Rozdział 3. Linux **100**

3.1. Pierwsze uruchomienie 101

3.2. Powłoki 103

3.3. Struktura systemu 104

3.4. Terminal 105

3.5. Połączenie z siecią bezprzewodową 107

3.5.1. Zapora sieciowa (firewall) 108

3.6. Podstawowe polecenia w trybie tekstowym 111

3.6.1. Składnia poleceń 112

3.6.2. Zarządzanie katalogami 112

3.6.3. Znaczenie znaków 114

3.7. Prawa dostępu 114

3.8. Zarządzanie użytkownikami i grupami 117

3.9. Procesy 121

3.9.1. Stany procesów 121

3.9.2. Polecenia służące do zarządzania procesami 121

3.10. Zarządzanie pakietami, źródłami, archiwami 123

3.11. Skrypty 126
3.11.1. Tworzenie skryptu 126
3.11.2. Uruchamianie skryptu 127
3.11.3. Instrukcje warunkowe i iteracyjne w skryptach 129
3.11.4. Wywoływanie skryptu z parametrami 130
3.12. Okna dialogowe 132
3.12.1. Składnia poleceń 133
3.13. Informacje sieciowe, komunikacja 136
3.13.1. Informacja o interfejsach sieciowych 136
3.13.2. Wyświetlenie aktywnych połączeń 136
3.13.3. Wyświetlenie bramy domyślnej 136
3.13.4. Komunikacja między użytkownikami 137
3.14. Serwer WWW 137
3.15. Tworzenie dysków startowych z pamięci USB 139
3.16. Instalacja Ubuntu obok Windowsa 140
3.16.1. Zmiana kolejności uruchamiania systemów 142
3.17. Sprawdź się — zadania na koniec rozdziału 143

Rozdział 3.

LINUX

„Wydanie Linuksa na GPL[3] było zdecydowanie najlepszą rzeczą, jaką uczyniłem kiedykolwiek"[4]
— Linus Torvalds

Linux to wielozadaniowy, stabilny system operacyjny, którego twórcą jest Linus Torvalds. Głównym przesłaniem autora było tworzenie i darmowe udostępnianie systemu wraz z kodami źródłowymi. Dzięki temu nad Linuksem oraz oprogramowaniem dla niego pracuje wielu programistów na całym świecie, a Linus Torvalds otrzymał prestiżowe wyróżnienie *Millennium Technology Prize* przyznawane przez *Technology Academy Finland* za wybitny wkład w prace nad jądrem Linux i stworzeniem wolnego systemu operacyjnego.

Nazwa Linux dotyczy właściwie jądra systemu (ang. *kernel*), chociaż używa się jej jako nazwy całego systemu.

Definicja

Kernel (jądro systemu) odpowiada za komunikację systemu z urządzeniami (odpowiednikami sterowników są moduły), uruchamianie i współpracę programów (steruje procesami), system plików, zarządzanie pamięcią.

Jądro wraz z oprogramowaniem jest nazywane dystrybucją Linuksa. Istnieją różne dystrybucje. Wybór zależy na ogół od indywidualnych upodobań użytkownika. My będziemy bazować na Ubuntu 12.04.1 — desktop. Obraz systemu można pobrać ze strony *ubuntu.pl*.

Celem tego rozdziału nie jest przekonanie do korzystania z nowoczesnego, stabilnego systemu i darmowych aplikacji, tylko pokazanie, jak to robić.

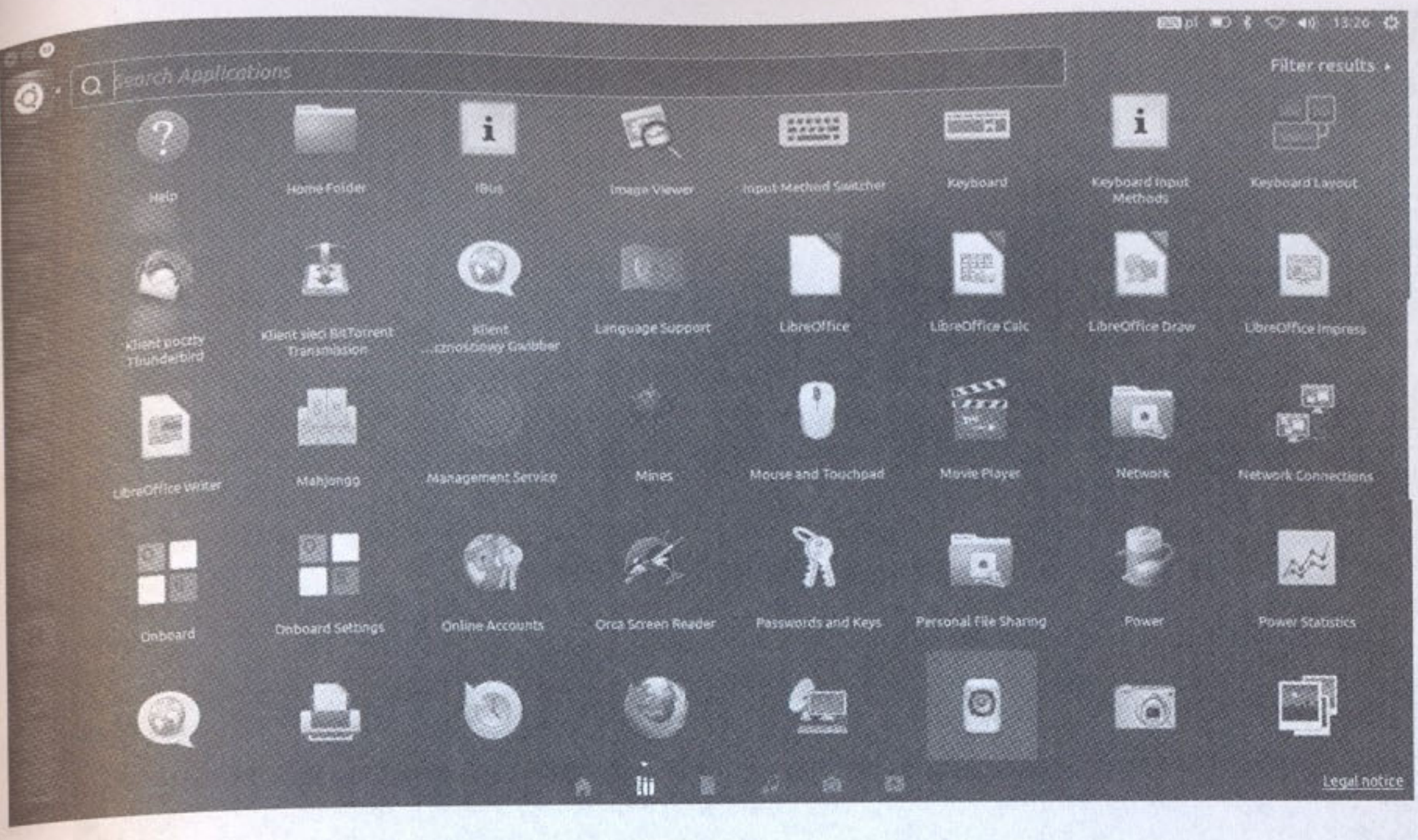
Linux może być zainstalowany jako samodzielny system operacyjny, obok innego systemu, lub uruchamiany z płyty czy innego nośnika zewnętrznego.

Niewątpliwą zaletą Linuksa jest pełna gotowość do pracy tuż po instalacji. Już od pierwszego uruchomienia system jest wyposażony w bogaty zestaw najczęściej używanych aplikacji — patrz rysunek 3.1.

---

3 GNU GPL — licencja wolnego i otwartego oprogramowania.

4 Źródło: Hiroo Yamagata, *The Pragmatist of Free Software: Linus Torvalds Interview* (28 sierpnia 2006).

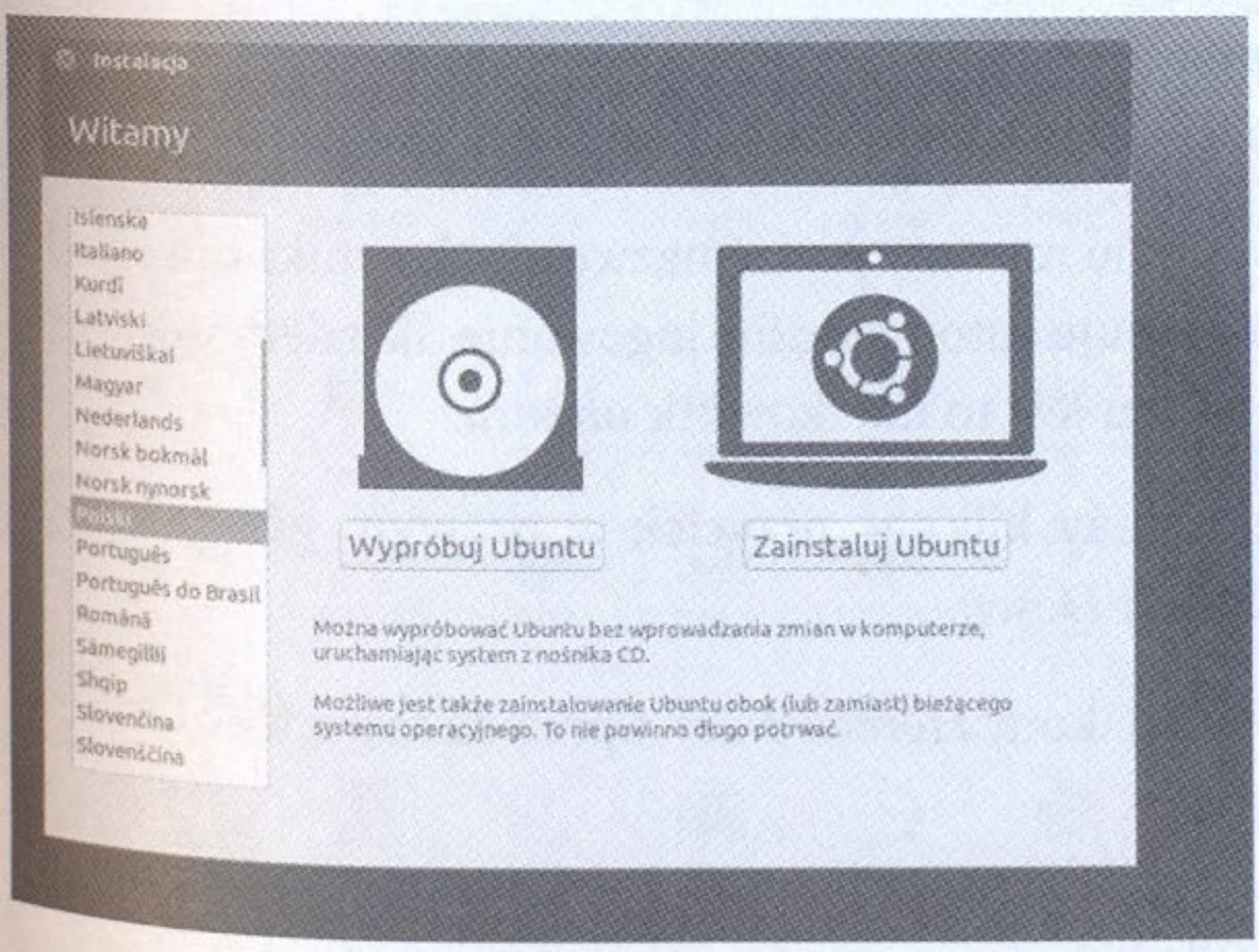


Rysunek 3.1. Przykładowe aplikacje w Linuksie

Po włączeniu komputera system uruchamia się w trybie graficznym (w tym przypadku jest to *Unity*), w którym można pozostać do końca pracy. Jednak znajomość trybu tekstowego jest przydatna w pracy zdalnej i pomaga w zrozumieniu systemu.

3.1. Pierwsze uruchomienie

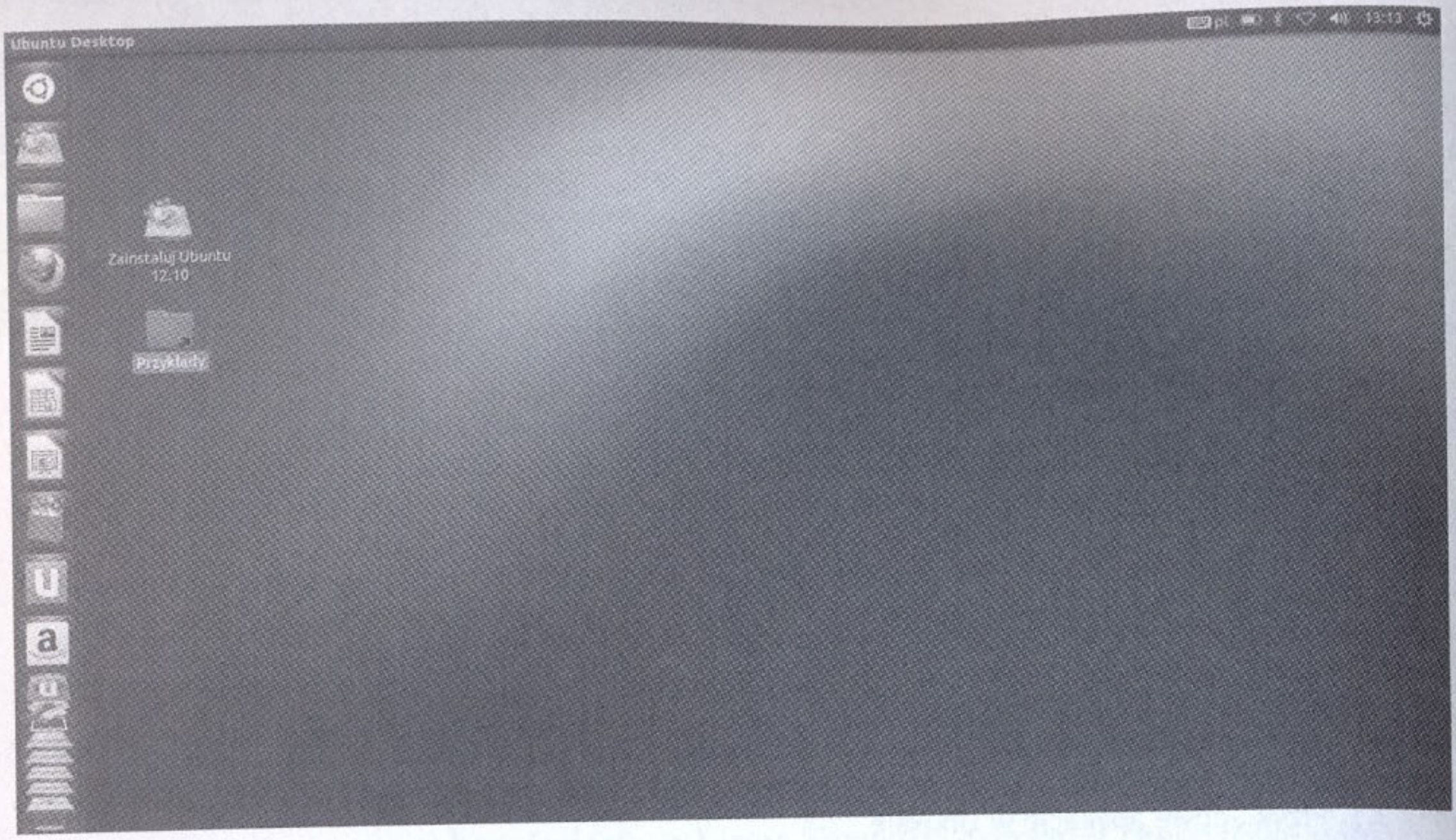
Po uruchomieniu systemu z płyty pojawia się ekran powitalny (rysunek 3.2).



Rysunek 3.2. Ekran powitalny

Z listy dostępnych wersji językowych wybieramy *Polski*, a następnie klikamy *Wypróbuj Ubuntu*.

Pojawia się pulpit użytkownika (rysunek 3.3).



Rysunek 3.3. Pulpit użytkownika

Zestaw najczęściej uruchamianych aplikacji mamy na pasku po lewej stronie. Do uruchomienia wystarczy jednokrotne kliknięcie lewym przyciskiem myszy.

Poruszanie się po systemie jest intuicyjne. Lewy i prawy przycisk myszy mają podobne zastosowanie jak w systemie Windows. Inaczej natomiast ułożone są przyciski do sterowania oknami — w lewym górnym rogu okna zamiast tradycyjnie (dla użytkowników Windows) w prawym.

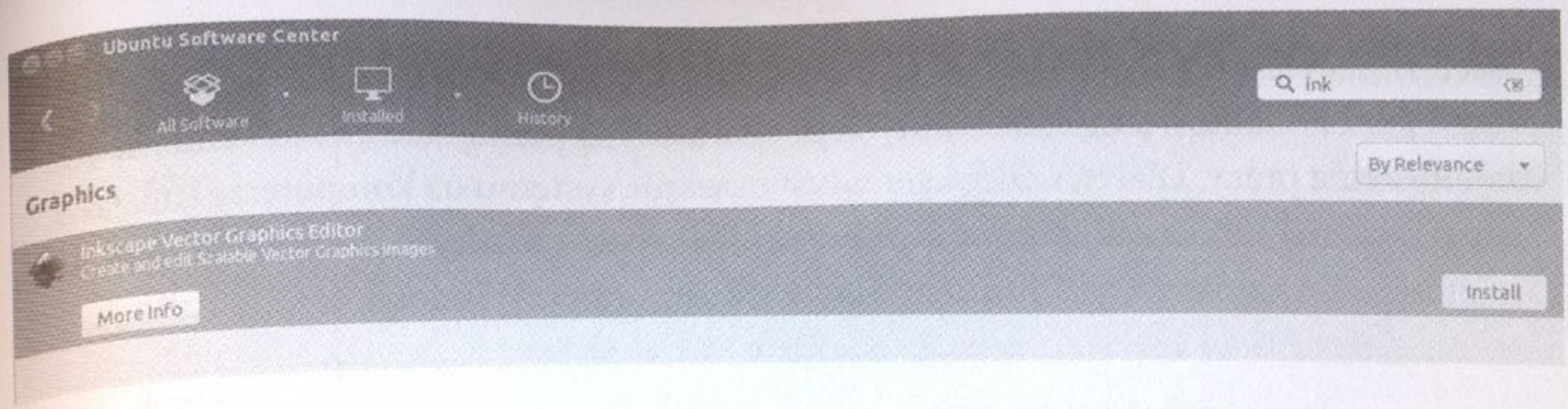
Przycisk w lewym górnym rogu pulpitu to *Dash Home*. Po kliknięciu go otwiera się panel ze spisem zainstalowanych i dostępnych do ściągnięcia programów.

Standardowo przy logowaniu się do systemu musimy podać nazwę użytkownika oraz hasło. W wersjach uruchamianych z płyty następuje automatyczne logowanie dla zdefiniowanego wcześniej użytkownika. W tym przypadku jest to użytkownik *ubuntu*.

Żeby zakończyć pracę w systemie, należy kliknąć przycisk w prawym górnym rogu i z pojawiającego się menu wybrać *Shut Down*.

Zadanie 3.1 Na pasku narzędzi poszukaj ikony *Home Folder*. Sprawdź zawartość tego folderu. Utwórz w nim katalog *Moje skrypty*.

Zadanie 3.2 Na pasku ikony poszukaj *Ubuntu Software Center*. Sprawdź, jakie programy są zainstalowane. Spróbuj zainstalować program graficzny Inkscape (bardzo dobry program do tworzenia grafiki wektorowej, oczywiście *open source*). Patrz rysunek 3.4.



