

Mój @ dawidkoch.szkoła@gmail.com Proszę o wysłanie na ten adres prac z poprzedniego tygodnia i wszelkich zaległości.

Na ostatniej stronie znajdują się 3 pytania. Wszyscy mają na nie odpowiedzieć i wysłać mi na maila do piątku 3 kwietnia.

„Wielodostępny system operacyjny Linux”

Oczywiście zakładam, że macie zainstalowanego Linuxa ☺

Czyli omawiamy architekturę systemu Linux, zarządzanie procesami, tworzenie kont użytkowników, zarządzanie grupami i użytkownikami, wielodostępność, system plików. Tematyka jest rozległa, więc przewiduję jego realizację na 4h lekcyjne (czyli ten tydzień i następny).

Zanim prześlę Wam zdjęcia z książki, pokażę kilka stron, które w dość ciekawy sposób pokazują omawiany przez nas materiał:

<https://pl.if-koubou.com/articles/how-to/how-to-manage-processes-from-the-linux-terminal-10-commands-you-need-to-know.html>

<https://itporady.pl/linux/zarzadzanie-uzytkownikami-w-systemie-linux/>

<https://pasja-informatyki.pl/sieci-komputerowe/linux-podstawowe-pojecia/>

<https://pl.if-koubou.com/articles/how-to/which-linux-file-system-should-you-use.html>

13

Wielodostępny system operacyjny Linux

ZAGADNIENIA

- Podstawowe wiadomości dotyczące architektury systemu Linux
- Zarządzanie procesami w systemie Linux
- Konto użytkownika w systemie Linux
- Konta użytkowników w środowisku tekstowym i graficznym
- Wielodostępność w systemie Linux

Linux jest systemem **wielozadaniowym**, co umożliwia mu równoczesne wykonywanie więcej niż jednego procesu (**proces** jest to egzemplarz wykonywanego programu, mający własną przestrzeń adresową). Za realizację wielozadaniowości odpowiada jądro systemu operacyjnego.

Wielozadaniowość jest zapewniona między innymi przez program nazywany **planistą**, który realizuje algorytm szeregowania zadań w kolejce do przyznania czasu procesora. Linux może wykorzystywać wiele procesorów. Gdy system ma mniej dostępnych procesorów niż zadań do wykonania, wtedy czas działania procesora jest dzielony między wszystkie zadania.

13.1. Wiadomości wstępne dotyczące architektury systemu Linux

Niektóre systemy wielozadaniowe (np. Linux) oferują **wyłączenie**, które polega na przerwaniu wykonywania procesu, odebraniu mu czasu przydzielonego do wykonania zadania przez procesor i przekazaniu sterowania do planisty. W systemach bez wyłączania procesy muszą same dbać o sprawiedliwy podział czasu procesora.

Do komunikowania się użytkownika z jądrem systemu jest wykorzystywana **powłoka systemowa** (ang. *shell*). Jest to program, który pełni funkcję pośrednika między jądrem systemu i użytkownikiem. Program ten udostępnia interfejs w postaci wiersza poleceń. Jądro systemu zawiera podprogramy potrzebne do obsługi urządzeń sprzętowych, przeprowadzania operacji wejścia i wyjścia, zarządzania plikami, powłoka natomiast pozwala korzystać z tych podprogramów za pomocą wiersza poleceń. Powłoka zawiera podstawowe polecenia, jednak gdy wydane przez użytkownika polecenie nie jest wbudowane, wtedy uruchamia się program zewnętrzny. Po zalogowaniu do systemu użytkownik uzyskuje **znak zachęty** (ang. *shell prompt*), np. \$ lub #. Po znaku zachęty użytkownik może wpisywać polecenia dla systemu.

Składnia polecenia jest następująca:

```
uczen@uczen-VirtualBox:~$ polecenie opcje argumenty,
```

gdzie:
uczen@
komput
nika ro
polec
opcje
argum

W L
na inn
tów pe
Do

Rys.

