

Unix

A. Matuszak

22 lutego 2008

UNIX jest systemem operacyjnym, który powstał około roku 1970 w Bell Labs i został pomyślany jako niezależny od architektury komputera system operacyjny.

Około 15 lat temu fiński student Linus Torvalds napisał pierwszą wersję darmowej¹ odmiany UNIXa nazwaną LINUX na popularną platformę komputerów PC opartą o procesor Intel x86, często skrótowo oznaczaną i386.

LINUX nie jest jednolitym oprogramowaniem, występuje w wielu odmianach zwanych dystrybucjami. Główne różnice między dystrybucjami polegają na domyślnej konfiguracji programów (a więc np. na wyglądzie) oraz sposobem zarządzania instalacją i konfiguracją oprogramowania. Często tworzone są też specjalne dystrybucje przeznaczone do specyficznych celów.

Najbardziej znane dystrybucje to: Slackware, Fedora (RedHat), Debian, Knoppix, Gentoo.

Login. LINUX (UNIX) jest systemem wielodostępnym, wieloprotocowym. Wielodostępność oznacza, że więcej niż jeden użytkownik może w tym samym czasie pracować na jednym komputerze. Wieloprotocowość oznacza, że więcej niż jeden program może być jednocześnie wykonywany. Z uwagi na wielodostępność, system operacyjny musi w jakiś sposób rozróżniać użytkowników. Dlatego każdy użytkownik otrzymuje unikalną nazwę użytkownika (identyfikator, ang. *username*).

Aby rozpocząć pracę w systemie należy zgłosić się (zalogować). W polu login należy wpisać przydzieloną przez administratora nazwę użytkownika a w polu password wpisać przydzielone przez administratora hasło.

Menadżer okien. Pomyślne zalogowanie powoduje otwarcie sesji graficznej. Na ekranie pokazuje się układ elementów tworzący środowisko sesji graficznej. W systemach UNIX/LINUX grafiką zarządza biblioteka zwana *X Window System* (X11R6).

Nieco inna jest koncepcja organizacji sesji graficznej. Samym procesem wyświetlania okien zarządza biblioteka *X window system*, natomiast ich (zewnętrznym) wyglądem i sposobem obsługi, reakcją na klawiaturę oraz myszkę zarządza osobny program zwany menadżerem okien (*window manager*). Skutkiem takiego podziału jest różnorodność wyglądu środowiska graficznego na różnych komputerach. Z jednej strony pozwala to dobranie takiego wyglądu (i funkcjonalności!), który najbardziej odpowiada użytkownikowi, z drugiej początkujący użytkownik może być lekko zmieszany bo za każdym razem ekran może wyglądać inaczej.

¹Rozpowszechnianej w oparciu o swobodną licencję GNU

Na szczęście, poza często bardzo różnym wyglądem, zasadniczy sposób obsługi jest na tyle zbliżony dla każdego menadżera okien, że stykając się z zupełnie nowym można, bez trudu domyśleć się zasad podstawowej obsługi.

IceWM. Zachowanie menadżera okien opisujemy na przykładzie programu o nazwie IceWM.

Sam ekran po uruchomieniu jest pusty (o ile domyślenie nie otwierają się jakieś programy) a jedynie u dołu jest widoczny pasek.

W lewym rogu paska jest umieszczony przycisk, który uruchamia menu programów dla IceWM. Górne pozycje tego menu są bardzo zależne od lokalnej konfiguracji, natomiast ważne są ostatnie dwie pozycje na dole związane z zarządzaniem sesją IceWM.

Mamy tam pozycje **Themes** oraz **Logout...** **Themes** odpowiada za wybór stylu dekoracji okien i można sobie dobrać go stosownie do swoich gustów, natomiast w rozwijalnym w poziomie **Logout...** można wybrać zamknięcie sesji IceWM co jest równoznaczne z zakończeniem pracy.

Dalsze przyciski są przypisane do programów, które w ten sposób można szybko uruchamiać, np. ikona symbolizująca telewizorek uruchamia program **xterm**.

Po ikonach programów, na pasku dolnym, występują przyciski z kolejnymi numerami (najczęściej cztery). Odpowiadają one tzw. wirtualnym ekranom, służącym do uporządkowania sobie pracy. Okno (np. przeglądarki internetowej) uruchomionej na jednym ekranie pozostanie na nim, przełączając na inny ekran możemy tam pogrupować okna związane z innym zadaniem.