Rysunek 3.4. Instalacja Inkscape

3.2. Powłoki

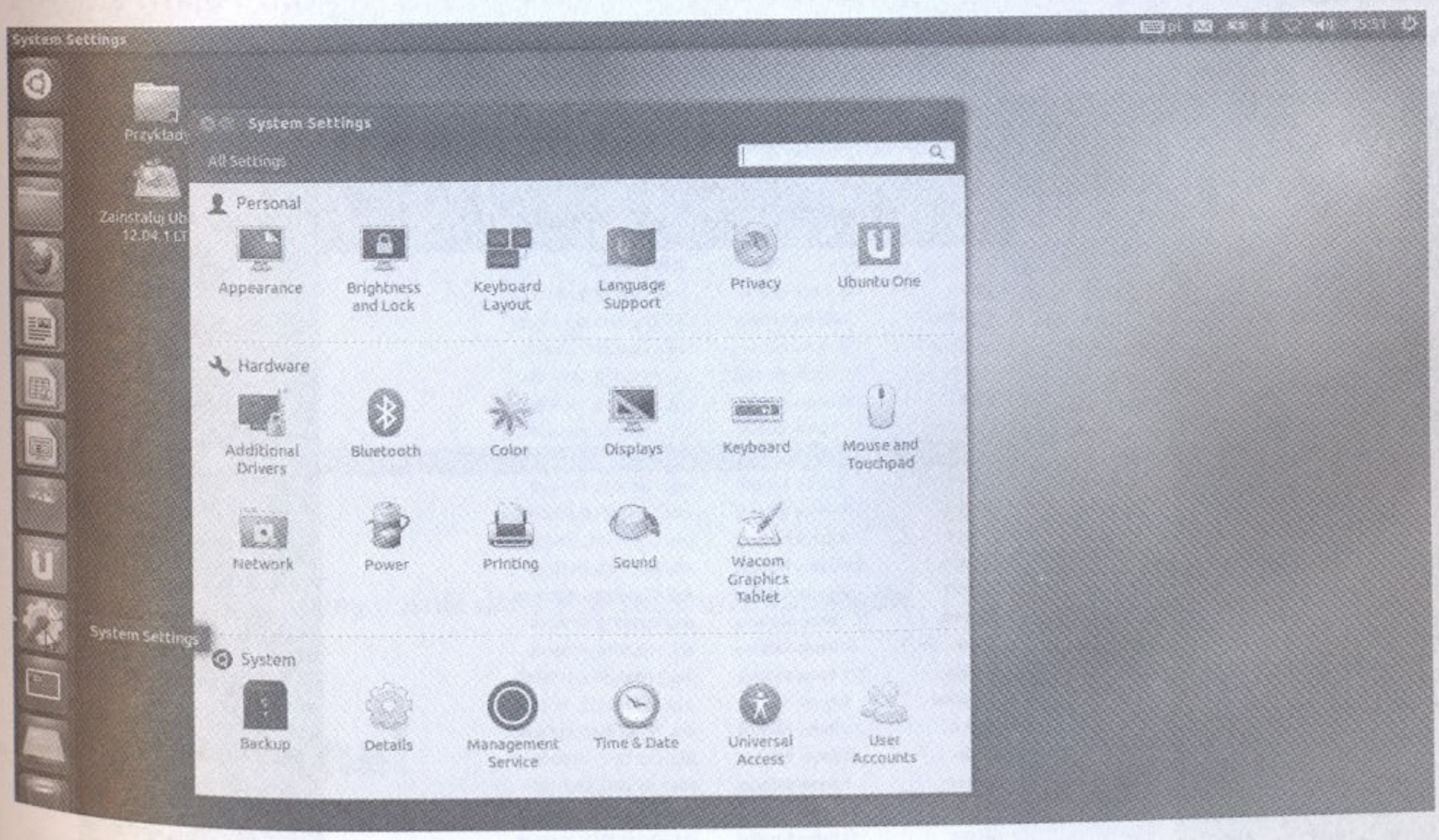
Do komunikacji użytkownika z komputerem służy interpreter poleceń. W Linuksie nosi on nazwę powłoki systemowej.

Standardowo powłoki mogą być:

- graficzne (X Windows system w Linuksie),
- tekstowe (shell w Linuksie).

Istnieje wiele powłok systemowych. W trybie tekstowym będziemy korzystać z najpopularniejszej — *bash* (Bourne Again SHell), a w trybie graficznym — z *GNOME* z nakładką *Unity*.

Personalizacja ustawień komputera jest intuicyjnie prosta — co widać na rysunku 3.5.



Rysunek 3.5. Personalizacja ustawień komputera

Zadanie 3.3 Uruchom aplikację System Settings. Ustaw wszystkie parametry pulpitu zgodnie ze swoimi potrzebami.

Jeżeli system jest uruchomiany z płyty, wprowadzone zmiany dotyczące zarówno ustawień, jak i dodatkowo zainstalowanych programów są pamiętane tylko do momentu zakończenia pracy. Dlatego zaleca się zainstalowanie systemu na komputerze (np. jako drugi system) lub uruchamianie z pamięci USB (patrz rozdział „Tworzenie dysków startowych z pamięci USB" lub „Instalacja Ubuntu obok Windows").

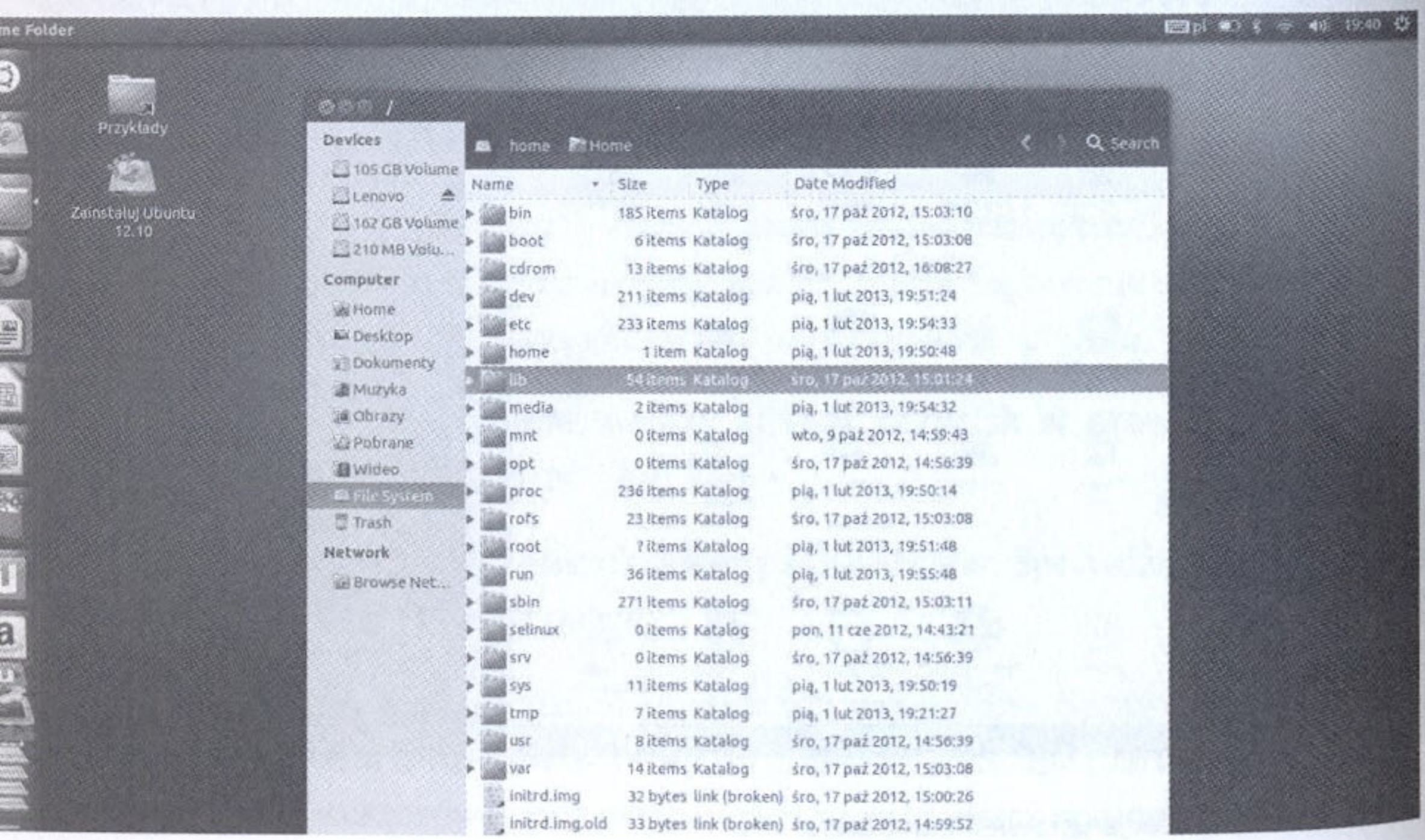
.3. Struktura systemu

ozmieszczenie plików i katalogów w systemie oraz ich przeznaczenie definiuje FHS (*Filesys-m Hierarchy Standard*). Dzięki temu cała struktura jest przejrzysta.

ystem plików zaczyna się od katalogu głównego (*root*) — /. Urządzenia fizyczne są dołą-zane jako katalogi.

Nazwa pliku może być dowolnym 256-bajtowym ciągiem znaków (oczywiście z wyjątkiem „/"). Dlatego nazywając pliki, których zamierzamy używać w systemie Windows, musimy pamiętać zarówno o długości nazwy, jak i dopuszczalnych symbolach.

adanie 3.4 Kliknij ikonę *Home Folder*. Następnie wybierz *File System*. Pojawi się lista andardowych katalogów katalogu głównego (*root*). Patrz rysunek 3.6.

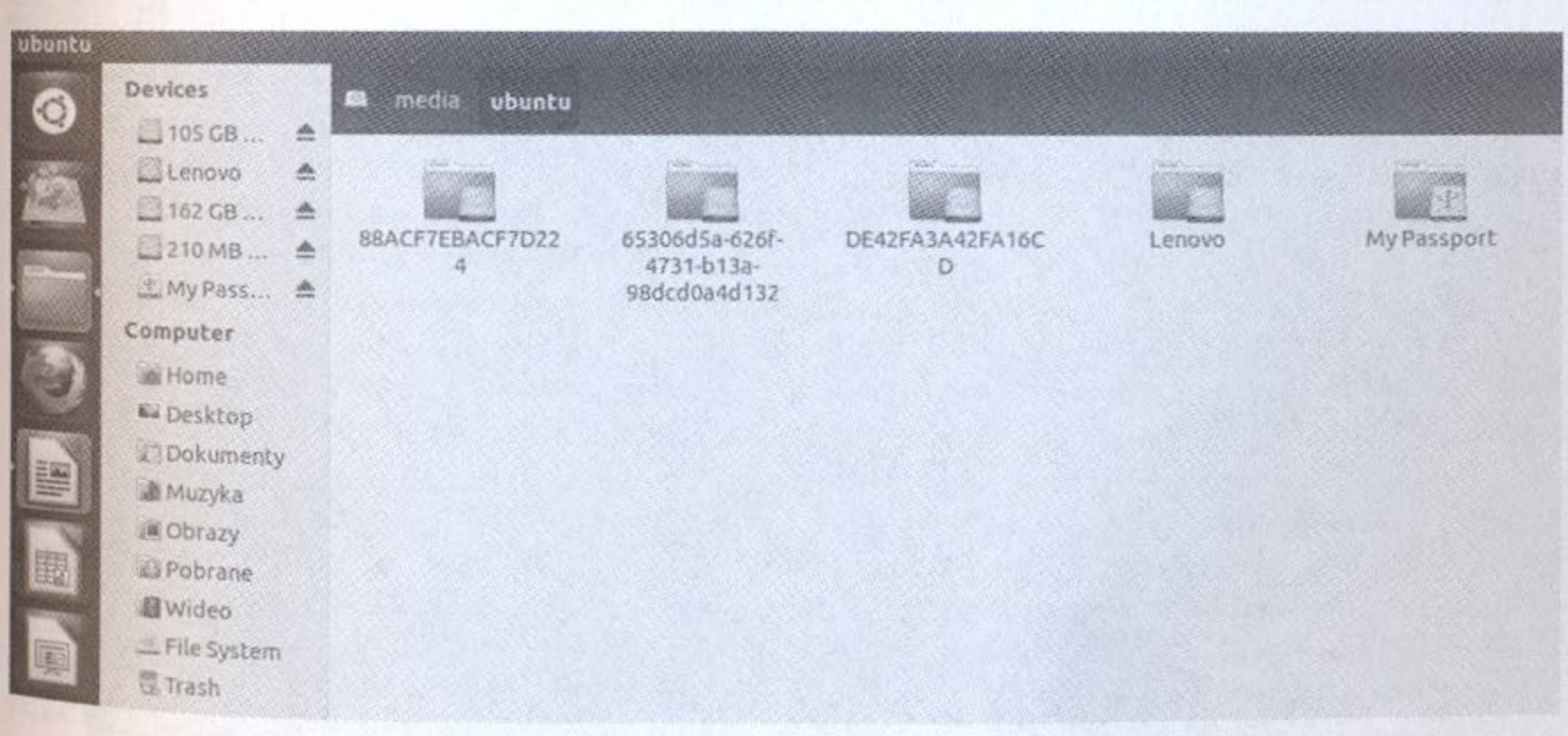


ysunek 3.6. Zawartość katalogu głównego (root)

Poszukaj w internecie informacji na temat przeznaczenia katalogów i uzupełnij tabelkę.

Nazwa katalogu	Przeznaczenie
/bin	
/boot	
/dev	
/etc	
/home	
/lib	
/media	
/mnt	
/proc	
/root	
/sbin	
/sys	
/tmp	
/usr	
/var	

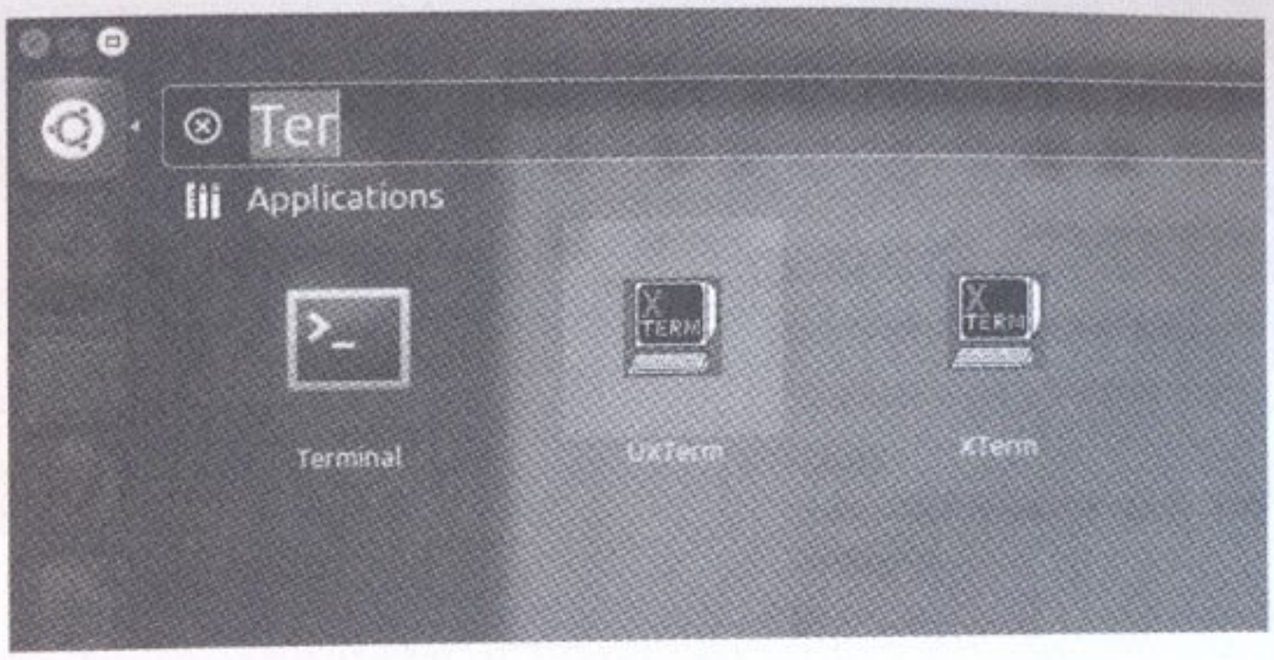
Zadanie 3.5 Podłącz zewnętrzny nośnik danych. Jego symbol pojawi się automatycznie na pasku z lewej strony. Urządzenie jest gotowe do pracy bez potrzeby instalacji. Zlokalizuj je w drzewie katalogów (rysunek 3.7).



Rysunek 3.7. Zewnętrzny nośnik danych w strukturze katalogów

3.4. Terminal

Do korzystania z linii poleceń (trybu tekstowego) wykorzystujemy *Terminal* (emulator konsoli), który uruchamiamy skrótem *Ctrl+Alt+T* lub wyszukujemy w *Dash Home* (rysunek 3.8).



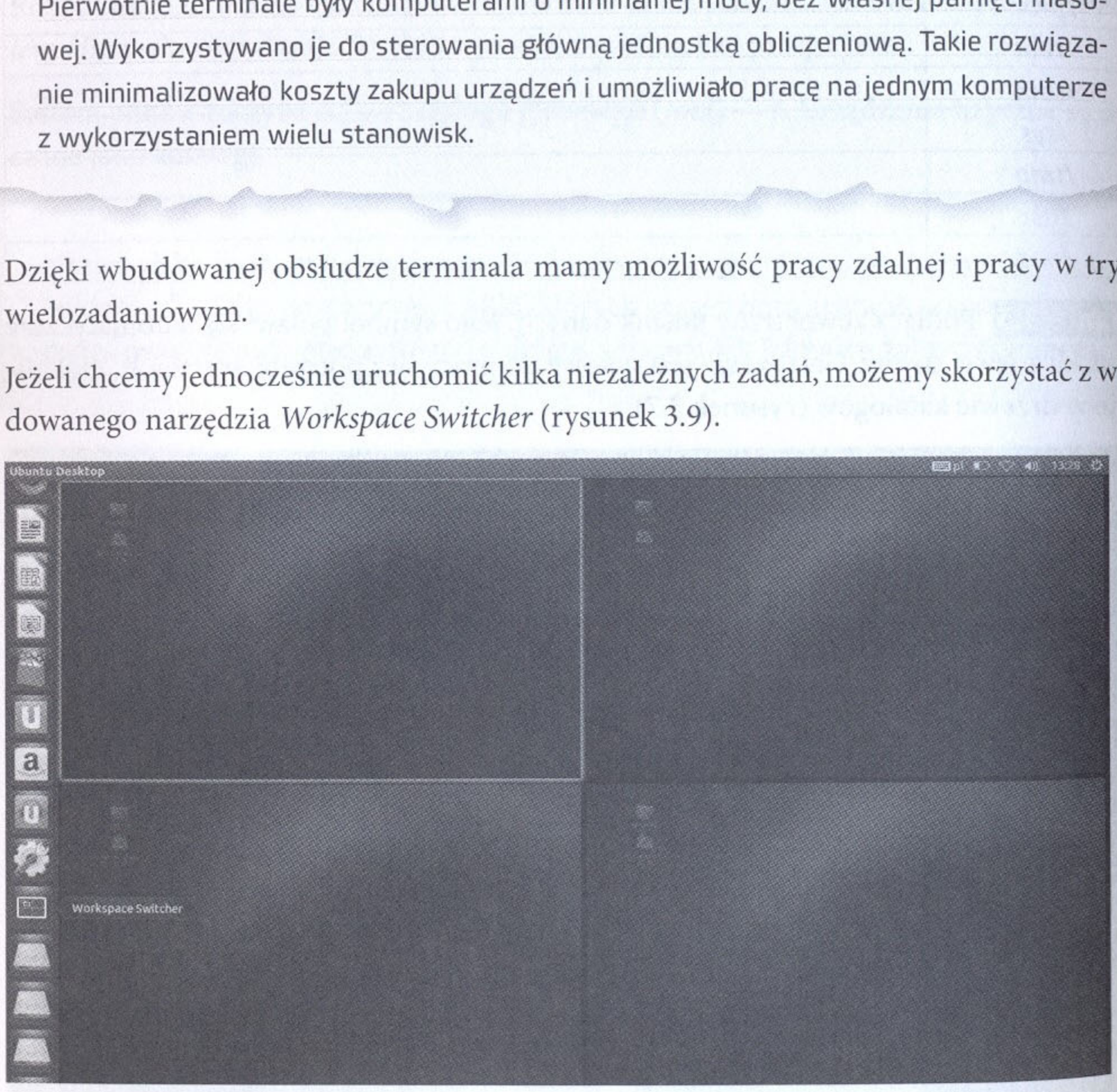
Rysunek 3.8. Uruchamianie terminala

Ciekawostka

Pierwotnie terminale były komputerami o minimalnej mocy, bez własnej pamięci masowej. Wykorzystywano je do sterowania główną jednostką obliczeniową. Takie rozwiązanie minimalizowało koszty zakupu urządzeń i umożliwiało pracę na jednym komputerze z wykorzystaniem wielu stanowisk.

Dzięki wbudowanej obsłudze terminala mamy możliwość pracy zdalnej i pracy w trybie wielozadaniowym.

Jeżeli chcemy jednocześnie uruchomić kilka niezależnych zadań, możemy skorzystać z wbudowanego narzędzia *Workspace Switcher* (rysunek 3.9).

