu uruchamia skojarzony z nim program. *kfm* pozwala również na manipulowanie plikami znajdującymi się na lokalnym dysku lub na serwerze FTP oraz umie wyświetlać strony WWW.

Należy jednak zdawać sobie sprawę, że KDE jest stosunkowo „młodym” projektem i jego aplikacje nie są jeszcze do końca stabilne. Nie mają też wszystkich niezbędnych opcji, jakich można by się spodziewać po tego typu programach. Prace nad KDE jednak trwają i niewątpliwie za jakiś czas ta sytuacja się poprawi.

Ten menedżer zajmuje sporo miejsca w pamięci – przy pięciu otwartych okienkach wszystkie programy, które są automatycznie uruchamiane przez plik *opt/kde/bin/startkde*, zajmują 36,5 MB. Oczywiście większość z nich jest od razu umieszczana w pamięci wirtualnej na dysku, ale do poprawnej pracy tego menedżera wymagane są co najmniej 32 MB pamięci RAM.

Dużym plusem KDE jest możliwość ustawiania języka dla całego środowiska – wystarczy raz wybrać polski, aby wszystkie menu i większość komunikatów w aplikacjach KDE pojawiła się w języku rodzimym.

Konfiguracja tego menedżera przebiega podobnie jak w Windows – wszystkie opcje wybieramy w okienkach dialogowych. Parametry dotyczące całego środowiska można ustawić w centrum sterowania (*KDE control center*), które znajduje się menu pojawiającym się po kliknięciu znaku K, na pasku biegnącym u dołu ekranu. Właśnie tutaj, w opcji *Desktop | Language*, można wybrać język polski. Do wyświetlania

podstawy

Zestawienie plików konfiguracyjnych

twm

Plik konfiguracyjny: *\$HOME/.twmrc*
Plik domyślny:
/usr/X11R6/lib/X11/twm/system.twmrc
Program: */usr/X11R6/bin/twm*
Pełny opis: *man twm*

fvwm2

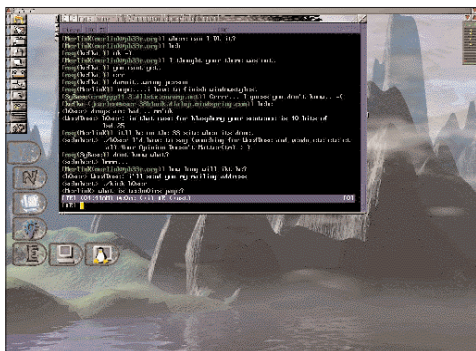
Plik konfiguracyjny: *\$HOME/.fvwm2rc*
Plik domyślny:
/usr/X11R6/lib/X11/fvwm2/.fvwm2rc
Program: */usr/X11R6/bin/fvwm2*
Pełny opis: *man fvwm2*

KDE

Program: */opt/kde/bin/startkde*

Enlightenment

Plik konfiguracyjny: *\$HOME/.enlightenment/themes/**
Plik domyślny: */usr/local/enlightenment/config/**
Program: */usr/local/enlightenment/bin/enlightenment*



Istnienie tematów umożliwia szybkie zmiany wyglądu środowiska graficznego w Linuksie

„ogonków” konieczne jest również dokonanie zamiany w menu **Style** czcionki na ISO-8859-2. Aby wszystkie działania odniosły skutek, należy, przez wybranie przycisku z ikonką X, ponownie uruchomić środowisko graficzne.

Gdy użytkownik kliknie jakąś ikonkę w *kfm*, KDE próbuje odgadnąć, jakiego rodzaju plik ona reprezentuje. Jeśli zbiór ma uprawnienia do wykonania, menedżer uruchamia go. W przeciwnym wypadku plik jest traktowany jako dokument, który należy otworzyć za pomocą jakiejś aplikacji. KDE używa specjalnej tablicy zwanej *mime types*, aby kojarzyć dokumenty z aplikacjami. By system ten działał poprawnie, trzeba określić, jakie aplikacje są zainstalowane oraz rodzaje dokumentów, które one obsługują. Tablicę skojarzeń można edytować programem **Menedżer plików (Super User Mode)**, znajdującym się w menu **K | System**. Jest to zwykły menedżer plików *kfm*, uruchamiany z uprawnieniami roota, co pozwala mu zmieniać globalne pliki konfiguracyjne.

Aby określić skojarzenia dla nowej aplikacji, wybieramy **Edit | Global Applications**. Pojawi się okienko z katalogami, z których każdy odpowiada pewnej kategorii programów. Należy wybrać jeden z nich, wejść do środka, a następnie, wybierając **New | Program**, otworzyć okno służące do tworzenia pliku opisującego nową aplikację. KDE poprosi nas o podanie nazwy pliku, która powinna mieć rozszerzenie *kdelnk*.