Z prawej strony są ozdobniki w postaci wykresów pokazujących obciążenie systemu, aktualny transfer karty sieciowej itp. oraz na samym końcu paska dolnego, okienko pokazujące bieżący czas.

Okno. Okno jest prostokątnym obszarem, którego zawartość określa program, do którego to okno należy (tak więc w oknie przeglądarki internetowej będzie strona www a w odtwarzaczu mp3 będą przyciski do sterowania odtwarzaniem muzyki) tu natomiast warto się na chwilę skupić na ramce okna. Niezależnie od rodzaju programu krawędź jest identyczna, gdyż to jest cech menadżera okien.

Przesuwając powoli kursor myszki na krawędź okna można zauważyć, że na samej ramce kursor zmienia kształt, sygnalizując możliwość zmiany wymiarów okna. Na pionowych i poziomych krawędziach możemy zmienić odpowiednio rozmiar poziomy lub pionowy. W narożach mamy możliwość zmiany obu wymiarów jednocześnie.

Oprócz ramki, każde okno posiada u góry listwę. Uchwycenie kursorem myszki listwy (trzymając przyciśnięty lewy przycisk myszki) pozwala przesunąć okno po ekranie. Przyciśnięcie prawego przycisku myszki na listwie powoduje wywołanie menu dotyczącego okna.

Na górze okna znajdują się również przyciski. Przycisk z lewej strony wywołuje menu. Natomiast po prawej stronie znajduje się kilka przycisków (najczęściej trzy w zależności od wyboru motywu (*theme*)). Jeden przycisk minimalizuje okno (zamyka do ikony widocznej na pasku dolnym), drugi maksymalizuje rozmiar okna (powtórny wybór przywraca początkowy rozmiar). Ostatni przycisk, często oznaczany krzyżykiem służy do zabijania okna. Nie należy go używać, gdyż jest to bardzo brutalny sposób przerywania działania programu.

Należy korzystać z mechanizmów kończenia pracy właściwej dla danego programu. Przycisk z krzyżykiem jest jedynie przeznaczony do sytuacji awaryjnych, kiedy np. program nie odpowiada.

Xterminal. W systemie UNIX/LINUX istnieje silna tradycja pracy w tzw. trybie terminalowym, czyli porozumiewanie się z komputerem poprzez wpisywanie poleceń z klawiatury oraz odpowiedzi komputera w postaci wypisywania tekstowych wyników.

Współcześnie taki tryb może wydawać się anachronizmem jednak okazuje się być wielokrotnie wydajniejszy od pracy przy pomocy klikania myszą. Z reguły im bardziej zaawansowany użytkownik tym częściej korzysta z narzędzi tekstowych, nawet mając do dyspozycji ładne i atrakcyjne odpowiedniki graficzne tychże narzędzi.

Historycznie terminal był skrzyżowaniem maszyny do pisania z drukarką (w tych czasach nie było jeszcze monitorów). Współcześnie używamy programu `xterm` (od *X terminal*) czyli okienka w którym można wpisywać polecenia a system wypisuje tekst będący wynikiem polecenia. Jest to odpowiednik historycznego terminala. Takich okienek możemy używać naraz dowolną ilość a każde okienko terminalowe tworzy osobną sesję tekstową (wieloprocusowość).

Aby uniknąć żmudnego wpisywania długich i skomplikowanych poleceń wymyślono wiele ułatwień, które czynią wygodniejszą i mniej podatną na pomyłki przy pisaniu.

Wpisując jako argument (o tym potem) nazwę pliku wystarczy wpisać tyle początkowych liter ile jednoznacznie identyfikuje nazwę pliku a następnie nacisnąć znak tabulacji a reszta sama się dopisze. W ten sposób można używać długich nazw plików bez konieczności wpisywania całej nazwy za każdym razem i bez możliwości omyłkowego wpisania `plik` zamiast `plik`.

Każde wykonane polecenie jest pamiętane jako historia i można je przywrócić poprzez strzałkę do góry i ponownie wykonać naciskając Enter. Jeśli chcemy wykonać nie takie samo polecenie ale bardzo podobne, możemy przywrócić poprzednie a następnie wyedytować je tak jakby to był wiersz tekstu, czyli przesuwamy się w lewo i w prawo strzałkami, usuwamy znaki poprzez Backspace bądź Delete oraz dopisujemy znaki w miejscu gdzie stoi kursor.