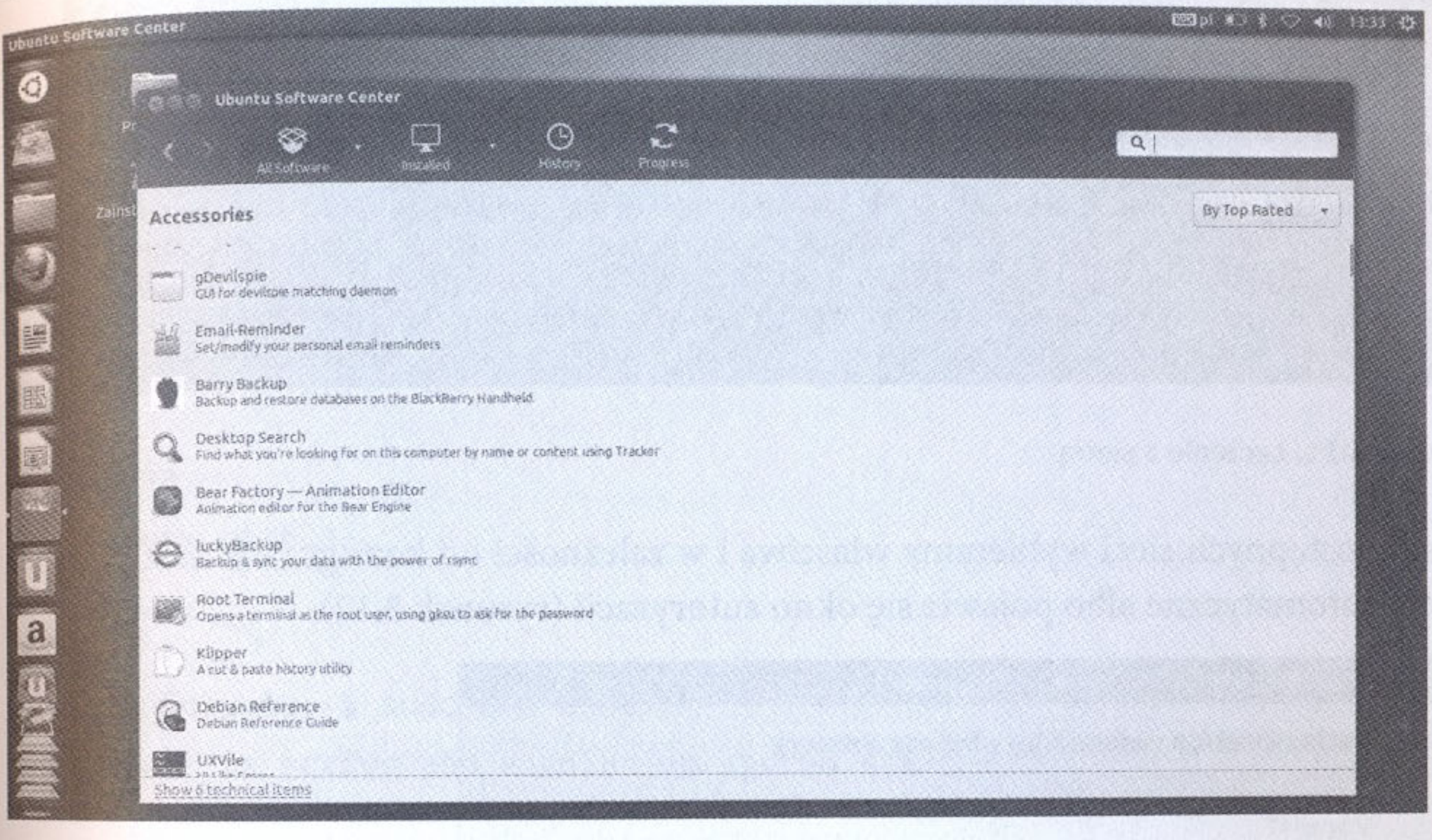


Rysunek 3.9. Workspace Switcher

Zadanie 3.6 Uruchom *Workspace Switcher*. Uruchom w okienkach:

- Terminal.
- System Settings.
- Dokument tekstowy.
- Ubuntu Software Center.

Jeżeli chcemy w terminalu wykonywać polecenia, które wymagają uprawnień *roota* (administratora), możemy uruchomić terminal z tymi uprawnieniami (rysunek 3.10).



Rysunek 3.10. Uruchamianie terminala z uprawnieniami roota

Po uruchomieniu terminala system zgłasza się nazwą użytkownika i komputera:

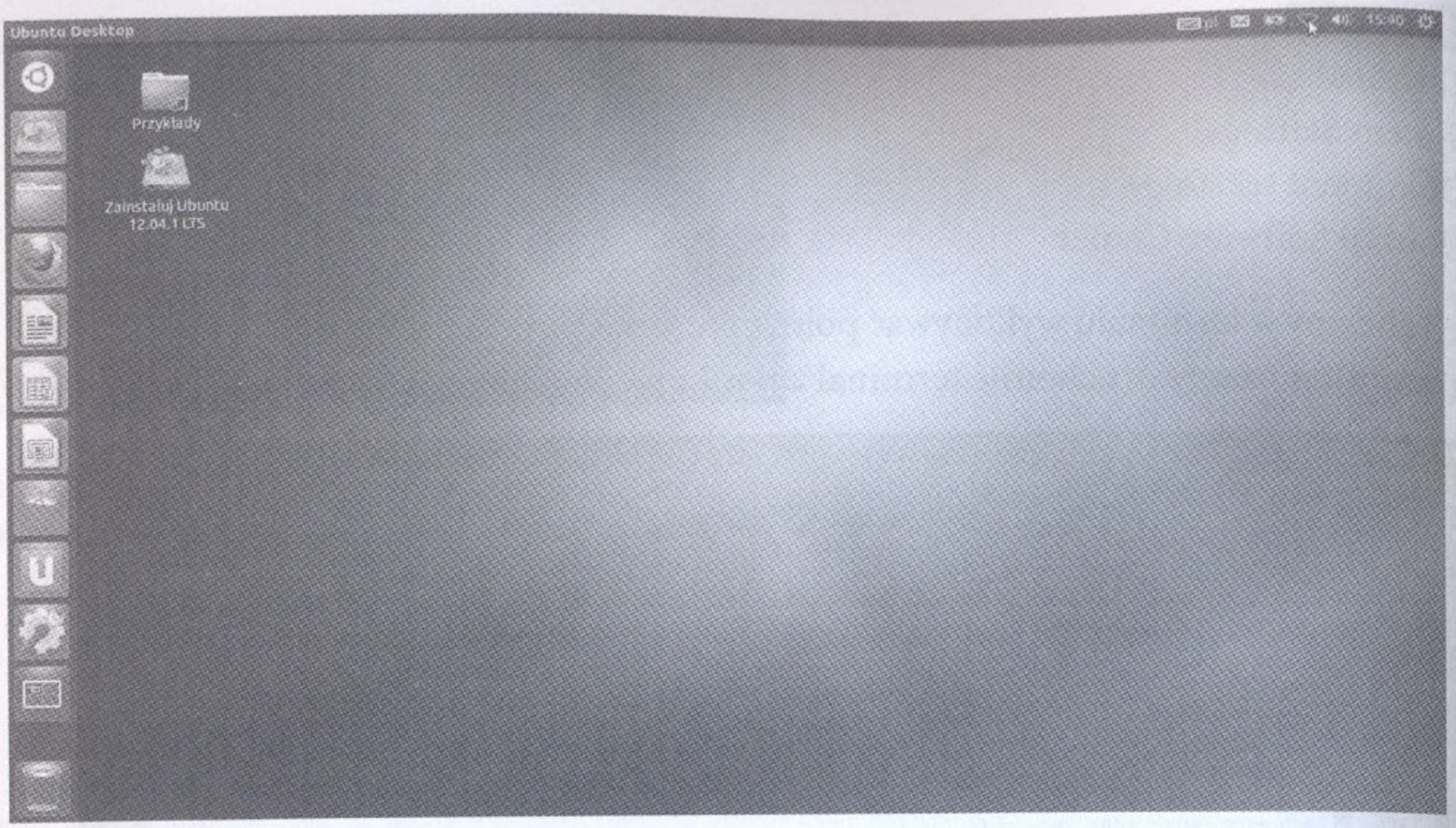
```
ubuntu@ubuntu:~$
```

Możemy teraz wpisywać polecenia. Zanim jednak do tego przejdziemy, powinniśmy zadbać o połączenie z siecią Internet. Praca w sieci umożliwia instalację i aktualizację oprogramowania z repozytoriów (bazy udostępnianych aplikacji).

Jeżeli korzystamy z sieci przewodowej (np. w szkolnej pracowni), połączenie następuje automatycznie (o ile nie ma szczególnych zabezpieczeń). Instrukcje dotyczące łączenia się z siecią bezprzewodową i konfiguracji firewalla znajdują się w kolejnym rozdziale.

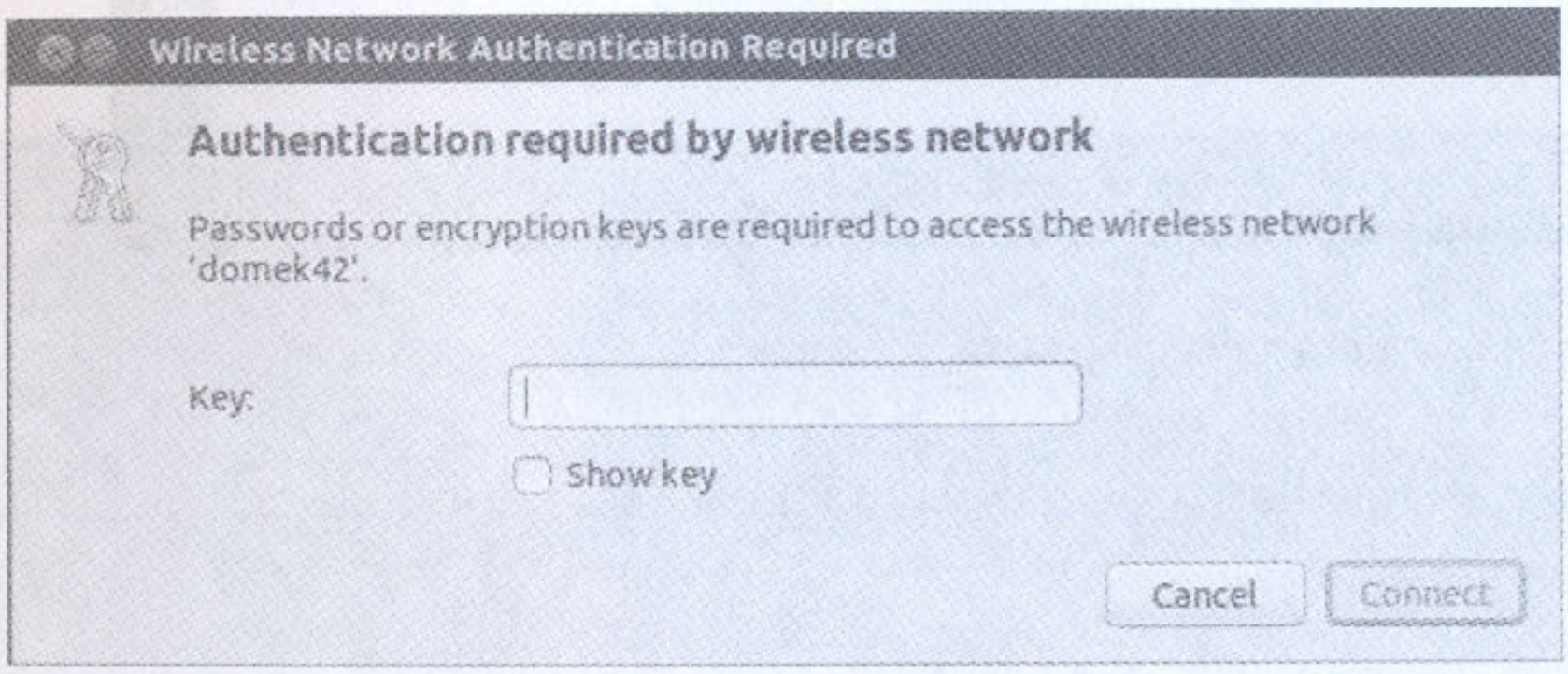
3.5. Połączenie z siecią bezprzewodową

W celu skonfigurowania połączenia sieciowego należy kliknąć ikonę, która znajduje się na górnym pasku z prawej strony (rysunek 3.11).



Rysunek 3.11. Łączenie z siecią

Z listy dostępnych sieci wybieramy właściwą i w zależności od konfiguracji następuje łączenie automatyczne albo pojawia się okno autoryzacji (rysunek 3.12).



Rysunek 3.12. Autoryzacja połączenia sieciowego

3.5.1. Zapora sieciowa (firewall)

Do konfiguracji firewalla użyjemy `ufw` (*uncomplicated firewall*), wbudowanego, prostego w obsłudze programu. Ustawieniami zapory zarządzamy z poziomu linii poleceń.

Domyślnie zapora jest wyłączona. Po włączeniu domyślne ustawienia blokują wszystkie połączenia przychodzące i akceptują wychodzące. Ustawione reguły są aktywne przy każdym uruchamianiu systemu (nie dotyczy LiveCD).

Wskazówka

Osobom, które nie korzystały dotychczas z poleceń w trybie tekstowym, proponuję przystąpienie do konfiguracji firewalla po zapoznaniu się z rozdziałem „Podstawowe polecenia w trybie tekstowym".

Ustawianie i sprawdzanie reguł

Sprawdzanie stanu

```
sudo ufw status
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw status
Status: inactive
```

Aktywacja

```
sudo ufw enable
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Szczegółowe informacje o ustawieniach

```
sudo ufw status verbose
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip
```

Blokowanie połączeń przychodzących

```
sudo ufw default deny incoming
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

Akceptowanie połączeń przychodzących

```
sudo ufw default allow incoming
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw default allow  incoming
Default incoming policy changed to 'allow'
(be sure to update your rules accordingly)
```

Blokowanie połączeń wychodzących

```
sudo ufw default deny outgoing
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
```

Akceptowanie połączeń wychodzących

```
sudo ufw default allow outgoing
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Blokowanie portów

Wskazówka

Szczegółowe informacje na temat portów komunikacyjnych znajdują się w rozdziale „Sieciowi eksperci".

```
sudo ufw deny numer_portu
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw deny 25
Rule added
Rule added (v6)
```

Wyświetlenie listy użytych reguł

```
sudo ufw status numbered
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw status numbered
Status: active

To Action From
-- ------ ----
[ 1] 25 DENY IN Anywhere
[ 2] 25 DENY IN Anywhere (v6)
```

Blokowanie połączeń z danego numeru IP

```
sudo ufw deny from numer_ip
```

Przykład

```
ubuntu@ubuntu:~$ sudo ufw deny from 85.128.139.156
Rule added
```

Zadanie 3.7 Aktywuj firewalla w swoim systemie. Zablokuj możliwość przeglądania stron internetowych oraz wysyłania poczty elektronicznej. Sprawdź działanie. Przywróć poprzedni stan.

3.6. Podstawowe polecenia w trybie tekstowym

Wskazówka

Wpisując polecenia, pamiętajmy, że Linux rozróżnia małe i wielkie litery.

Pomoc systemową zapewnia podręcznik, tzw. *man* (*manual*). Jeżeli chcemy sprawdzić składnię polecenia, piszemy:

```
man nazwa_polecenia
```

Innym sposobem uzyskania informacji na temat polecenia jest:

```
nazwa_polecenia --help
```

Przykład

```
man pwd
```

Standardowym urządzeniem wyjściowym jest konsola, ale możemy korzystać z funkcji przekierowania, czyli zmiany urządzenia wyjściowego. Jeżeli np. chcemy, żeby wynik operacji został zapisany w pliku, stosujemy składnię:

```
polecenie>>nazwa_pliku
```

Przykład

Zapisanie fragmentu podręcznika dotyczącego polecenia `ls` pliku *opis.txt*:

```
ubuntu@ubuntu:~$ man ls>>opis.txt
```

3.6.1. Składnia poleceń

polecenie -opcje argumenty

Miejsce, w którym jesteśmy, to katalog domowy użytkownika. Przekonamy się o tym, wpisując polecenie `pwd`.

Nazwę użytkownika możemy sprawdzić poleceniem `whoami`.

Do czyszczenia okna terminala wykorzystujemy polecenie `clear`.

Zadanie 3.8 Sprawdź nazwę użytkownika i aktualną lokalizację w strukturze katalogów.

3.6.2. Zarządzanie katalogami

`ls` — wyświetlanie zawartości aktualnego katalogu.

`find` — wyszukiwanie pliku.

`tree` — wyświetlanie drzewa katalogów.

`chown` — zmiana właściciela.

`mkdir` *nazwa_katalogu* — tworzenie katalogu.

`cd` *nazwa_katalogu* — wejście do katalogu.

`cd /` — zmiana katalogu na główny (*root*).

`cd..` — zmiana katalogu na nadrzędny.

`rmdir` *nazwa_katalogu* — usuwanie katalogu.

Przykład

Wykonajmy ciąg poleceń:

- Sprawdzamy swoje położenie:

```
ubuntu@ubuntu:~$ pwd
/home/ubuntu
```

- Tworzymy katalog *prace*:

```
ubuntu@ubuntu:~$ mkdir prace
```

- Wchodzimy do katalogu *prace*:

```
ubuntu@ubuntu:~$ cd prace
ubuntu@ubuntu:~/prace$
```

- Przechodzimy do katalogu głównego:

```
ubuntu@ubuntu:~$ cd /
ubuntu@ubuntu:/$
```

- Sprawdzamy swoje położenie:

```
ubuntu@ubuntu:/$ pwd
/
```

- Sprawdzamy zawartość katalogu:

```
ubuntu@ubuntu:/$ ls
bin dev initrd.img media proc run srv usr vmlinuz.old
boot etc initrd.img.old mnt rofs sbin sys var
cdrom home lib opt root selinux tmp vmlinuz
```

- Przechodzimy do katalogu *home*:

```
ubuntu@ubuntu:/$ cd home
```

- Usuwamy katalog *prace*:

```
ubuntu@ubuntu:~$ rmdir prace
```

Zadanie 3.9 Uruchom terminal.

Za pomocą polecenia `ls` (*list*) wyświetl zawartość aktualnego katalogu.

Zmień katalog na główny za pomocą polecenia `cd /`.

Przy użyciu polecenia `ls -l` (*long list*) wyświetl szczegółowe informacje o plikach i katalogach (rysunek 3.13).