M
on v

L

F

N

stał

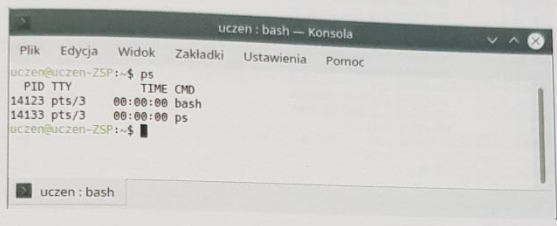
-
-
-

ko

gdzie:
uczen@uczen-VirtualBox: ~\$ – standardowy znak zachęty określający nazwę konta, komputera oraz bieżący katalog; symbol # poprzedza polecenia wpisujące nazwę konta, nika **root**, natomiast \$ – polecenia pozostałych użytkowników;
polecenie – określa, co system ma wykonać;
opcje – zawierają dodatkowe informacje o sposobie wykonania polecenia;
argumenty – określają obiekty, których dotyczy polecenie.

W Linuksie najczęściej używana jest powłoka **bash**, lecz użytkownik może zmienić ją na inną. Powłoki obsługują również język programowania umożliwiający pisanie skryptów powłoki.

Do wyświetlenia listy uruchomionych procesów używa się polecenia **ps** (rys. 13.1).



Rys. 13.1. Lista uruchomionych procesów

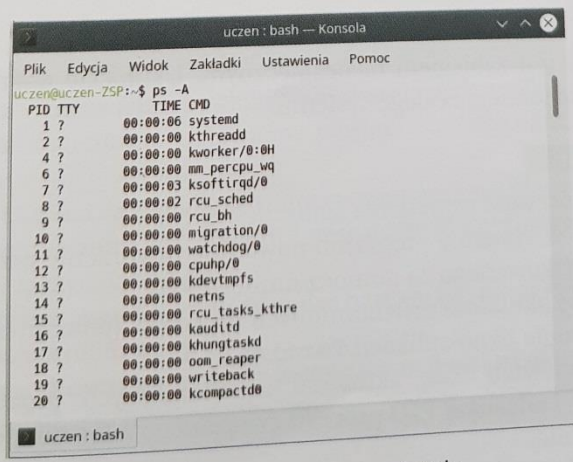
Może go używać każdy użytkownik, ale wyświetlane będą tylko te procesy, których jest on właścicielem.

Listę wszystkich procesów można wyświetlić za pomocą polecenia **ps** z opcjami **-aux**, np. **ps -aux**.

Najważniejszą spośród wyświetlanych informacji jest **identyfikator procesu PID**. Pozostałe pola oznaczają:

- **TTY** – identyfikator konsoli, z której proces został uruchomiony;
- **TIME** – czas procesora wykorzystany do tej pory;
- **CMD** – polecenie, którym uruchomiono proces.

Administrator może wyświetlić listę procesów uruchomionych przez wszystkich użytkowników za pomocą polecenia **ps -A** (rys. 13.2).



Rys. 13.2. Lista wszystkich procesów uruchomionych w systemie

13.2. Zarządzanie procesami w systemie Linux

Linux jest systemem bardzo stabilnym, lecz i w nim proces może się zawiesić. Czasami zachodzi konieczność usunięcia procesu działającego. Zwykły użytkownik może usuwać tylko te procesy, których jest właścicielem. Administrator może usuwać wszystkie procesy. Do usuwania procesów z systemu służy polecenie **kill** (rys. 13.3).

```

Plik  Edycja  Widok  Zakładki  Ustawienia  Pomoc
18671 ?      00:00:02 pulseaudio
18672 ?      00:00:00 rtkit-daemon
18679 ?      00:00:01 kactivitymanag
18686 ?      00:00:00 at-spi-bus-lau
18693 ?      00:00:00 dbus-daemon
18698 ?      00:00:00 gconf-helper
18700 ?      00:00:00 at-spi2-registr
18700 ?      00:00:00 gconfd-2
18702 ?      00:00:00 org_kde_powerde
18710 ?      00:00:00 kscreen_backend
18737 ?      00:00:00 obexd
18767 ?      00:00:00 kuiserver
18773 ?      00:00:00 ksysguardd
19214 ?      00:00:00 systemd-resolve
29919 ?      00:00:00 systemd-timesyn
29923 ?      00:00:00 systemd-journal
29926 ?      00:00:00 polkitd
32098 ?      00:00:00 update-apt-xapi
32609 ?      00:00:00
uczen@uczen-ZSP:~$ kill 14558
uczen@uczen-ZSP:~$
  
```

Rys. 13.3. Usunięcie procesu

Polecenie to musi otrzymać jako argument identyfikator PID procesu. Usuwa ono proces w sposób bezpieczny (zamyka otwarte pliki, zwalnia obszary pamięci). Jeżeli usunięcie procesu w ten sposób się nie uda, można wymusić jego zakończenie, dodając opcję **-9**, np.

```
sudo kill -9 2463.
```

Inne opcje związane z poleceniem **kill**:

- **-2** – przerwanie wykonywania procesu nakazane z klawiatury [Ctrl] + [C];
- **-3** – zakończenie wykonywania procesu z jednoczesnym utworzeniem pliku, który zawiera obraz pamięci procesu (plik **core**);
- **-15** – domyślny sygnał polecenia **kill**;
- **-19** – zatrzymanie wykonywania procesu z możliwością późniejszego wznowienia.