Jeśli wydamy polecenie, to linia poleceń terminala jest zajęta aż do zakończenia wykonywania polecenia. Czasami jest to niewygodne, gdyż uruchamiamy programy, które mają długo pracować. W takiej sytuacji możemy zlecić wykonanie programu jako proces drugoplanowy (w tle), np.:

```
gedit &
```

Znaczek `&` dodany na końcu² spowoduje, że otworzy się okienko z edytorem `gedit` a linia poleceń będzie gotowa na przyjęcie następnych poleceń.

Podstawowe komendy. Z punktu widzenia użytkownika podstawowym zadaniem jakie spełnia system operacyjny jest uruchamianie programów. Program uruchamia się wpisując w linii poleceń³ jego nazwę np.:

```
octave
```

²Nie można wykonywać w ten sposób wszystkich poleceń.

³Uruchomienie programu może też odbywać poprzez naciśnięcie odpowiedniego przycisku, bądź wybranie odpowiedniej pozycji z menu. Jednak są to tylko skróty, które wywołują program poprzez jego nazwę.

Uruchomiany program może otrzymać parametry, np.:

```
octave -q skrypt.m
```

Argumenty ze znakiem - zwane są opcjami i zostaną opisane później. Pozostałe argumenty są najczęściej nazwami plików zawierających dane dla danego programu. Należy zaznaczyć, że poprzez „dane” rozumieć należy nie tylko kolumny liczb, z którymi się kojarzą ale również zapisany dźwięk (audio, np. .mp3), obraz nieruchomy (np. .jpeg) bądź ruchomy (np. .avi).

Ponieważ w przeważającej mierze wszelkie dane zawarte są w plikach, więc w konsekwencji drugą funkcją systemu operacyjnego jest dokonywanie operacji na plikach. Podstawowymi operacjami na plikach są:

1. kopiowanie
`cp plik1 plik2`
2. usuwanie
`rm plik`

Ten zestaw operacji na plikach byłby wystarczającym minimum, gdyby nie fakt, że gdyby wszystkie pliki na dysku (a może ich być setki tysięcy) były w jednym miejscu to praktycznie uniemożliwiłaby to korzystanie z nich. Dlatego praktycznie wszystkie współczesne systemy komputerowe pozwalają organizować pliki w drzewiastą strukturę katalogów. Daje to możliwość grupowania plików i zapanowania nad chaosem. Dlatego pojawia się koncepcja kartoteki (katalogu, folderu) w której mogą znajdować się nie tylko pliki ale również kartoteki.

Użytkownik w każdym momencie swojej pracy znajduje się w jakiejś kartotece i ta kartoteka nazywa się kartoteką bieżącą.

System operacyjny dostarcza poleceń, które pozwalają operować na kartotekach. Minimum tych poleceń to:

1. tworzenie kartoteki⁴
`mkdir kartoteka`
2. usuwanie kartoteki
`rmdir kartoteka`
3. wyświetlanie zawartości kartoteki
`ls`
4. zmiana kartoteki
`cd kartoteka`
ważną i potrzebną operacją jest przejście do kartoteki nadrzędnej, polecenie to można wykonać skrótem:
`cd ..`

Początkowo, kartoteką bieżącą każdego użytkownika jest tzw. kartoteka domowa, dla każdego użytkownika inna.

⁴Warto zwrócić uwagę, że nie istnieje operacja tworzenia pliku (pliki tworzą programy a nie system operacyjny), natomiast istnieje polecenie tworzenia kartoteki.

Ścieżki i nazwy plików System operacyjny rozróżnia wielkie i małe litery, więc nazwy `Plik`, `plik` oraz `pLIK` dotyczą trzech różnych plików. Formalnie nie ma podziału na nazwę i rozszerzenie. Z punktu widzenia systemu operacyjnego kropka jest takim samym znakiem jak każdy inny⁵, więc dozwolona jest nazwa typu `pl...ik`.

Rozszerzenia w nazwach plików są używane przez użytkowników, aby łatwo mogli się zorientować w zawartości plików. Również niektóre programy czasami wymagają określonych rozszerzeń (np. `octave` poszukuje przy wywołaniu funkcji pliku o nazwie takiej jak nazwa funkcji i rozszerzeniu `.m`)

W nazwie oprócz liter (małych i wielkich) oraz cyfr mogą występować dowolne znaki, jednak niektóre z nich mają specjalne znaczenie i początkującym należy odradzać ich użytkowanie. W szczególności kłopoty sprawi użycie znaku odstępu (spacji) w nazwie pliku. Na pewno bezpieczniejsze jest użycie kropki, znaku podkreślenia (`_`) lub znaku minus (`-`).