```
ubuntu@ubuntu: /
lrwxrwxrwx   1 root root   32 paź 17 15:00 initrd.img -> boot/initrd.img-3.5.0-1
7-generic
lrwxrwxrwx   1 root root   33 paź 17 14:59 initrd.img.old -> /boot/initrd.img-3.
5.0-17-generic
drwxr-xr-x  23 root root 1353 paź 17 15:01 lib
drwxr-xr-x   1 root root   80 gru 28 19:26 media
drwxr-xr-x   2 root root    3 paź  9 14:59 mnt
drwxr-xr-x   2 root root    3 paź 17 14:56 opt
dr-xr-xr-x 186 root root    0 gru 28 19:09 proc
drwxr-xr-x  21 root root  405 paź 17 15:03 rofs
drwx------   1 root root  100 gru 28 19:11 root
drwxr-xr-x  21 root root  720 gru 28 19:23 run
drwxr-xr-x   2 root root 4601 paź 17 15:03 sbin
drwxr-xr-x   2 root root    3 cze 11  2012 selinux
drwxr-xr-x   2 root root    3 paź 17 14:56 srv
dr-xr-xr-x  13 root root    0 gru 28 19:10 sys
drwxrwxrwt   8 root root  240 gru 28 21:17 tmp
drwxr-xr-x   1 root root  120 paź 17 14:56 usr
drwxr-xr-x   1 root root  120 paź 17 15:03 var
lrwxrwxrwx   1 root root   29 paź 17 15:00 vmlinuz -> boot/vmlinuz-3.5.0-17-gene
ric
lrwxrwxrwx   1 root root   29 paź 17 14:59 vmlinuz.old -> boot/vmlinuz-3.5.0-17-
generic
ubuntu@ubuntu:/$
```

Rysunek 3.13. Szczegółowe informacje o plikach i katalogach w katalogu root

Pierwsza kolumna

Pierwszy znak pokazuje, czy jest to plik, katalog, czy dowiązanie (odnośnik do innego pliku).

`d` — katalog.

`-` — plik.

`l` — dowiązanie.

Kolejne znaki pokazują uprawnienia do pliku dla właściciela pliku, grupy i pozostałych użytkowników (uprawnienia zostaną omówione w dalszej części rozdziału).

Druga kolumna

Liczba dowiązań do pliku (dowiązania, czyli odnośniki, umożliwiają odwoływanie się do pliku z innego miejsca, przechowują informacje o lokalizacji; są tworzone automatycznie podczas instalacji Linuksa lub można je dodawać za pomocą instrukcji `ln`).

Pozostałe kolumny

Pozostałe kolumny pokazują odpowiednio:

- właściciela pliku,
- grupę,
- rozmiar,
- datę i czas utworzenia lub modyfikacji,
- nazwę.

Zadanie 3.10 W katalogu *home* utwórz katalog *zadania*. Wyświetl zawartość katalogu *home* wraz z nazwami właścicieli.

Zadanie 3.11 Przejdź do katalogu głównego. Wyświetl strukturę katalogów w postaci drzewa.

3.7. Prawa dostępu

Pora teraz na rozszyfrowanie danych z pierwszej kolumny, które pojawiły się przy wyświetlaniu szczegółowych informacji o plikach i katalogach (rysunek 3.13) — uprawnieniach do plików i katalogów.

Oznaczenia uprawnień

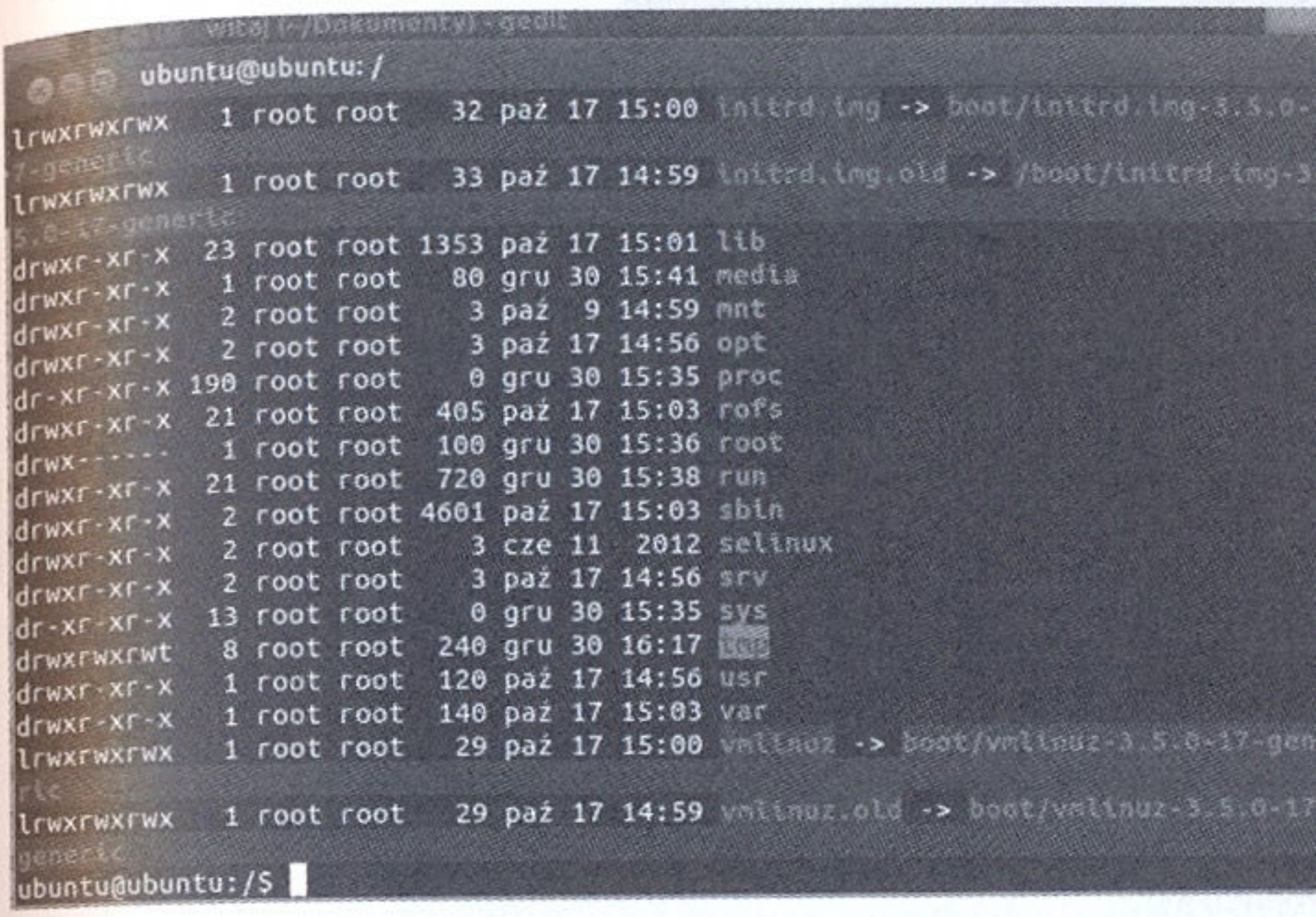
`r` — prawo do czytania.

`w` — prawo do zapisu (modyfikowania).

`x` — prawo do wykonywania.

Trzy pierwsze symbole dotyczą właściciela pliku — u (*user*), grupy — g (*group*), trzy ostatnie dotyczą pozostałych użytko reszty świata ☺.

Patrz rysunek 3.14.



Rysunek 3.14. Prawa dostępu

Porównajmy uprawnienia do katalogu *lib* i *root*.

Do katalogu *lib* wszystkie prawa ma właściciel — `rwx`; grupa i tylko prawo czytania i wykonywania — `r -x`. Uprawnienia d łącznie właściciel (*root*).

Zwróć uwagę, że brak uprawnień jest symbolizowany znakien

Do modyfikacji uprawnień służy polecenie:

```
chmod opcje nazwa
```

Wykorzystujemy do tego metodę symboliczną lub binarną.

W przypadku metody symbolicznej podajemy, kogo ma dotyc lub „-" odpowiednio nadajemy lub zabieramy uprawnienia.

Przykład

Jeżeli w bieżącym katalogu mamy plik o nazwie *plik1* i chcemy konywania, stosujemy instrukcję:

```
chmod u+x plik1
```

Patrz rysunek 3.15.

```
ubuntu@ubuntu: ~/Dokumenty
ubuntu@ubuntu:~$ cd Dokumenty
ubuntu@ubuntu:~/Dokumenty$ ls -l
total 28
-rw-rw-r-- 1 ubuntu ubuntu 481 gru 30 15:59 baza~
-rw-rw-r-- 1 ubuntu ubuntu 508 gru 30 16:15 baza2~
-rw-rw-r-- 1 ubuntu ubuntu  98 gru 30 16:47 dane.txt
-rw-rw-r-- 1 ubuntu ubuntu 313 gru 30 16:46 dane.txt~
-rw-rw-r-- 1 ubuntu ubuntu 481 gru 30 16:43 plik1
-rw-rw-r-- 1 ubuntu ubuntu 508 gru 30 16:44 plik2
-rw-rw-r-- 1 ubuntu ubuntu   0 gru 30 16:57 plik3
-rw-rw-r-- 1 ubuntu ubuntu  41 gru 30 16:52 witaj
ubuntu@ubuntu:~/Dokumenty$ chmod u+x plik1
ubuntu@ubuntu:~/Dokumenty$ ls -l
total 28
-rw-rw-r-- 1 ubuntu ubuntu 481 gru 30 15:59 baza~
-rw-rw-r-- 1 ubuntu ubuntu 508 gru 30 16:15 baza2~
-rw-rw-r-- 1 ubuntu ubuntu  98 gru 30 16:47 dane.txt
-rw-rw-r-- 1 ubuntu ubuntu 313 gru 30 16:46 dane.txt~
-rwxrw-r-- 1 ubuntu ubuntu 481 gru 30 16:43 plik1
-rw-rw-r-- 1 ubuntu ubuntu 508 gru 30 16:44 plik2
-rw-rw-r-- 1 ubuntu ubuntu   0 gru 30 16:57 plik3
-rw-rw-r-- 1 ubuntu ubuntu  41 gru 30 16:52 witaj
ubuntu@ubuntu:~/Dokumenty$
```

Rysunek 3.15. Zmiana uprawnień do pliku

Wskazówka

Zwróć uwagę na zmianę koloru nazwy *plik1*. (Nie dotyczy to wszystkich dystrybucji).

W poleceniach możemy stosować znaki specjalne, takie jak „*” i „?” (rysunek 3.16).

```
ubuntu@ubuntu: ~/Dokumenty
-rw-rw-r-- 1 ubuntu ubuntu  41 gru 30 16:52 witaj
ubuntu@ubuntu:~/Dokumenty$ chmod u+x plik1
ubuntu@ubuntu:~/Dokumenty$ ls -l
total 28
-rw-rw-r-- 1 ubuntu ubuntu 481 gru 30 15:59 baza~
-rw-rw-r-- 1 ubuntu ubuntu 508 gru 30 16:15 baza2~
-rw-rw-r-- 1 ubuntu ubuntu  98 gru 30 16:47 dane.txt
-rw-rw-r-- 1 ubuntu ubuntu 313 gru 30 16:46 dane.txt~
-rwxrw-r-- 1 ubuntu ubuntu 481 gru 30 16:43 plik1
-rw-rw-r-- 1 ubuntu ubuntu 508 gru 30 16:44 plik2
-rw-rw-r-- 1 ubuntu ubuntu   0 gru 30 16:57 plik3
-rw-rw-r-- 1 ubuntu ubuntu  41 gru 30 16:52 witaj
ubuntu@ubuntu:~/Dokumenty$ chmod g+x p*
ubuntu@ubuntu:~/Dokumenty$ ls -l
total 28
-rw-rw-r-- 1 ubuntu ubuntu 481 gru 30 15:59 baza~
-rw-rw-r-- 1 ubuntu ubuntu 508 gru 30 16:15 baza2~
-rw-rw-r-- 1 ubuntu ubuntu  98 gru 30 16:47 dane.txt
-rw-rw-r-- 1 ubuntu ubuntu 313 gru 30 16:46 dane.txt~
-rwxrwxr-- 1 ubuntu ubuntu 481 gru 30 16:43 plik1
-rw-rwxr-- 1 ubuntu ubuntu 508 gru 30 16:44 plik2
-rw-rwxr-- 1 ubuntu ubuntu   0 gru 30 16:57 plik3
-rw-rw-r-- 1 ubuntu ubuntu  41 gru 30 16:52 witaj
ubuntu@ubuntu:~/Dokumenty$
```

Rysunek 3.16. Przydzielenie grupie prawa wykonywania do wszystkich plików na literę „p”

Metoda binarna zmienia wszystkie prawa jednocześnie. Wykorzystujemy do tego maskę binarną. Jedynka symbolizuje przydzielenie prawa, zero — odebranie go.

Zgodnie z tą zasadą 101 przydziela prawa czytania i wykonywania, zabiera prawo pisania. Uprawnienia użytkownika, grupy i pozostałych są reprezentowane przez trzy pozycje.

Zatem 111 101 100 stanowi odpowiednio: `rwx r-x r--`.

111 w systemie binarnym to 7 w systemie dziesiętnym, 101 w systemie binarnym to 5 w systemie dziesiętnym, a 100 w systemie binarnym to 4 w systemie dziesiętnym.

Jeżeli chcemy przydzielić takie właśnie prawa, stosujemy zapis:

```
chmod 754 nazwa
```

Zadanie 3.12 W katalogu *Dokumenty* utwórz dwa dowolne pliki tekstowe. Przydziel wszystkim pełne prawa do tych plików.

Wskazówka

Najprostszy sposób tworzenia plików polega na kliknięciu prawym przyciskiem myszy w oknie wyświetlającym zawartość katalogu i wyborze z menu *Create New Document*. Uruchamia się wtedy *domyślny edytor tekstowy systemu* — `gedit`.

Zadanie 3.13 Co znaczą podane wartości uprawnień dla właściciela, grupy i pozostałych? Uzupełnij tabelkę.

Uprawnienia	Właściciel	Grupa	Pozostali
777			
660			
755			
744			
775			

Zadanie 3.14 Sprawdź, czy możesz zmodyfikować prawa dostępu do katalogu */bin*.

3.8. Zarządzanie użytkownikami i grupami

Głównym przeznaczeniem Linuksa jest praca w sieci, dlatego bardzo ważny jest podział na użytkowników i grupy.

Najważniejszym użytkownikiem tworzonym podczas instalacji systemu jest *root*. Ma nieograniczone uprawnienia w systemie, stąd powiedzenie „Root or God — what's a difference?".

Tworzenie użytkowników i grup jest zalecane przede wszystkim ze względu na bezpieczeństwo systemu. Z tego samego powodu nie należy pracować na koncie *roota*, jeżeli nie jest to absolutnie konieczne.

Każdy użytkownik i każda grupa ma w systemie swoją nazwę i identyfikator.

Identyfikatory użytkowników sprawdzamy poleceniem `id`. Wywołanie polecenia z opcją `-u` wyświetli identyfikator aktualnego użytkownika.

Ze względu na bezpieczeństwo systemu nie zalecamy tej opcji.

Przykład

```
sudo chmod 775 cgi-bin
```

Tworzenie użytkowników

Użytkowników w Ubuntu możemy tworzyć na dwa sposoby:

```
adduser nazwa_użytkownika
useradd nazwa_użytkownika
```

Wskazówka

Wygodniejsze w użyciu jest polecenie `adduser` (w rzeczywistości jest to skrypt dla polecenia `useradd`).

Przykład

Utwórzmy użytkownika *wiesia* i dodajmy go do tworzonej grupy *wiesia*.

Wskazówka

Podczas tworzenia użytkownika system automatycznie prosi o hasło, które koniecznie trzeba podać. Podawanie pozostałych informacji nie jest obligatoryjne. System dla każdego użytkownika tworzy katalogi domowe — *home*.

```
ubuntu@ubuntu:~$ sudo adduser wiesia
Adding user `wiesia' ...
Adding new group `wiesia' (1000) ...
Adding new user `wiesia' (1000) with group `wiesia' ...
Creating home directory `/home/wiesia' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for wiesia
Enter the new value, or press ENTER for the default
Full Name []:
```

```
Is the information correct? [Y/n] y
```

Zadanie 3.15 Utwórz użytkownika, wykorzystując polecenia adduser i useradd. Porównaj efekty działania.

Usuwanie użytkowników

```
userdel nazwa_użytkownika
```

Przykład

```
ubuntu@ubuntu:~$ sudo userdel wiesia
```

Modyfikowanie informacji o użytkowniku

```
usermod nazwa_użytkownika
```

Zmiana hasła

```
passwd [nazwa_użytkownika]
```

Polecenie bez żadnych parametrów zmienia hasło aktualnego użytkownika.

Przykład

```
ubuntu@ubuntu:~$ sudo passwd wiesia
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Logowanie

```
login nazwa_użytkownika
```

Przykład

```
ubuntu@ubuntu:~$ sudo login yatra
Password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)
```

W przeciwieństwie do systemu Windows hasło w Linuksie nie jest reprezentowane żadnymi symbolami na ekranie. Gwarantuje to automatycznie wyższy poziom bezpieczeństwa.

Tworzenie grupy

```
groupadd nazwa_grupy
```

Usuwanie grupy

```
groupdel nazwa_grupy
```

Dodawanie użytkownika do grupy

```
adduser nazwa_użytkownika nazwa_grupy
```

Przykład

Utwórzmy użytkownika *yatra* oraz grupę *moja_klasa* i dodajmy użytkownika *yatra* do tej grupy.

Patrz rysunek 3.17.

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sudo adduser yatra
Adding user `yatra' ...
Adding new group `yatra' (1001) ...
Adding new user `yatra' (1001) with group `yatra' ...
Creating home directory `/home/yatra' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for yatra
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
ubuntu@ubuntu:~$ sudo groupadd moja_klasa
ubuntu@ubuntu:~$ sudo adduser yatra moja_klasa
Adding user `yatra' to group `moja_klasa' ...
Adding user yatra to group moja_klasa
Done.
ubuntu@ubuntu:~$
```

Rysunek 3.17. Tworzenie użytkownika i przypisanie go do grupy

Zadanie 3.16 Utwórz grupę *kolko_informatyczne*. Dodaj do niej pięciu wybranych uczestników zajęć.

3.9. Procesy

W systemie Linux proces jest to ciąg poleceń wykonywanych aktualnie przez procesor; może to być program lub jakaś jego funkcja. Pojedynczy program może się wykonywać w wielu procesach.

Każdy proces w systemie ma nadany unikalny numer PID (*Process IDentifer*). Podczas startu systemu uruchamiany jest proces *init*, który ma identyfikator 1. Wszystkie inne procesy są procesami potomnymi. Każdy proces ma przypisanego użytkownika, który go uruchomił.

Usługi uruchamiane zazwyczaj przy starcie systemu i działające przez cały czas w tle (zużywają niewielkie zasoby) są to tzw. *demony*.

3.9.1. Stany procesów

1. Działający.
2. Uśpiony (czeka na zdarzenie).
3. Gotowy (czeka na przydzielenie czasu procesora).
4. Zombie (zakończył działanie, czeka na zakończenie go przez proces macierzysty).

3.9.2. Polecenia służące do zarządzania procesami

`ps` — wyświetlanie procesów.

`kill` — zabijanie procesów.

`nice` — zmiana priorytetów procesów.

`top` — lista procesów najbardziej obciążających system.

Opcje polecenia `ps`:

- `ps` — wyświetla procesy aktualnego użytkownika;
- `ps -A` — wyświetla wszystkie procesy;
- `ps -l` — długi format informacji (dodatkowo czas utworzenia i prawa dostępu);
- `ps -m` — wyświetla informacje o używanej pamięci;
- `ps -r` — lista pracujących procesów.

Patrz rysunek 3.18.

```
ubuntu@ubuntu:~$ ps
  PID TTY          TIME CMD
 7578 pts/5    00:00:00 bash
 7634 pts/5    00:00:00 ps
ubuntu@ubuntu:~$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   999  7578  6605  0  80   0 -  2082 wait   pts/5    00:00:00 bash
0 R   999  7744  7578  0  80   0 -  1477 -      pts/5    00:00:00 ps
ubuntu@ubuntu:~$ ps -Al
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0     1     0  0  80   0 -   880 poll_s ?        00:00:00 init
1 S     0     2     0  0  80   0 -     0 kthrea ?        00:00:00 kthreadd
1 S     0     3     2  0  80   0 -     0 run_ks ?        00:00:00 ksoftirqd/0
1 S     0     6     2  0 -40   - -     0 cpu_st ?        00:00:00 migration/0
5 S     0     7     2  0 -40   - -     0 watchd ?        00:00:00 watchdog/0
1 S     0     8     2  0 -40   - -     0 cpu_st ?        00:00:00 migration/1
1 S     0    10     2  0  80   0 -     0 run_ks ?        00:00:00 ksoftirqd/1
5 S     0    12     2  0 -40   - -     0 watchd ?        00:00:00 watchdog/1
1 S     0    13     2  0  60 -20 -     0 rescue ?        00:00:00 cpuset
1 S     0    14     2  0  60 -20 -     0 rescue ?        00:00:00 khelper
5 S     0    15     2  0  80   0 -     0 devtmp ?        00:00:00 kdevtmpfs
1 S     0    16     2  0  60 -20 -     0 rescue ?        00:00:00 netns
1 S     0    18     2  0  80   0 -     0 bdi_sy ?        00:00:00 sync_supers
1 S     0    19     2  0  80   0 -     0 bdi_fo ?        00:00:00 bdi-default
1 S     0    20     2  0  60 -20 -     0 rescue ?        00:00:00 kintegrityd
1 S     0    21     2  0  60 -20 -     0 rescue ?        00:00:00 kblockd
1 S     0    22     2  0  60 -20 -     0 rescue ?        00:00:00 ata_sff
1 S     0    23     2  0  80   0 -     0 hub_th ?        00:00:00 khubd
1 S     0    24     2  0  60 -20 -     0 rescue ?        00:00:00 md
1 S     0    25     2  0  80   0 -     0 worker ?        00:00:00 kworker/0:2
```

Rysunek 3.18. Lista uruchomionych procesów

Rozszyfrujmy istotne parametry procesów, które widać na rysunku.

1. PID — identyfikator procesu.

2. PPID — identyfikator procesu macierzystego.

3. UID — identyfikator użytkownika, który uruchomił proces.

Listę zadań najbardziej obciążających procesor przedstawiono na rysunku 3.19.

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ top

top - 15:44:14 up 7 min,  8 users,  load average: 0.68, 1.47, 0.91
Tasks: 203 total,   1 running, 202 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.5%us,  0.8%sy,  0.0%ni, 98.7%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3092672k total,  1099184k used,  1993488k free,   143140k buffers
Swap:  4390908k total,        0k used,  4390908k free,   594148k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5070 root      20   0 86324  28m 7700 S    1  0.9   0:09.53 Xorg
 5181 ubuntu    20   0  219m  65m  31m S    1  2.2   0:18.75 compiz
   11 root      20   0     0    0    0 S    1  0.0   0:02.20 kworker/0:1
 5712 ubuntu    20   0 91088  15m  10m S    1  0.5   0:00.99 gnome-terminal
 5279 ubuntu    20   0 42244 9.8m 8028 S    0  0.3   0:00.31 gtk-window-deco
 5774 ubuntu    20   0  2836 1192  880 R    0  0.0   0:00.25 top
    1 root      20   0  3520 2036 1352 S    0  0.1   0:00.67 init
    2 root      20   0     0    0    0 S    0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S    0  0.0   0:00.11 ksoftirqd/0
    6 root      RT   0     0    0    0 S    0  0.0   0:00.00 migration/0
    7 root      RT   0     0    0    0 S    0  0.0   0:00.17 watchdog/0
    8 root      RT   0     0    0    0 S    0  0.0   0:00.00 migration/1
    9 root      20   0     0    0    0 S    0  0.0   0:01.88 kworker/1:0
   10 root      20   0     0    0    0 S    0  0.0   0:00.08 ksoftirqd/1
   12 root      RT   0     0    0    0 S    0  0.0   0:00.01 watchdog/1
```

Rysunek 3.19. Lista procesów najbardziej obciążających system

Zadanie 3.17 Kto jest właścicielem procesów wyświetlonych na rysunku 3.19?