Plik pojawi się w okienku i można go otworzyć, klikając prawym przyciskiem myszy i wybierając **properties**. W rezultacie ukaże się dialog, który pozwoli nam wskazać, gdzie znajduje się program i odpowiadająca mu ikonka (**Execute**), oraz wybrać rodzaje dokumentów obsługiwanych przez ten program (**Application**). Po podaniu odpowiednich wartości naciska-

my **Ok**. Od tego momentu, gdy użytkownik kliknie prawym przyciskiem myszy plik tego rodzaju, program będzie się pojawiał wśród aplikacji obsługujących ten typ dokumentu.

Gra, buczy i... ma luz

W menedżerze *Enlightenment* nacisk położono na wygląd – wszystkie elementy środowiska (ramki i tytuły okienek, menu, paski) są bardzo wymyślne. Każda część ramki okna oraz pulpitu jest obrazkiem, który może zostać zamieniony na inny. W zależności od tego, co wybierzemy, mogą powstać różnorodne, bardzo efektowne wystroje. Zbiory obrazków potrzebnych do zdefiniowania wszystkich elementów menedżera nazywane są tematami (*themes*). Można je łatwo podmieniać, a w Internecie znajdują się dziesiątki gotowych zestawów (patrz ramka „info”) – w efekcie każdego dnia można pracować w środowisku o zupełnie nowym wyglądzie.

Od strony funkcjonalnej *Enlightenment* pozostawia wiele do życzenia. Dokumentacja do systemu praktycznie nie istnieje, a format plików konfiguracyjnych jest bardzo zagmatwany. Instalacja tematów jest prosta, jednak modyfikowanie ich jest bardzo trudne.

Ilość pamięci zajmowanej przez tego menedżera różni się w zależności od wybranego tematu. Domyślny zestaw obrazków wraz z pięcioma otwartymi okienkami zajmuje 2,6 MB. Jednak pod wpływem *Enlightenment* sam X-Server rozrasta się w pamięci o kolejne 4 MB.

Ponieważ *Enlightenment* nie ma systemu menu, aplikacje uruchamiane są przez wybieranie ikonki znajdujących się na pulpicie. Stanowią one część tematu i są w tym samym stylu co reszta grafiki. Stwarza to pewien problem: skąd wziąć ikonkę dla nowo zainstalowanej aplikacji? Piktogram ściągnięty z sieci, nie stworzony specjalnie dla tego tematu, będzie „klócił” się z resztą wystroju. Jedynym wyjściem jest samodzielne wykonanie ikony, ale dla większości z nas może się to okazać za trudne.

Pomimo wszystkich problemów efektowny wygląd *Enlightenment* przyciąga wielu użytkowników. Można się też spodziewać, że wiele usterek zostanie poprawionych, gdyż co kilka miesięcy pojawia się nowa wersja tego środowiska.

Przy pierwszym uruchomieniu menedżer tworzy w katalogu domowym użytkownika folder *.enlightenment*, po czym

ładuje domyślny temat, znajdujący się w kartotece *usr/local/enlightenment/config/*. Tematy pobrane z Internetu są w postaci skompresowanego archiwum, zazwyczaj z rozszerzeniem *tgz*, *tar.gz* lub *rpm*.

W dwóch pierwszych przypadkach należy wykonać następujące polecenia:

```
$ mkdir ~/.enlightenment/themes
$ cp temat.tgz ~/.enlightenment/themes/
$ cd ~/.enlightenment/themes
$ tar xzf temat.tgz
```

Jeżeli plik z tematem ma rozszerzenie *rpm*, instalujemy go komendą *rpm -i temat.rpm*. Jeśli jest to tylko aktualizacja już posiadanych zbiorów, zamiast *-i* piszemy *-u*. Program *rpm* automatycznie umieści pliki w odpowiednich miejscach.

Jeśli chcemy mieć więcej niż jeden temat i w każdej chwili móc używać dowolnego z nich, pliki każdego z nich powinniśmy umieścić we własnym folderze w katalogu *themes*. Należy zapamiętać nazwę utworzonego katalogu, gdyż jest ona potrzebna przy wyborze tematu. Dokonujemy go, edytując plik *.Xclients* i do polecenia wywołującego *Enlightenment* dopisując:

```
exec /usr/local/enlightenment/bin/
enlightenment -theme nazwa-podfolderu
```

Przy następnym uruchomieniu X Window nasz temat powinien pojawić się na ekranie.

Jacek Suraziński

info

Grupa dyskusyjna

Pytania, uwagi i komentarze do artykułu można umieścić na liście dyskusyjnej [news://news.vogel.pl/chip.zastosowania](http://news.vogel.pl/chip.zastosowania)

Internet

fvmw2: <http://www.hpc.uh.edu/fvwm/>
Enlightenment: <http://www.enlightenment.org/>
Tematy do Enlightenment: <http://enlightenment.themes.org/>
KDE: <http://www.kde.org/>
twm: <http://www.plig.org/xwinman/vtwm.html>
Inne menedżery okien: <http://www.plig.org/xwinman/>

CD 12/98 Na CD w dziale *CHIP-offline* | *Zastosowania* | *Menedżery okien dla Linuksa* znajduje się program KDE wraz ze źródłami