Pełna ścieżka dostępu do pliku określa w sposób jednoznaczny jego położenie. Może ona wyglądać np.:

```
/usr/local/sources/ala
```

Jest to ciąg kartotek oddzielonych znakiem lewego ukośnika `/` (slash) w odróżnieniu od znaku prawego ukośnika `\` (backslash), który służy do odbierania znakom specjalnego znaczenia.

Druga rzecz warta uwagi, to fakt, że ścieżka nie zawiera na początku litery oznaczającej stację czy partycję dyskową, gdyż drzewo plików w systemie UNIX jest jedno, niezależnie ile i jakich nośników przechowuje pliki.

Każdy następny nośnik jest montowany (polecenie `mount`), czyli drzewo plików, które nośnik zawiera jest dołączane do całego drzewa plików w podanej kartotece, zwanej punktem montowania.

Opcje. Większość programów akceptuje opcje czyli takie argumenty, które w pewnym stopniu zmieniają działanie programu. Program `ls` wyświetla nazwy plików znajdujących się w bieżącej kartotece. Ten sam program wywołany z opcją `-l` (od *long*) czyli polecenie `ls -l` wyświetli oprócz nazw plików, dodatkowe informacje o każdym pliku, np. prawa dostępu, właściciel, grupa, czas modyfikacji i rozmiar.

Zazwyczaj obowiązuje konwencja, że opcje można dowolnie łączyć i przedstawiać, tak więc polecenia:

```
ls -l -r -t
```

```
ls -t -l -r
```

```
ls -lrt
```

są równoważne.

Oprócz opcji krótkich, w postaci jednej litery (znaku) istnieją też opcje długie, w postaci całego słowa i te najczęściej pisze się z dwoma znakami - np.:

```
octave --version
```

W przypadku programu `octave` najwygodniej jest uruchamiać program z opcją `-q` (*quiet*) co zapobiega wyświetlaniu długiego komunikatu wstępnego.

⁵Za wyjątkiem sytuacji kiedy nazwa zaczyna się od kropki.

gedit. Edytor tekstowy⁶ nie należy do systemu operacyjnego, jednak jest niezbędny aby móc w praktyce pracować w systemie typu UNIX.

W systemie dostępnych jest bardzo wiele programów do edycji tekstu, m.in. vi, emacs, jed oraz wręcz setki innych. Tutaj opiszemy pracę z edytorem **gedit**.

Edytor uruchamiamy poleceniem:

```
gedit &
```

(znaczek **&** powoduje zwolnienie linii poleceń). Na ekranie otwiera się okno edytora. Edytor zawiera czytelne menu graficzne, więc nie ma sensu opisywanie pozycji w menu. Skupimy się raczej na elementach, które nie są łatwo dostępne a bywają przydatne.

Edytor może być uruchomiony z argumentem w postaci nazwy pliku, który zostanie otwarty i będzie gotowy do edycji:

```
gedit skrypt.m &
```

Edytor, po rozszerzeniu rozpoznaje typ pliku i dla popularnych klas plików koloruje tekst, co bardzo ułatwia pracę np. nad tekstem programu. Tak więc plik z rozszerzeniem **.m** zostanie pokolorowany zgodnie z regułami składni programu **octave**.

W przypadku, kiedy rozszerzenie nie jest typowe (a jak powiedzieliśmy, rozszerzenia służą tylko użytkownikom i ktoś może chcieć oznaczać pliki octave rozszerzeniem **.exe**) możemy wymusić kolorowanie według określonej składni poprzez wybór:

Widok → Tryb wyróżniania elementów składni → Skrypty → Octave

Edytor **gedit** (jak większość edytorów) potrafi się ustawić we wskazanej linii. Jest to wyjątkowo użyteczne, kiedy octave wykonując skrypt sygnalizuje błąd np. w linii 123. Wtedy polecenie:

```
gedit +123 skrypt.m &
```

uruchamia **gedit**, otwiera plik **skrypt.m** i ustawia kursor w linii 123.

Ćwiczenie* Załóż kartotekę o nazwie będącej numerem bieżącego roku, a w niej dwanaście kartotek nazwanych rzymskimi numerami miesięcy. W wybranym miesiącu załóż kartoteki z numerami dni w które wypadają niedziele.

Polecenie wpisane w terminalu:

```
cal 2008
```

wypisuje skrócony kalendarzyk dla roku 2008.

⁶Czyli program, który umożliwia edycję tekstu ASCII. Nie należy tego mylić z programami typu Word, które są wordprocessorami.