Zadanie 3.18 Uruchom polecenie wyświetlające wszystkie procesy. Które z nich uruchomił użytkownik *ubuntu*?

Zadanie 3.19 Sprawdź, które procesy najbardziej obciążają system.

Zadanie 3.20 Uruchom przeglądarkę internetową. Sprawdź numer procesu i zabij go.

Wskazówka

Sprawdź w *manualu*, jakie parametry podajemy w poleceniu `kill`.

3.10. Zarządzanie pakietami, źródłami, archiwami

System Linux standardowo oferuje użytkownikom dużo aplikacji. Jak możemy doinstalować dodatkowe programy lub zaktualizować istniejące?

Definicja

Programy na licencji GNU są rozpowszechniane z kodem źródłowym, przez co nazywa się je źródłami. Skompilowane źródła to **pakiety**.

Zarządzanie pakietami w trybie tekstowym

Instalacja programu z pakietu

```
sudo apt-get install nazwa pakietu
```

Aktualizacja listy pakietów

```
sudo apt-get update
```

Aktualizacja pakietów

```
sudo apt-get upgrade
```

Usuwanie pakietu

```
sudo apt-get remove nazwa pakietu
```

Wyszukiwanie pakietów

```
sudo apt-cache search nazwa pakietu
```

Usuwanie pobranych plików archiwów

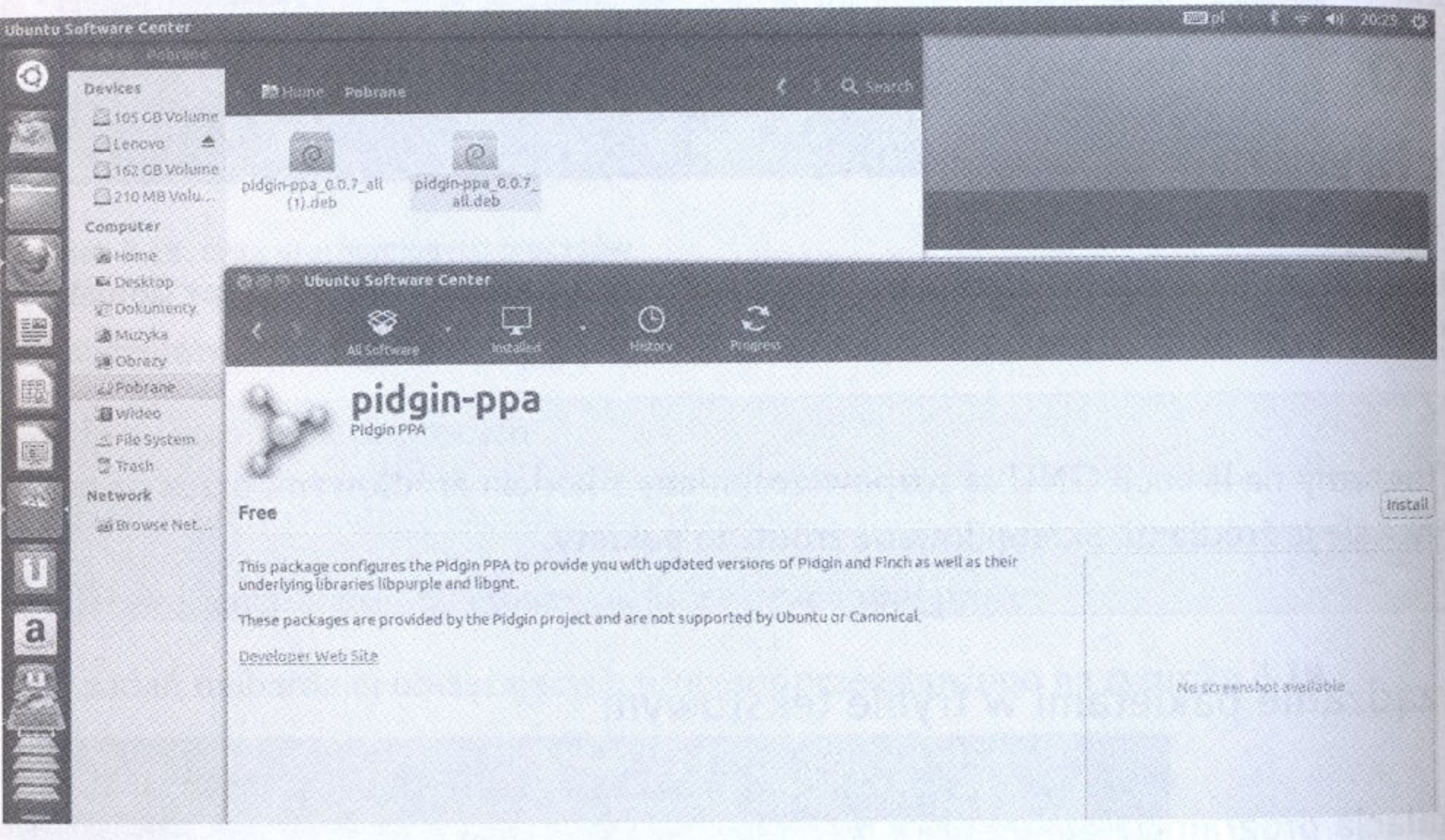
```
sudo apt-get clean
```

Zarządzanie pakietami w trybie graficznym

Instalacja

Instalacja pakietów w trybie graficznym jest bardzo prosta. Należy wykonać kilka kroków:

- ściągnąć pakiet programu w wersji odpowiedniej do naszej dystrybucji;
- wejść do katalogu, w którym pakiet się zapisał (standardowo jest to katalog *Pobrane*, który znajduje się w katalogu *Home*);
- kliknąć nazwę pakietu — automatycznie zostanie uruchomiony *Ubuntu Software Center* (rysunek 3.20);

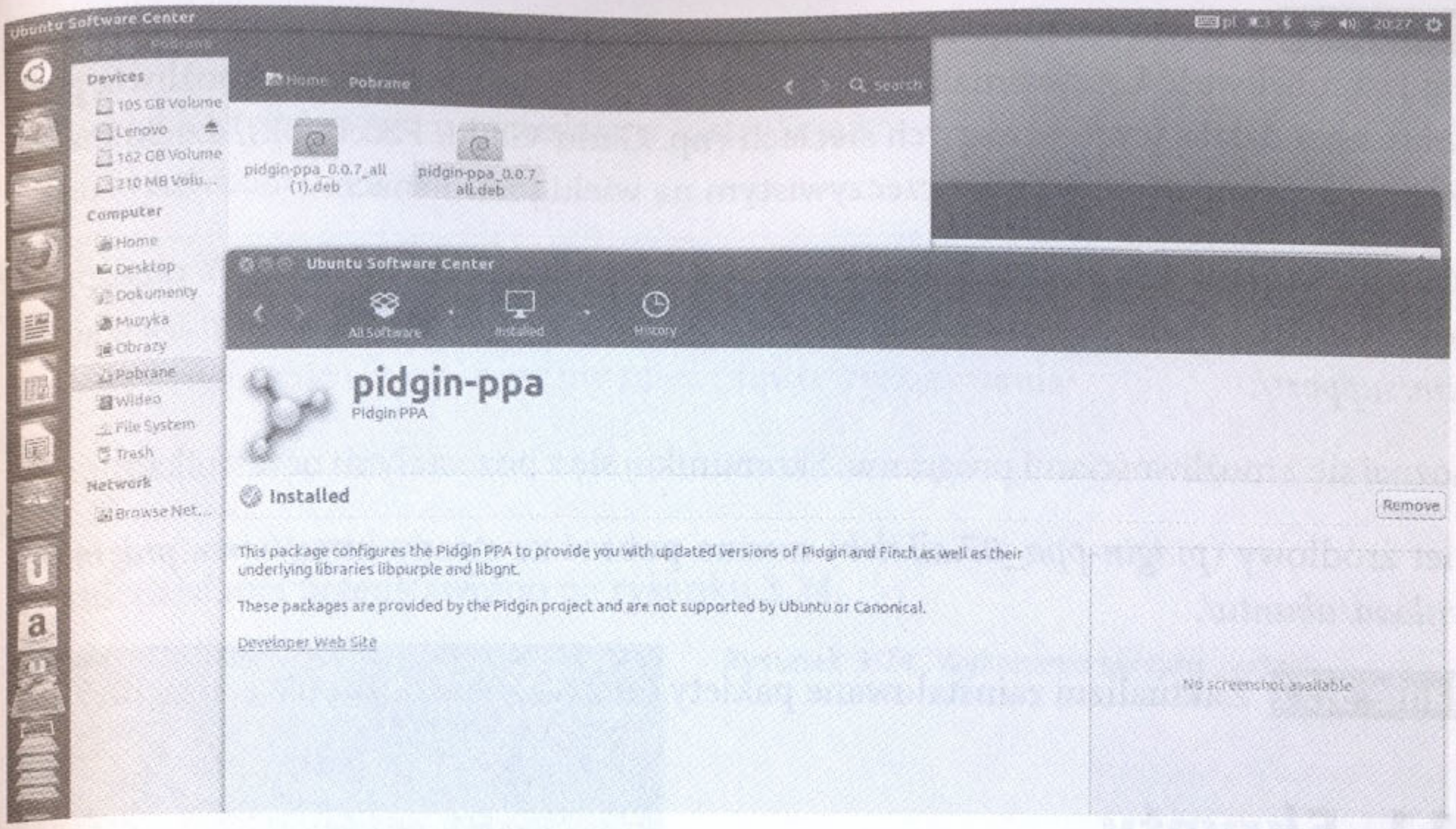


Rysunek 3.20. Instalacja programu z pakietu

- kliknąć przycisk *Install*.

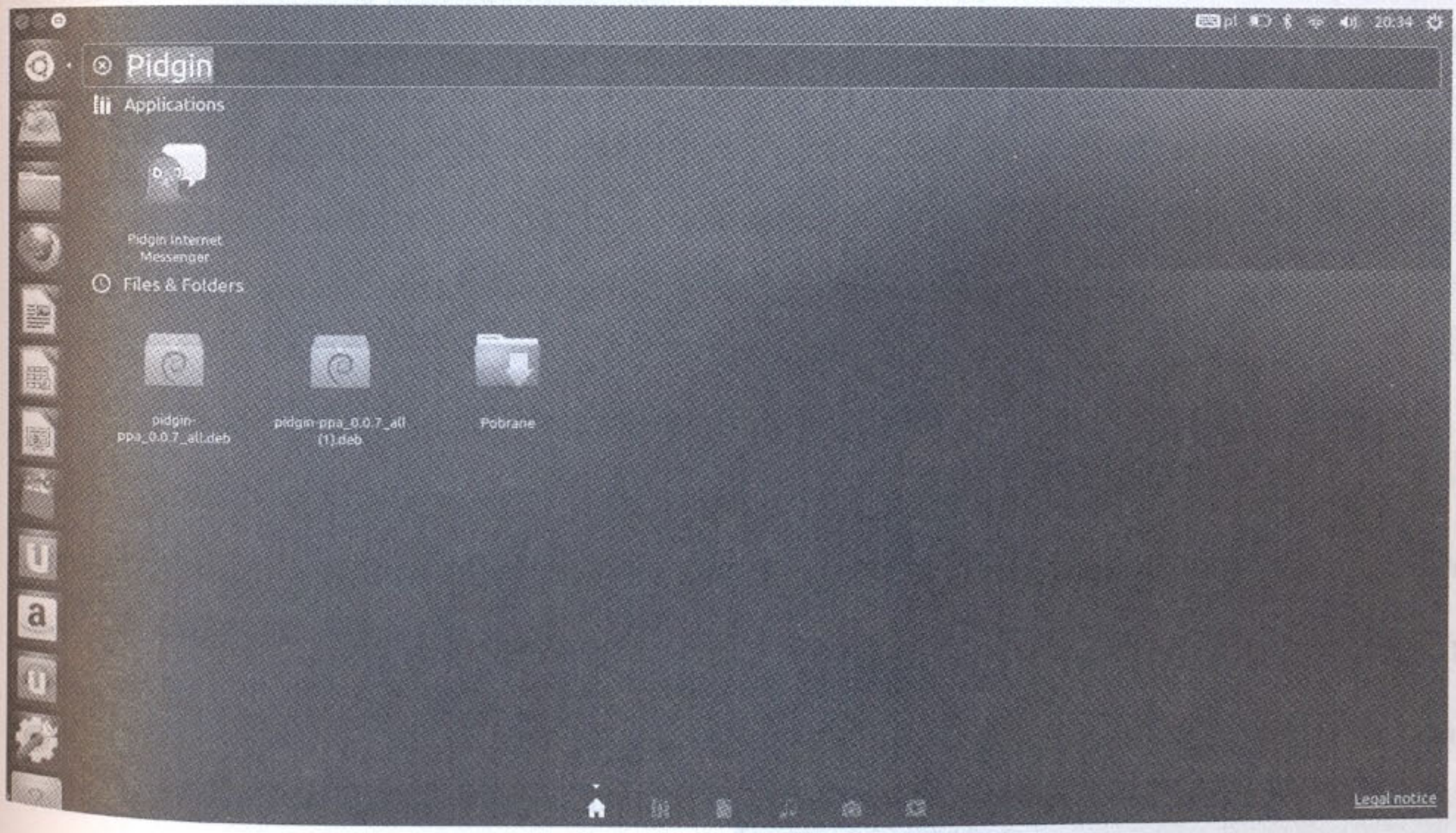
Jeśli instalacja przebiegnie pomyślnie, pojawi się okno informujące, że program jest zainstalowany.

Patrz rysunek 3.21.



Rysunek 3.21. Informacja o prawidłowym zainstalowaniu programu

Zainstalowany program znajdziemy wśród aplikacji (rysunek 3.22).



Rysunek 3.22. Wyszukiwanie zainstalowanego programu

Usuwanie

Do usuwania programów w trybie graficznym wykorzystujemy również *Ubuntu Software Center*. Znajdujemy program wśród aplikacji i klikamy przycisk *Remove*.

Zadanie 3.21 Jednym z najpopularniejszych komunikatorów jest Pidgin. Jego niewątpliwą zaletą jest możliwość korzystania z wielu komunikatorów jednocześnie. Umożliwia podłączenie kont działających w różnych sieciach (np. Gadu-Gadu, Facebook), dzięki czemu można się komunikować w czasie rzeczywistym na wielu platformach.

Zainstaluj komunikator na swoim komputerze. Zaimportuj kontakty z GG. W razie problemów skorzystaj z pomocy, która znajduje się w programie lub na stronie *http://www.pidgin.im/support/*.

Zapoznaj się z możliwościami programu. Skomunikuj się z pozostałymi uczestnikami zajęć.

Pakiet źródłowy (*pidgin-ppa_07.all.deb*) można pobrać ze strony *http://www.pidgin.im/download/ubuntu/*.

Zadanie 3.22 Zaktualizuj zainstalowane pakiety oraz odczytaj liczbę aktualizacji.

3.11. Skrypty

Jeżeli chcemy w Linuksie zautomatyzować powtarzające się czynności lub połączyć kilka poleceń w jedno, tworzymy skrypty — programy powłoki systemowej. W tym celu w dowolnym edytorze tekstowym (np. *gedit*) tworzymy plik i umieszczamy w nim potrzebne nam polecenia. Do dyspozycji mamy prawdziwy język programowania, który zawiera instrukcje warunkowe i iteracyjne. Najczęściej skrypty są wykorzystywane do automatyzacji prac administracyjnych, ale nie tylko.

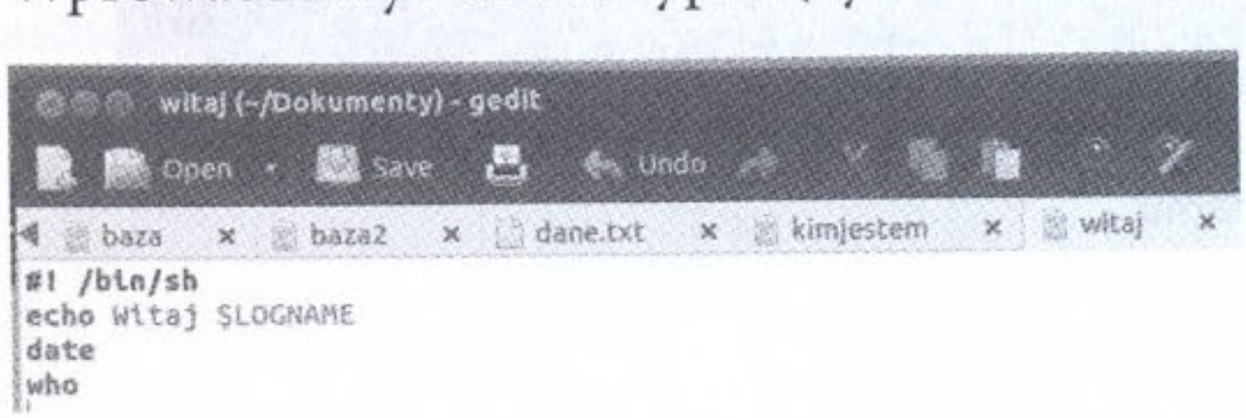
Z mechanizmem tworzenia i uruchamiania skryptów zapoznamy się na prostych przykładach.

3.11.1. Tworzenie skryptu

Utwórzmy skrypt wyświetlający tekst powitania z nazwą użytkownika oraz aktualną datę.

Uruchamiamy edytor tekstowy — *gedit*.

Wprowadzamy tekst skryptu (rysunek 3.23).



Rysunek 3.23. Tworzenie skryptu „witaj"

Zwróć uwagę, że edytor automatycznie koloruje kod.

Zapisujemy w katalogu *Dokumenty* pod nazwą *witaj*.

3.11.2. Uruchamianie skryptu

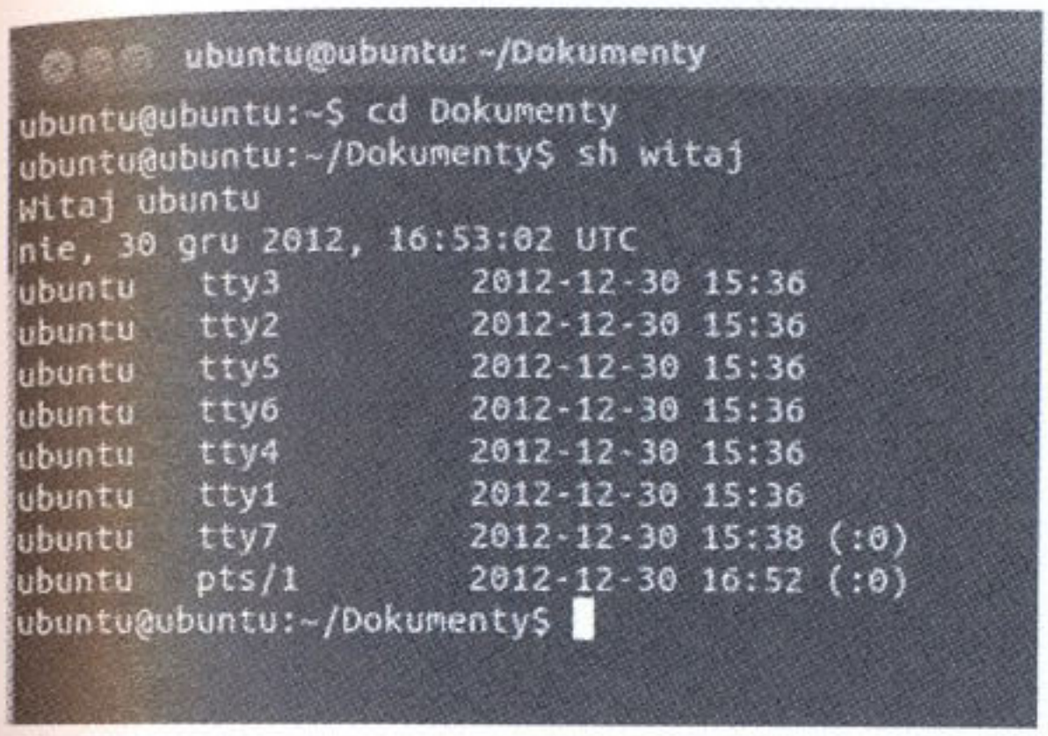
Nazwę pliku tekstowego poprzedzamy instrukcją `sh`

```
ubuntu@ubuntu:~$ sh witaj
```

Wskazówka

Żeby uruchomić skrypt, musimy mieć prawo wykonywania.

Wynik działania przedstawiono na rysunku 3.24.



Rysunek 3.24. Wykonanie skryptu „witaj"

Zadanie 3.23 Sprawdź, do czego służy polecenie `who`.

Wyniki działania skryptów mogą być wyświetlane bezpośrednio na ekranie, ale mogą być też zapisywane w plikach. Korzystamy w tym celu z operacji przekierowania:

```
echo "$nazwa_zmiennej" >> nazwa_pliku
```

Pliki można tworzyć również bezpośrednio w konsoli:

```
touch nazwa_pliku
```

Do pobierania danych z klawiatury stosujemy instrukcję:

```
read zmienna
```

Przykład

```
#!/bin/bash
echo "Podaj swoje imię :"
read name
echo "Witaj $name"
```

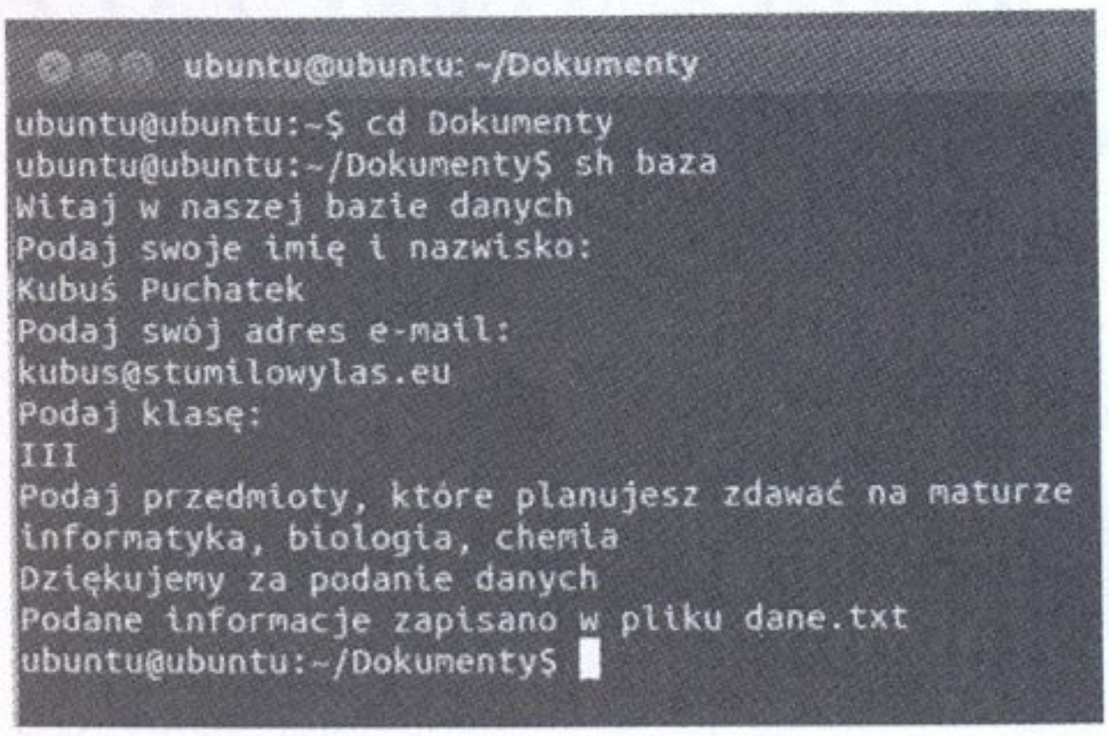
Wskazówka

Zwróć uwagę, że przy wypisywaniu wartości zmiennej jej nazwę poprzedzamy znakiem $.

Utworzymy skrypt o nazwie *baza*, zapisujący dane do pliku *dane.txt*.

```
#!/bin/bash
echo "Witaj w naszej bazie danych"
echo "Podaj swoje imię i nazwisko:"
read name
echo "Podaj swój adres e-mail:"
read mail
echo "Podaj klasę:"
read klasa
echo "Podaj przedmioty, które planujesz zdawać na maturze"
read przedmioty
echo "Dziękujemy za podanie danych"
touch dane.txt
echo "$name" >> dane.txt
echo "e-mail: $mail" >> dane.txt
echo "klasa: $klasa" >> dane.txt
echo "przedmioty: $przedmioty">> dane.txt
echo "Podane informacje zapisano w pliku dane.txt"
```

Wykonanie skryptu jest widoczne na rysunku 3.25.



Rysunek 3.25. Wykonanie skryptu „baza”

Zawartość pliku *dane.txt*

```
Kubuś Puchatek
e-mail: kubus@stumilowylas.eu
klasa: III
przedmioty: informatyka, biologia, chemia
```

3.11.3. Instrukcje warunkowe i iteracyjne w skryptach

Instrukcja warunkowa

```
if [warunek]
then
instrukcja
fi
```

lub

```
if [warunek]
then
instrukcja
else
instrukcja
fi
```

Operatory porównania

`-gt` — większy.

`-lt` — mniejszy.

`-ge` — większy lub równy.

`-le` — mniejszy lub równy.

`-eq` — równy.

`-ne` — różny.

Przykład

```
#!/bin/bash
echo "Podaj pierwszą liczbę"
read liczba1
echo "Podaj drugą liczbę"
read liczba2
if [ $1iczba -gt $liczba2 ]
then
echo "Pierwsza liczba jest większa"
else
echo "Druga liczba jest większa"
fi
```

Instrukcje iteracyjne

Pętla for

```
for nazwa_zmiennej in lista
do
polecenie1
polecenie2
...
done
```

Wskazówka

Instrukcje są wykonywane tyle razy, ile jest elementów listy.

Pętla while

```
while [ warunek ]
do
polecenie1
polecenie2
...
done
```

Wskazówka

Pętla wykonuje się, dopóki warunek jest prawdziwy.

TP **Zadanie 3.24** Zmodyfikuj skrypt *baza* tak, by po wywołaniu wprowadzić dane pięciu osób.

3.11.4. Wywoływanie skryptu z parametrami

Skrypty można wywoływać z parametrami, które są traktowane podobnie jak zmienne z tą różnicą, że ich wartości nie ulegają modyfikacjom. Każdy parametr ma numer, do którego odwołujemy się w treści skryptu w następujący sposób:

- `$*` — wszystkie parametry,
- `$0` — nazwa skryptu (pliku),
- `$1` — parametr numer 1,
- `$#` — liczba parametrów.

Przykład

Skrypt wyświetlający swoją nazwę i komunikat, która liczba wprowadzona jako parametr jest większa:

```
#!/bin/bash
echo $0
if [ "$1" -gt "$2" ]
then
echo 'Pierwsza liczba jest większa'
else
echo 'Druga liczba jest większa'
fi
```

Wywołanie skryptu:

```
ubuntu@ubuntu:~/Dokumenty$ sh porownanie 5 7
porownanie
Druga liczba jest większa
```

Przykład

Skrypt nadający wszystkim pełne uprawnienia do pliku podanego jako parametr, jednocześnie sprawdzający, czy została podana nazwa pliku:

```
#!/bin/bash
if [ "$#" -lt 1 ]
then
echo 'Nie podałeś nazwy pliku!'
else
chmod 777 $1
fi
```

Wywołanie i sprawdzenie działania skryptu:

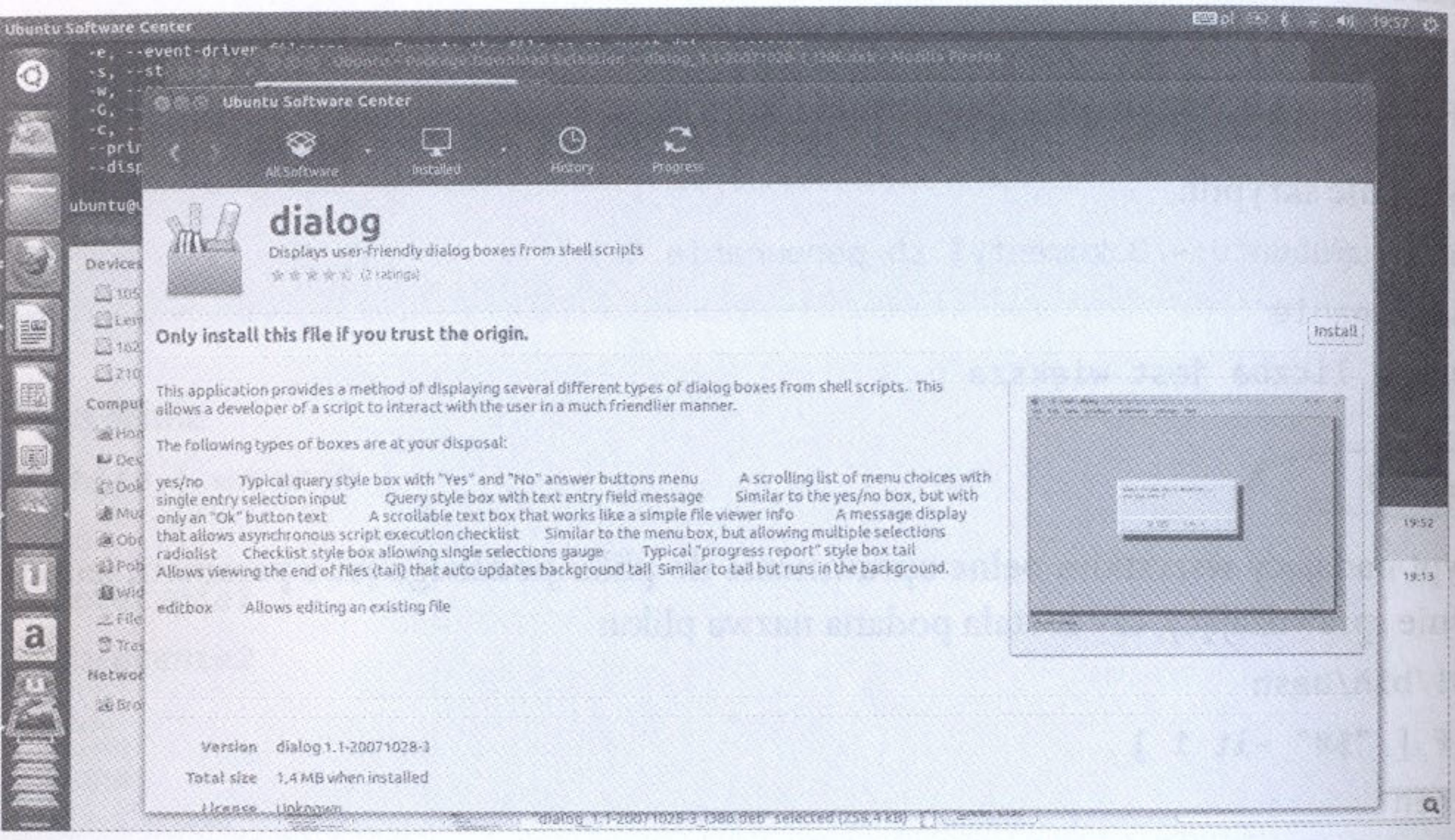
```
ubuntu@ubuntu:~/Dokumenty$ sh prawa_dla_wszystkich
Nie podałeś nazwy pliku!
ubuntu@ubuntu:~/Dokumenty$ sh prawa_dla_wszystkich plik
ubuntu@ubuntu:~/Dokumenty$ ls -l
total 8
-rwxrwxrwx 1 ubuntu ubuntu 0 lut 10 16:22 plik
```

TP **Zadanie 3.25** Do zmiany właściciela pliku i katalogu służy polecenie **chown** *użytkownik plik*. Napisz skrypt, który zmienia właściciela pliku i w którym nazwa zarówno pliku, jak i właściciela jest podawana jako parametr.

czać komunikaty dla użytkownika, informować o stanie wykonywanego procesu czy pobierać dane od użytkownika.

W tym celu zainstalujemy pakiet dialog — *dialog_1.1-20071028-3_i386.deb*, który można pobrać ze strony *http://packages.ubuntu.com/hardy/dialog*.

Instalację przeprowadzimy w trybie graficznym — patrz rysunek 3.26.



Rysunek 3.26. Instalacja pakietu dialog

Po zainstalowaniu pakietu możemy wywoływać okna dialogowe bezpośrednio w terminalu lub umieszczać kody w skryptach.

Najczęściej używane okna dialogowe to:

- `yesno` — wybór jednej z opcji;
- `inputbox` — okno do pobierania (wprowadzania) danych;
- `msgbox` — wyświetla podany komunikat, zamyka się po naciśnięciu przycisku *OK*;
- `infobox` — podobnie jak `msgbox` z tą różnicą, że zamyka się samoczynnie (brak przycisku);
- `textbox` — wykorzystywany do wyświetlania zawartości plików tekstowych;
- `menu` — wyświetla się zdefiniowana lista, z której można wybrać tylko jedną pozycję;
- `checklist` — pojawia się lista wyboru, na której można zaznaczyć dowolną liczbę pozycji;
- `radiolist` — podobnie jak `menu`, tyle że inny jest sposób zaznaczania wyboru;
- `gauge` — pokazuje wynik procentowy wykonywania operacji (np. postęp instalacji programu) w postaci miarki na dole okna.

Zapoznamy się z funkcjami niektórych z nich, wywołując polecenia bezpośrednio w terminalu. Po wykonaniu ćwiczeń wykorzystanie kodów w skryptach nie powinno sprawić żadnego kłopotu.

3.12.1. Składnia poleceń

`dialog` *`opcje okno`*

Wskazówka

Przed wywołaniem okien dialogowych dobrze jest zajrzeć do *manuala*, żeby sprawdzić, jakie opcje są dostępne i w jakiej kolejności powinniśmy je wywoływać.

Kalendarz

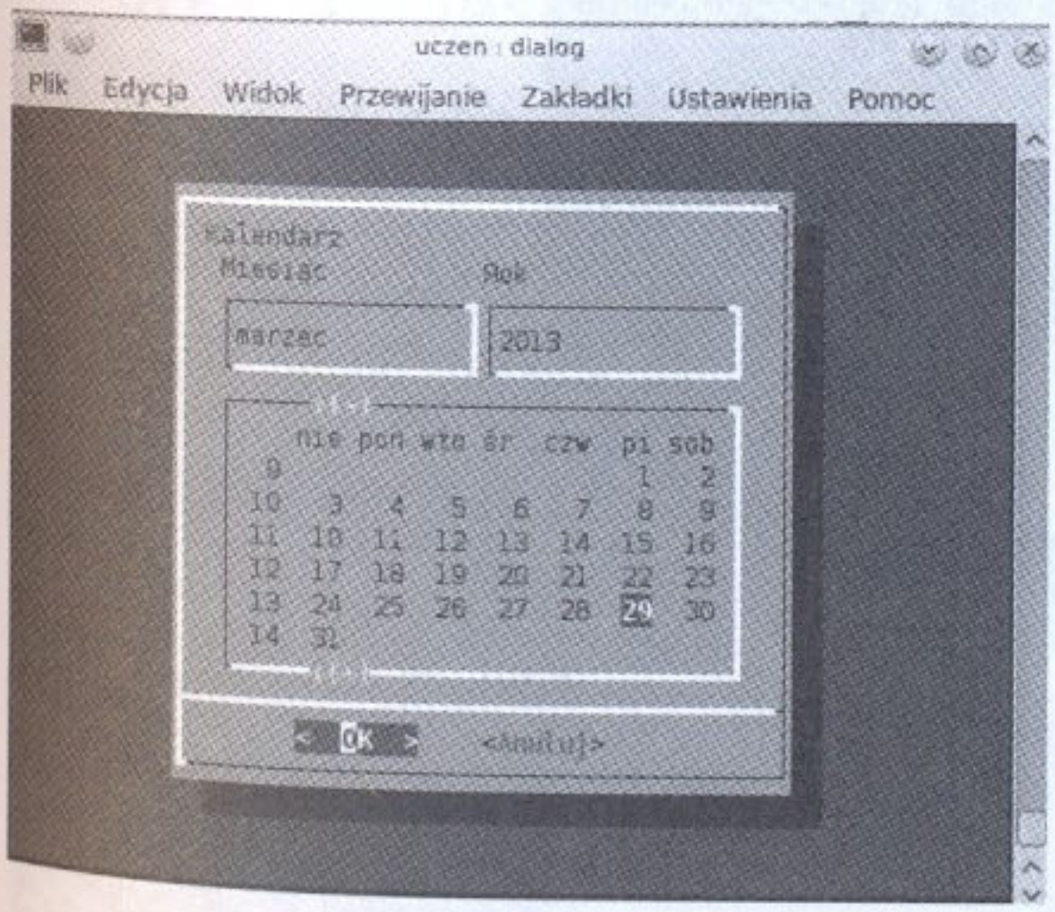
Składnia:

```
dialog --calendar  "Tekst" wysokość szerokość
```

Przykład

```
dialog --calendar "Kalendarz" 2 40
```

Patrz rysunek 3.27.



Rysunek 3.27. Okno dialogowe — kalendarz

Zadanie 3.26 Wywołaj okna dialogowe:

- zegar:

```
dialog --timebox "Zegar" 5 5
```

- msgbox:

```
dialog --msgbox "Uwaga" 5 10
```

- yesno:

```
dialog --title "koniec" --yesno "Wybierz tak lub nie" 5 30
```

Spróbuj wywołać je również z innymi parametrami.

Okno menu

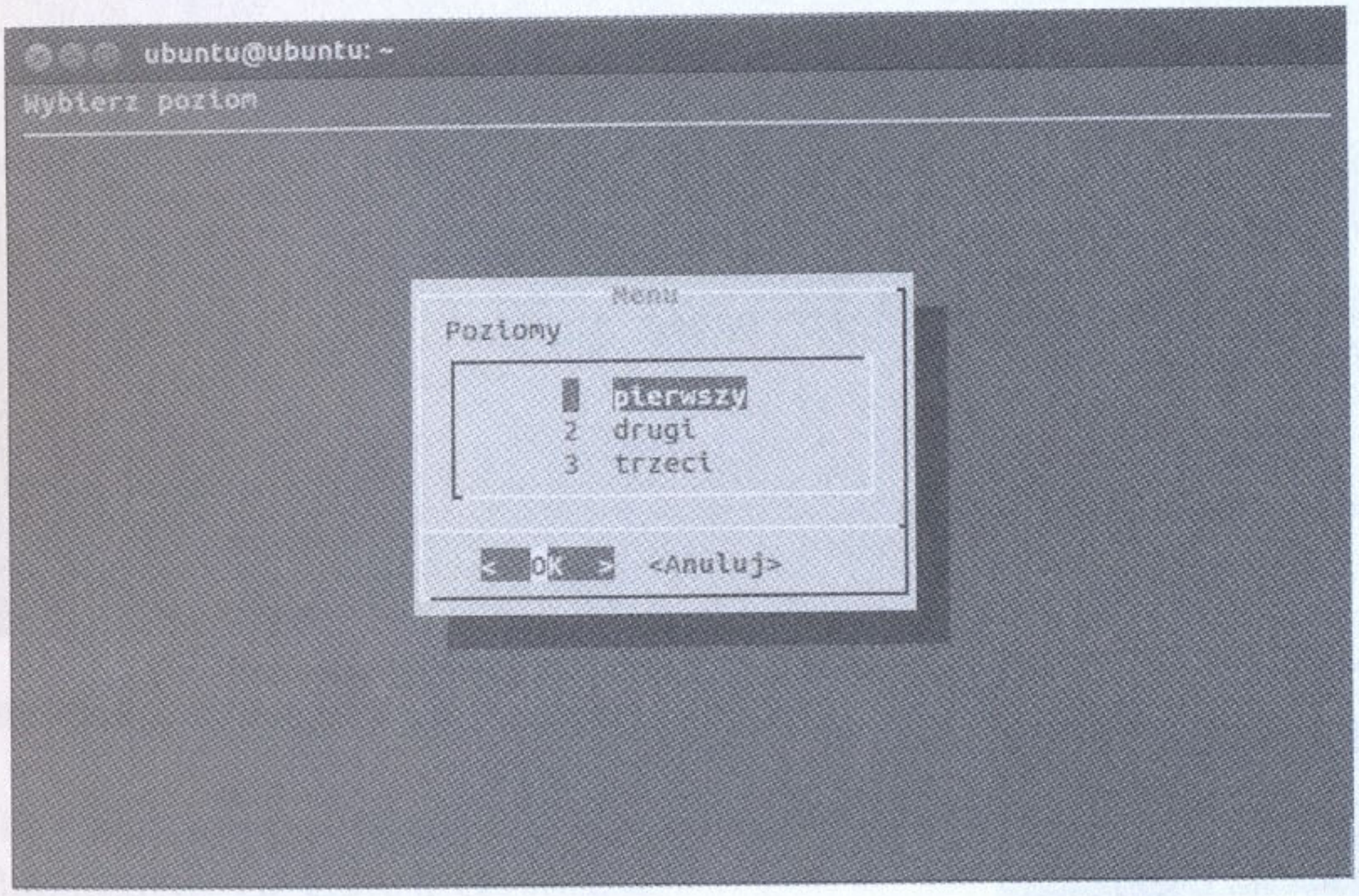
Składnia:

```
dialog --title "Tekst" --backtitle "Tekst" --menu "Tekst" wysokość
szerokość wysokość_menu lista
```

Przykład

```
dialog --title "Menu" --backtitle "Wybierz poziom" --menu "Poziomy" 10
30 3 1 pierwszy 2 drugi 3 trzeci
```

Patrz rysunek 3.28.



Rysunek 3.28. Okno dialogowe — Menu

Okno textbox

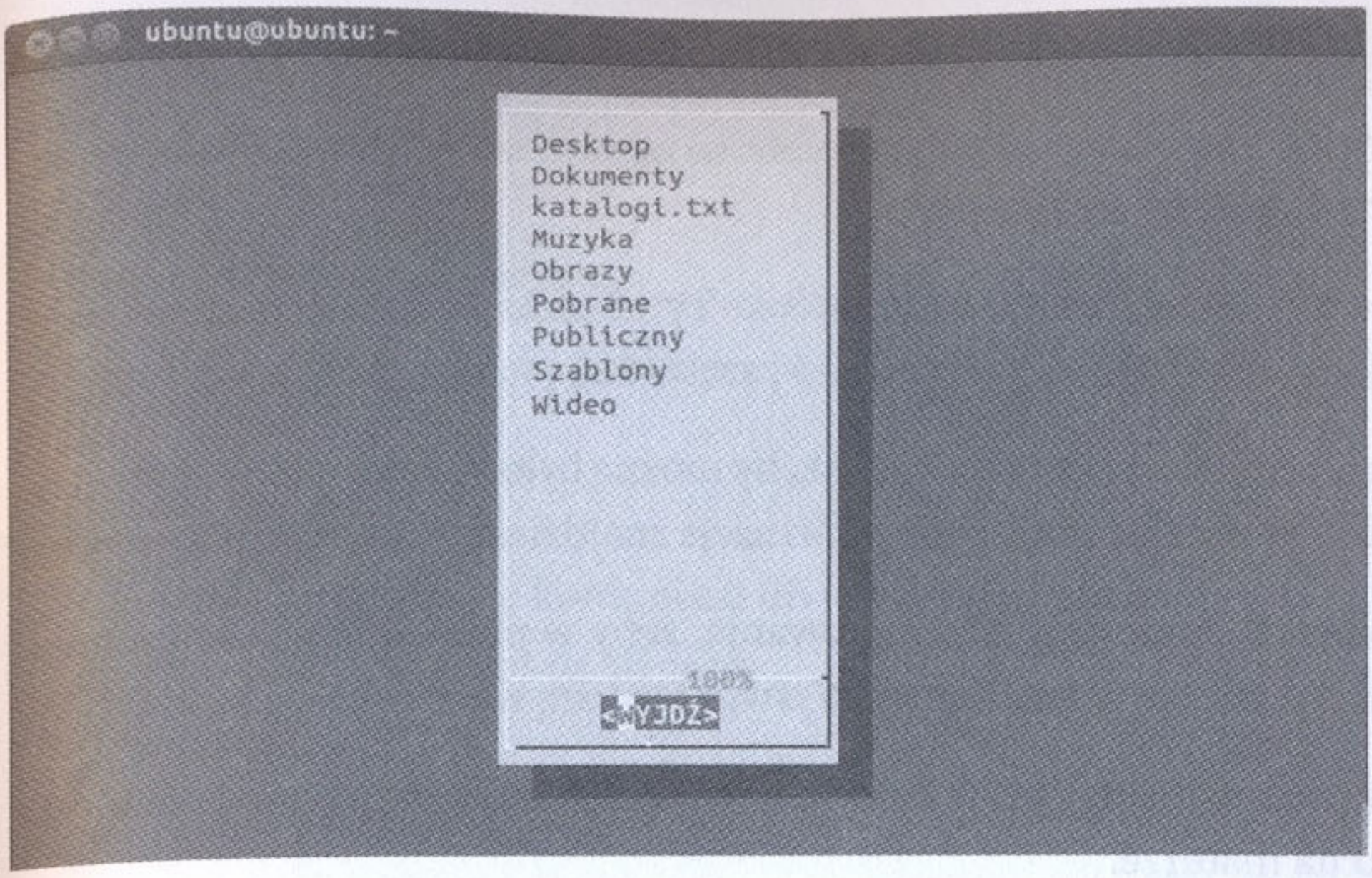
Składnia:

```
textbox nazwa_pliku wysokość szerokość
```

Przykład

```
dialog --textbox katalogi.txt 20 20
```

Patrz rysunek 3.29.



Rysunek 3.29. Okno dialogowe — textbox

Okno checklist

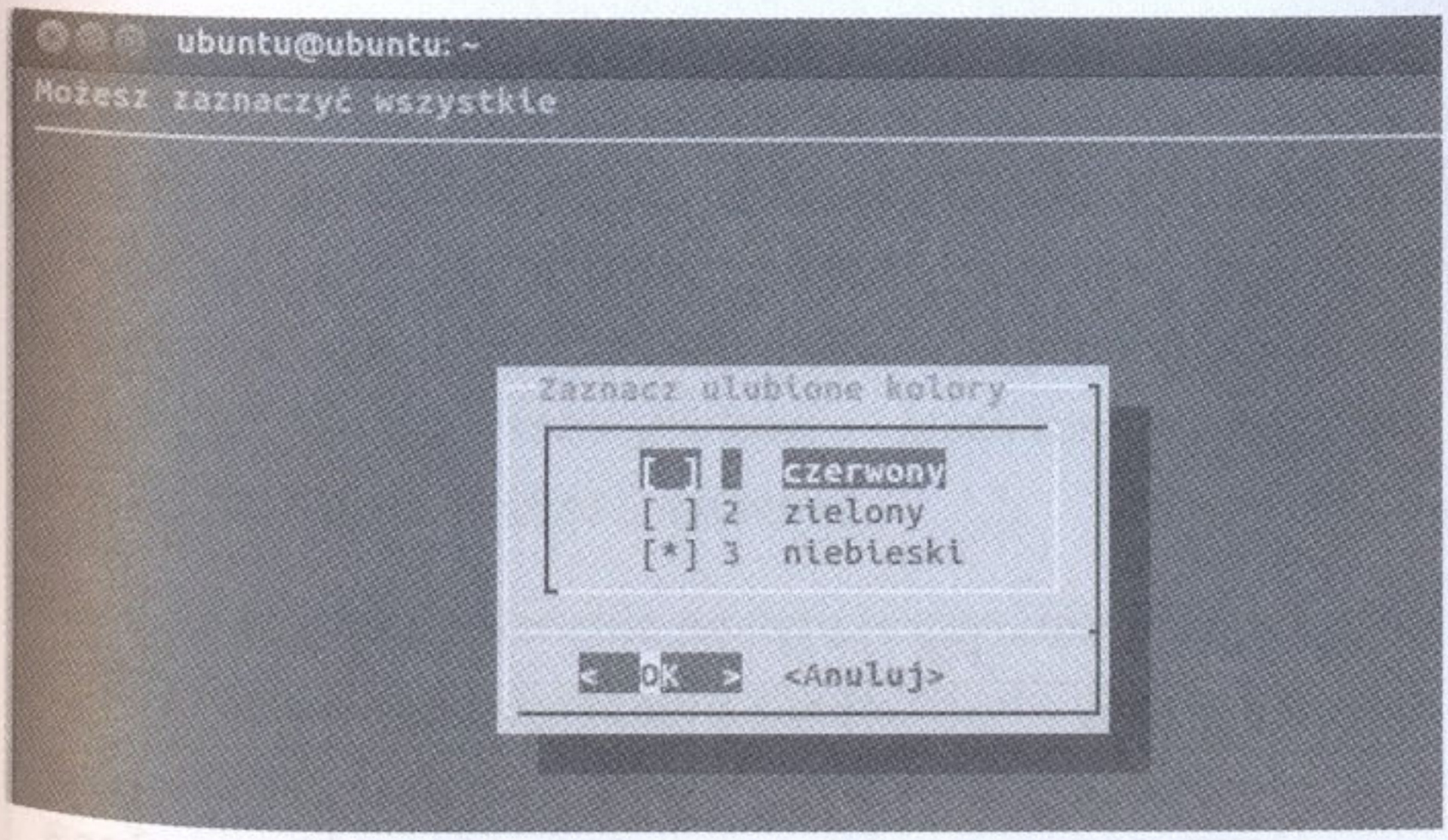
Składnia:

```
checklist tekst wysokość szerokość wysokość_listy lista
```

Przykład

```
dialog --title "Zaznacz ulubione kolory" --backtitle "Możesz zaznaczyć
wszystkie" --checklist "" 9 30 3 "1" "czerwony" "off" "2" "zielony"
"off" "3" "niebieski" "on"
```

Patrz rysunek 3.30.



Rysunek 3.30. Okno dialogowe — checklist

Okno inputbox

Składnia:

```
inputbox "Text" wysokość szerokość
```

Zadanie 3.27 Wywołaj okno dialogowe *inputbox*. Tytuł okna — „Pobieranie danych", tekst — „Do której klasy chodzisz", wysokość — 9, szerokość — 30.

Zadanie 3.28 Zmień przykład z rysunku 3.30 tak, by można było wybrać tylko jedną opcję. Zastosuj okno *radiolist*. Wszystkie potrzebne informacje znajdziesz w podręczniku *manual*.

Zadanie 3.29 Wykorzystaj operację przekierowania, żeby w pliku *katalogi.txt* umieścić listę katalogów znajdujących się w katalogu /. Wyświetl tę listę w oknie *textbox*.

Zadanie 3.30 Utwórz okno dialogowe, w którym użytkownik może zaznaczyć przedmioty zdawane przez niego na maturze.

Zadanie 3.31 Napisz skrypt *terminy*, po którego wykonaniu na ekranie pojawia się okno informujące o najbliższych zajęciach koła informatycznego.

3.13. Informacje sieciowe, komunikacja

3.13.1. Informacja o interfejsach sieciowych

```
ifconfig
```

3.13.2. Wyświetlenie aktywnych połączeń

```
sudo netstat -tup
```

3.13.3. Wyświetlenie bramy domyślnej

```
route -n
```

Przykład

```
ubuntu@ubuntu:~$ route -n
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface

3.13.4. Komunikacja między użytkownikami

Aby wysłać komunikat do innego użytkownika pracującego w systemie, korzystamy z polecenia:

```
write nazwa_użytkownika
```

Pisanie komunikatu kończy kombinacja klawiszy: *Ctrl+D*.

Przykład

Użytkownik *ubuntu* pisze komunikat do użytkownika *wiesia*:

```
ubuntu@ubuntu:~$ write wiesia
Komunikat
```

Na terminalu użytkownika *wiesia* pojawia się:

```
wiesia@ubuntu:~$
Message from ubuntu@ubuntu on pts/1 at 17:17 ...
Komunikat
EOF
```

Wskazówka

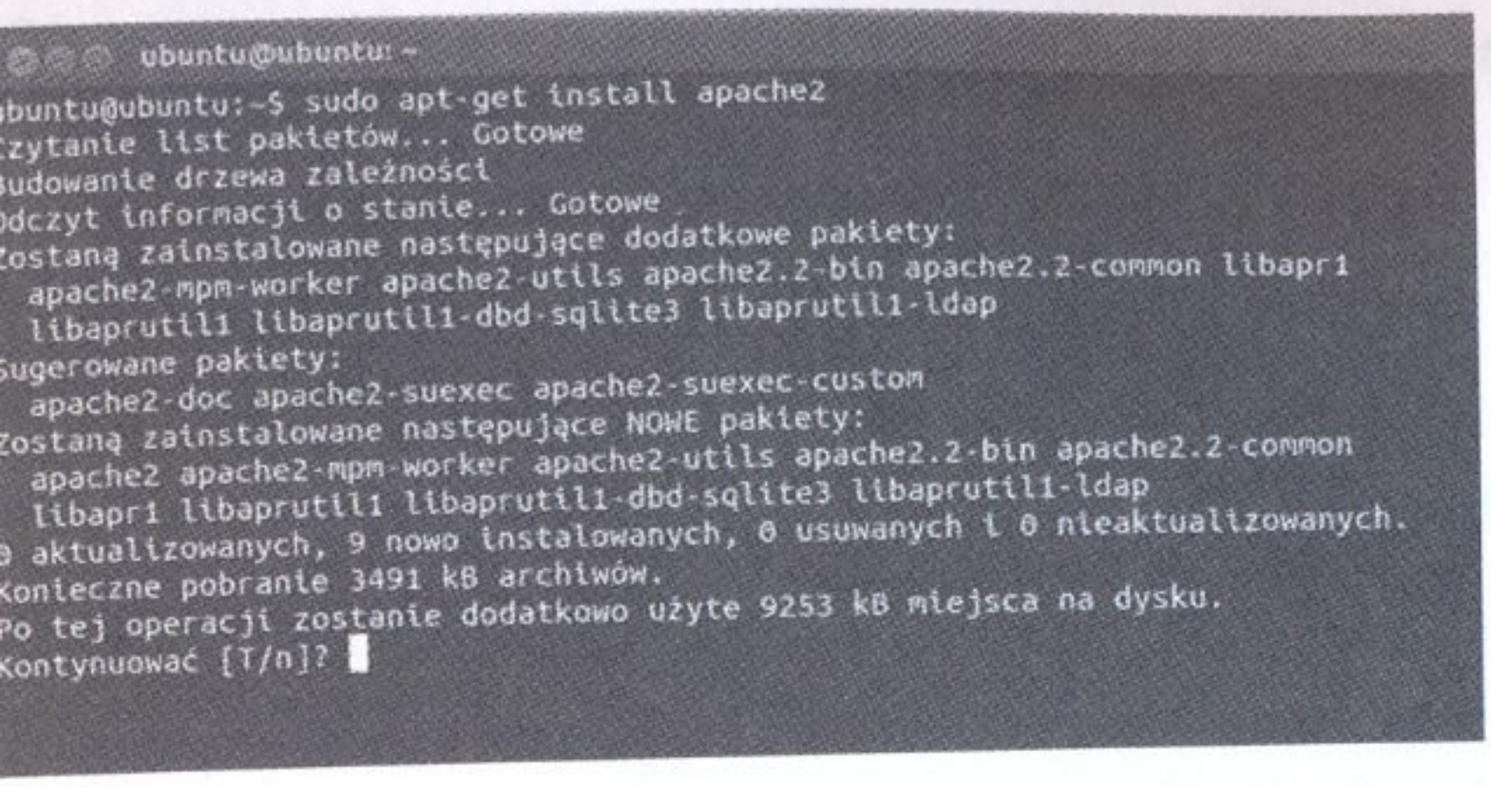
Żeby użytkownik mógł odbierać komunikaty, musi mieć odblokowany terminal — polecenie `mesg y`. Do blokowania go służy polecenie `mesg n`.

3.14. Serwer WWW

Serwer WWW to program obsługujący żądania protokołu internetowego HTTP. Najpopularniejszym serwerem jest Apache. Jest on zainstalowany w większości systemów Linux — wystarczy go tylko uruchomić. My zajmiemy się jego instalacją. Można to zrobić, ściągając pakiet *Apache HTTP Serwer metapackage* i przeprowadzając instalację z pakietu, co nie jest trudne. Można jeszcze prościej — jeżeli mamy połączenie z siecią Internet, wpisujemy:

```
sudo apt-get install apache2
```

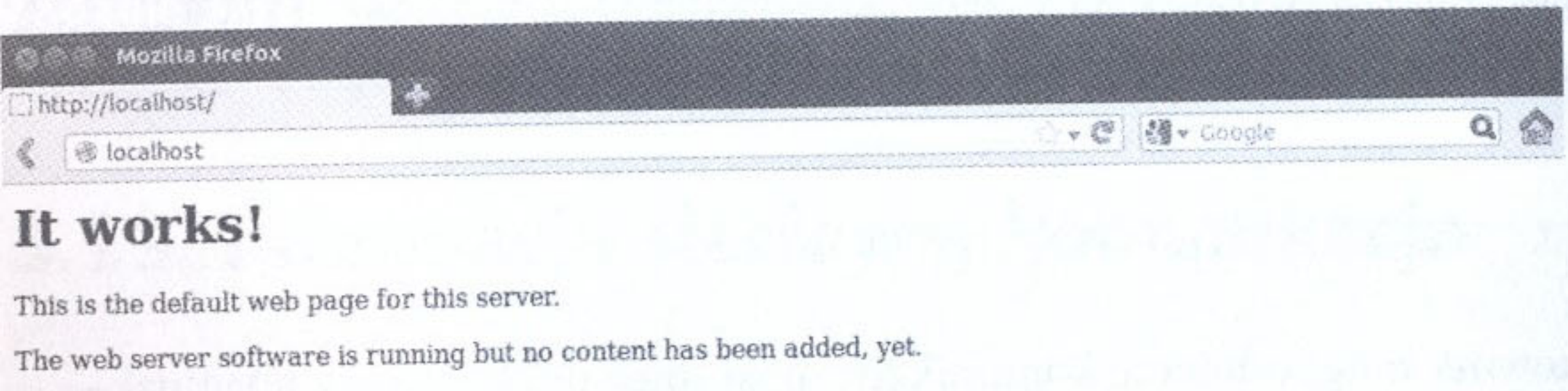
Po uruchomieniu polecenia następuje czytanie listy pakietów. W trakcie pracy pojawia się pytanie związane z używanym miejscem na dysku. Tylko po wyborze *T* serwer zostanie



Rysunek 3.31. Instalacja Apache

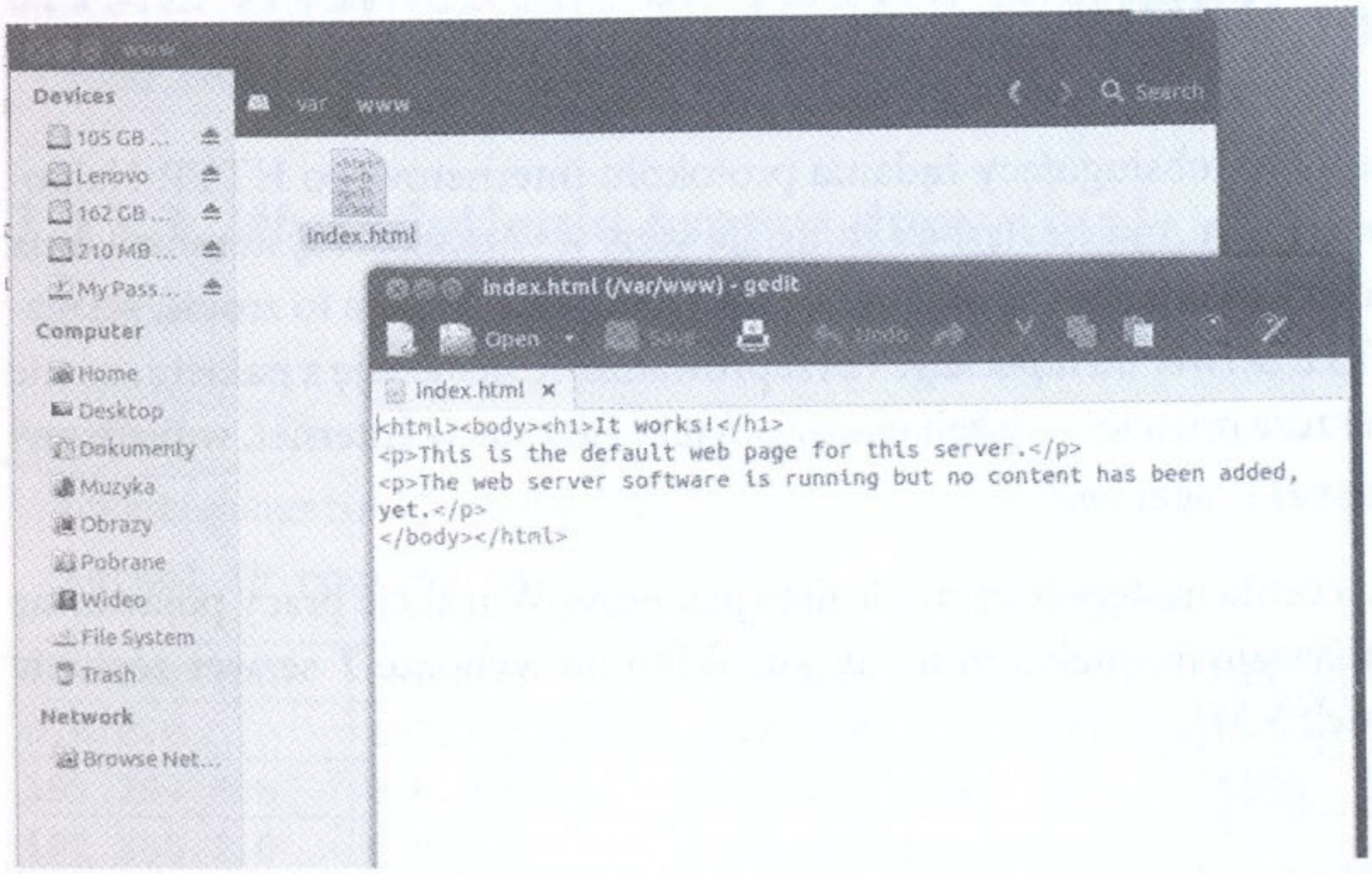
Jeżeli instalacja przebiegła prawidłowo, możemy uruchomić przeglądarkę internetową i wpisać adres `http://localhost`.

Wyświetli się zawartość pliku *index.html* (domyślna strona Apache) (rysunek 3.32).



Rysunek 3.32. Domyślna strona Apache

Plik *index.html* znajduje się w katalogu */var/www* (rysunek 3.33).



Rysunek 3.33. Lokalizacja i zawartość pliku index.html

Plik możemy edytować programem *gedit*, chociaż wygodniej jest wykorzystać program służący do tworzenia stron internetowych, np. *KompoZer*.

Żeby nasz serwer był w pełni funkcjonalny, tzn. obsługiwał dynamiczne strony WWW, skrypty PHP oraz bazy, trzeba doinstalować pakiet *php* i *mysql*:

```
sudo apt-get install php5
sudo apt-get install mysql-server
```

Bez serwera MySQL nie będzie działać CMS[5].

Wskazówka

Jeżeli po instalacji usługa nie działa automatycznie, należy ją uruchomić poleceniem:

```
sudo service nazwa_usługi start
```

Przykład

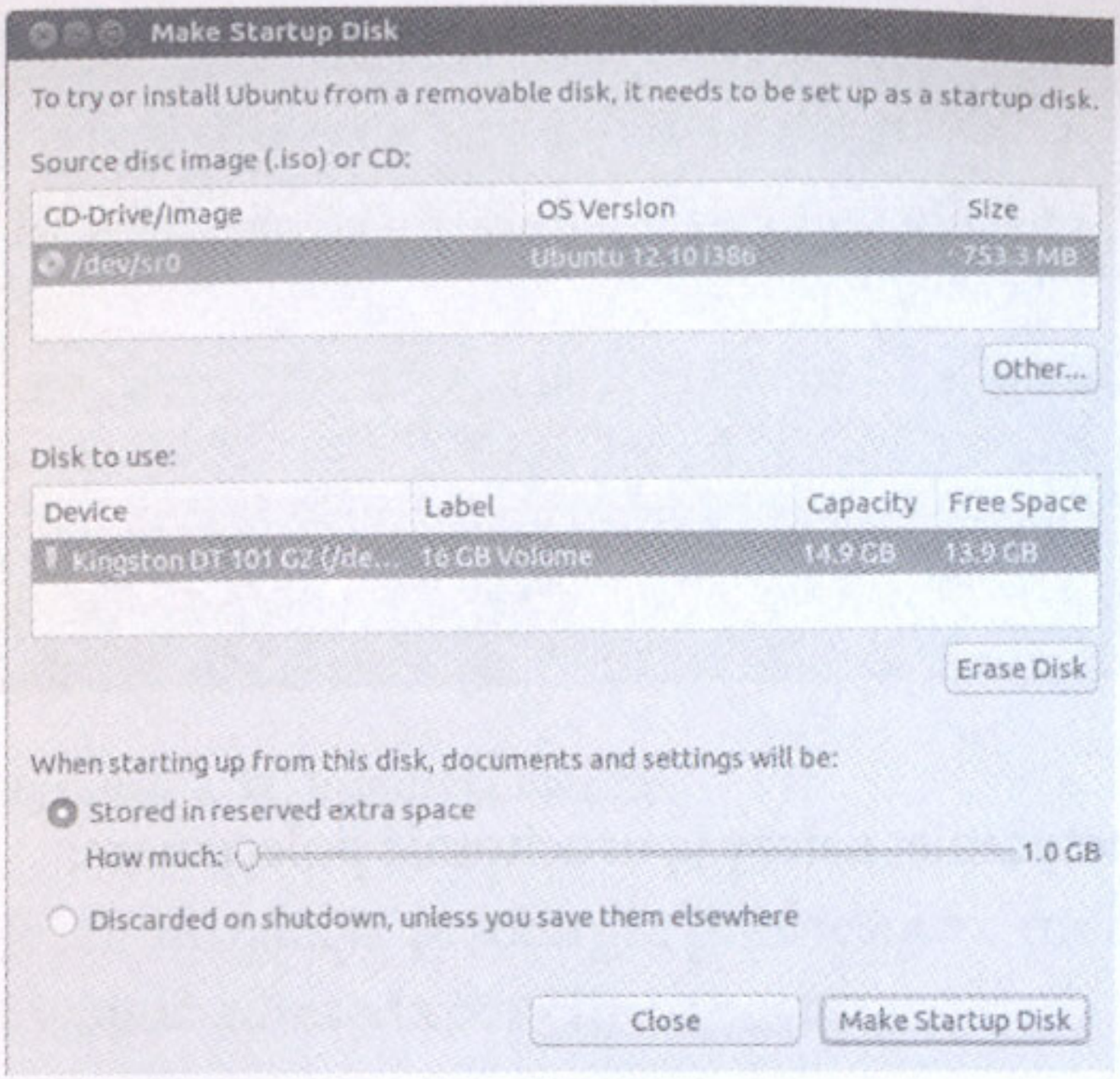
```
sudo service mysql start
```

Zadanie 3.32 Zainstaluj w swoim systemie serwery: Apache, PHP i MySQL.

3.15. Tworzenie dysków startowych z pamięci USB

Jeżeli do tej pory nie masz zainstalowanego systemu na swoim komputerze i stwierdzasz, że praca z LiveCD jest zbyt uciążliwa, możesz utworzyć dysk startowy z pamięci USB.

W tym celu po uruchomieniu systemu należy włożyć do gniazda USB pamięć flash (pojemność przynajmniej 1 GB). Następnie trzeba kliknąć *Dash Home*, odszukać program *Startup Disk Creator*, uruchomić go i postępować zgodnie z instrukcją (rysunek 3.34).



Rysunek 3.34. Tworzenie dysku startowego

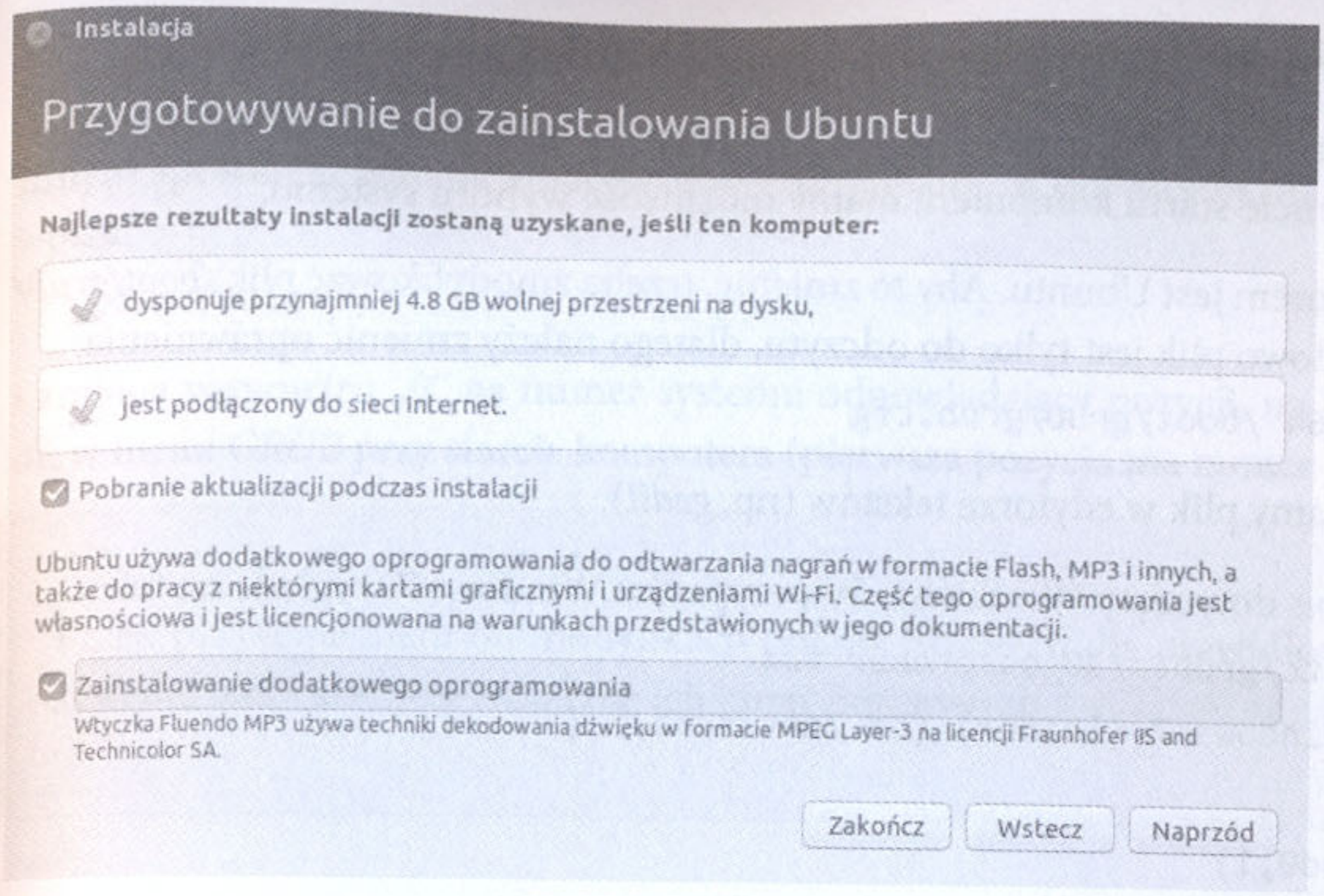
... 3.16. Instalacja Ubuntu obok Windowsa

Wskazówka

Jeżeli na komputerze mają być oba systemy, Ubuntu należy zainstalować jako drugi system.

Ze względów bezpieczeństwa przed instalacją należy zrobić backup wszystkich danych.

Instalację systemu najlepiej przeprowadzić na komputerze podłączonym do sieci Internet. Mamy wtedy możliwość pobrania aktualizacji i zainstalowania dodatkowego oprogramowania (rysunek 3.35).

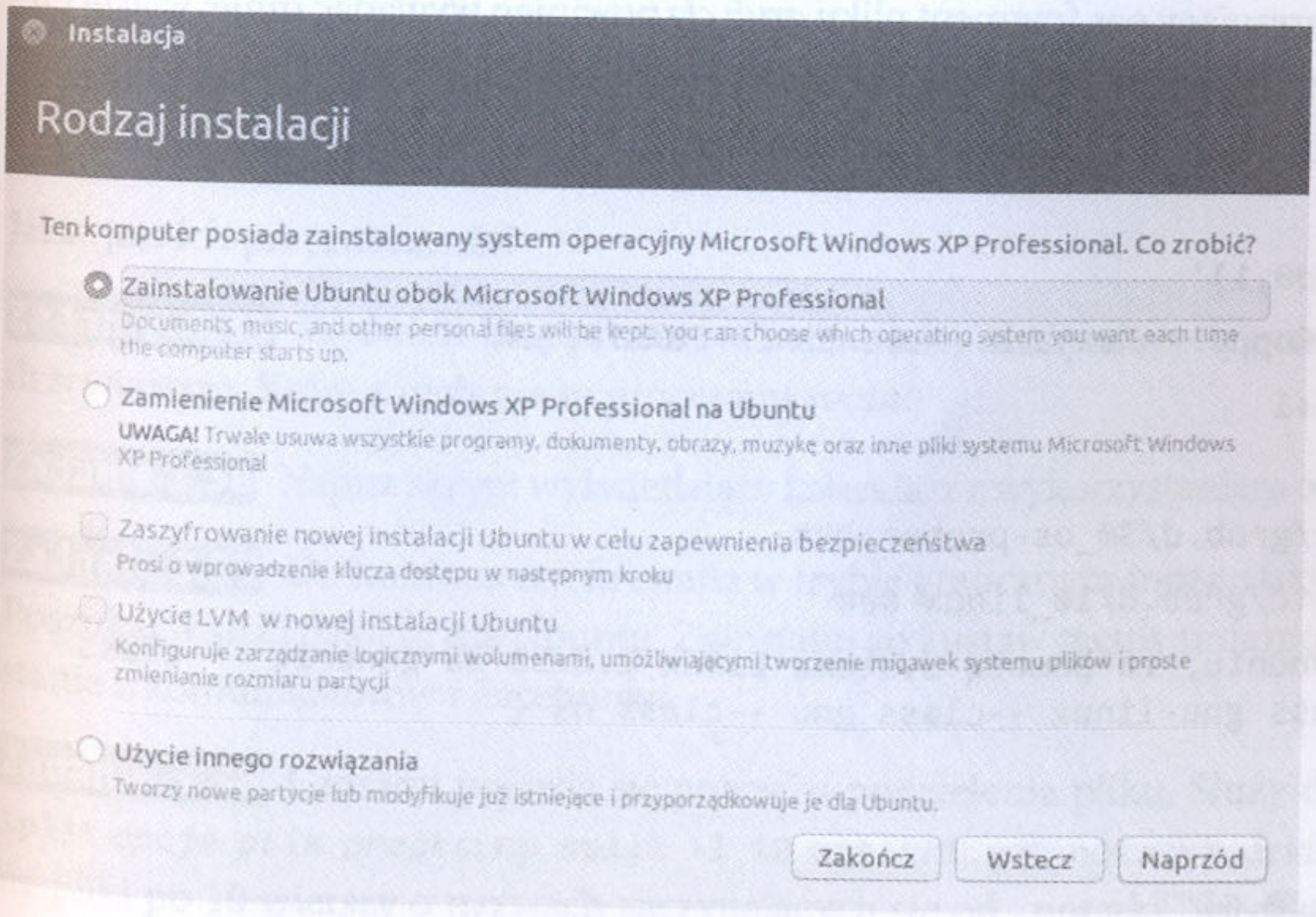


Rysunek 3.35. Przygotowanie do instalacji

Kolejnym ważnym krokiem jest wybór rodzaju instalacji. Jeżeli chcemy mieć dwa systemy, koniecznie wybieramy pierwszą opcję.

W razie pomyłki odzyskamy system Windows tylko z kopii zapasowej.

Patrz rysunek 3.36.



Rysunek 3.36. Wybór rodzaju instalacji

3.16.1. Zmiana kolejności uruchamiania systemów

Po zainstalowaniu w MBR dysku twardego zostanie umieszczony boot loader *GRUB*, dzięki któremu w momencie startu komputera mamy możliwość wyboru systemu.

Domyślnym wyborem jest Ubuntu. Aby to zmienić, trzeba zmodyfikować plik */boot/grub/grub.cfg*. Standardowo plik jest tylko do odczytu, dlatego należy zmienić uprawnienia:

```
sudo chmod 664 /boot/grub/grub.cfg
```

Następnie otwieramy plik w edytorze tekstów (np. *gedit*).

Odszukujemy wpis dotyczący Windowsa. W przypadku Windows 7 wygląda on tak:

```
### BEGIN /etc/grub.d/30_os-prober ###
menuentry "Windows 7 (loader) (on /dev/sda1)" {
insmod ntfs
set root='(hd0,1)'
search --no-floppy --fs-uuid --set 88ACF7EBACF7D224
chainloader +1
}
### END /etc/grub.d/30_os-prober ###
```

Należy go przenieść powyżej wpisu dotyczącego Linuksa zaczynającego się linią:

```
### BEGIN /etc/grub.d/10_linux ###
```

Po tej operacji interesujący nas fragment pliku *grub.cfg* powinien wyglądać mniej więcej tak:

```
### BEGIN /etc/grub.d/30_os-prober ###
menuentry "Windows 7 (loader) (on /dev/sda1)" {
insmod ntfs
set root='(hd0,1)'
search --no-floppy --fs-uuid --set 88ACF7EBACF7D224
chainloader +1
}
### END /etc/grub.d/30_os-prober ###
### BEGIN /etc/grub.d/10_linux ###
menuentry 'Ubuntu, za pomocą systemu Linux 2.6.32-38-generic' --class
ubuntu --class gnu-linux --class gnu --class os {
recordfail
insmod ext2
set root='(hd0,6)'
search --no-floppy --fs-uuid --set 65306d5a-626f-4731-b13a-98dcd0a4d132
linux /boot/vmlinuz-2.6.32-38-generic root=UUID=65306d5a-626f-4731-b13a-
98dcd0a4d132 ro quiet splash
```

```
initrd /boot/initrd.img-2.6.32-38-generic
}
```

Innym sposobem zmiany kolejności jest odszukanie w początkowym fragmencie pliku wpisu:

```
set default="0"
```

i zmiana parametru „0" na numer systemu odpowiadający pozycji, na której pojawia się on w menu *GRUB* przy starcie komputera (pierwsza pozycja ma numer 0).

Wskazówka

Przed przystąpieniem do modyfikacji plików istotnych dla prawidłowego działania systemu wskazane jest zrobienie ich kopii zapasowych.

3.17. Sprawdź się — zadania na koniec rozdziału

Zadanie 3.33 Uruchom terminal. Poleceniem `echo $SHELL` sprawdź nazwę powłoki tekstowej. Co to jest powłoka?

Zadanie 3.34 Wykonaj polecenia:

```
lsb_realase -a
sudo lshw
uname -a
ps -el
```

Jakie jest ich przeznaczenie?

Zadanie 3.35 Wyświetl listę procesów najbardziej obciążających procesor w formacie drzewiastym. Które z nich nie są procesami *roota*?

Zadanie 3.36 Napisz skrypt wyświetlający kalendarz z wykorzystaniem okna dialogowego.

Zadanie 3.37 Do konfiguracji firewalla w trybie graficznym można użyć aplikacji *Gufw*. Poszukaj w sieci pakietu dla Ubuntu. Zainstaluj go i ustaw reguły uniemożliwiające korzystanie z komunikatorów i Facebooka.

Zadanie 3.38 Czasami pojawia się potrzeba podzielenia pliku. Służy do tego polecenie `split` *`opcje plik prefiks`*; np. `split -l 10 notatki.odt notatki` dzieli plik *notatki.odt* na pliki po 10 wierszy o nazwach zaczynających się od „notatki" (prefiks). Napisz skrypt dzielący plik, w którym nazwa pliku i liczba wierszy są podawane jako parametry.

Zadanie 3.39 Sprawdź aktywne połączenia w systemie.