! UWAGA

Usuwanie procesów jest zabiegiem niebezpiecznym. Usunięcie ważnego procesu może spowodować unieruchomienie całego systemu.

PRZYKŁAD 13.1

Zarządzanie procesami w systemie Linux

Uruchomienie w systemie programu powoduje utworzenie przynajmniej jednego procesu identyfikowanego za pomocą numeru PID.

1. Wyświetl listę aktualnie uruchomionych procesów (polecenie **ps -A**).
2. Uruchom drugie okno aplikacji **Terminal**.
3. Ponownie wyświetl listę aktualnie uruchomionych procesów, porównaj ją z poprzednią i odzyskaj PID procesu związanego z terminalem uruchomionym w drugim oknie.
4. Za pomocą identyfikatora PID zakończ działanie terminalu w drugim oknie.

13.3. Tworzenie kont użytkowników w środowisku tekstowym

W systemach wielodostępnych z systemu może jednocześnie korzystać wielu użytkowników. Każdy z nich ma przydzielone zasoby, np. pamięć, czas procesora. W systemie istnieje system kont pozwalających na zarządzanie poszczególnymi użytkownikami oraz ich jednoznaczne identyfikowanie za pomocą unikatowych nazw, podawanych podczas logowania. **Konto** to wszystkie pliki, zasoby i informacje należące do użytkownika. Każdy użytkownik jest identyfikowany przez unikatową liczbę całkowitą – **id użytkownika** (UID). **Katalogi domowe** użytkowników są przeznaczone do przechowywania zbiorów określających ich środowisko pracy (np. pulpit) oraz innych zbiorów prywatnych. Każdy użytkownik ma wszystkie prawa do własnego katalogu domowego. Standardowym miejscem dla katalogów domowych jest **/home**.

W celu łatwiejszego zarządzania uprawnieniami są tworzone **grupy użytkowników**.

Uprawnienia przyznane grupie użytkowników dotyczą wszystkich użytkowników będących członkami tej grupy. Uprawnienia wynikające z przynależności do wielu grup sumują się, np., jeżeli **uczen** należy do grupy **Wszyscy** i grupy **Uczniowie**, to jego efektywne uprawnienia są sumą uprawnień nadanych grupie **Wszyscy**, grupie **Uczniowie** i indywidualnie uczniowi.

Aby zarządzać użytkownikami, należy mieć odpowiednie uprawnienia (domyślnie ma je **root**). Do zarządzania użytkownikami i grupami najczęściej są używane następujące polecenia:

1. **useradd** – dodawanie nowego użytkownika, np.

```
sudo useradd uczen
```

utworzy konto **uczen**. Konto to nie ma ustawionego hasła ani innych właściwości.

Najważniejsze opcje polecenia **useradd**:

- **-c komentarz** – dodanie komentarza (imię i nazwisko użytkownika) do pola komentarza w pliku **hasel**;
 - **-d katalog_domowy** – wskazanie katalogu domowego dla nowego użytkownika, standardowo jest tworzony nowy katalog w **/home**;
 - **-m** – tworzenie katalogu domowego, jeśli nie istnieje;
 - **-g początkowa_grupa** – numer lub nazwa początkowej grupy logowania użytkownika;
 - **-G grupa [, ...]** – grupa lub lista grup, do których również ma należeć tworzony użytkownik;
 - **-s powłoka** – ustawienie powłoki systemowej użytkownika, domyślnie jest wybierana powłoka **systemowa**. Ustawienie powłoki na **/sbin/nologin** powoduje, że użytkownik nie może się zalogować;
 - **-u id_użytkownika** – podanie numerycznej wartości identyfikatora użytkownika (UID). Numer ten musi być dodatni i unikatowy (domyślnie jest ustawiana wartość najmniejsza, zaczynając od 500);
 - **-p zakodowane_hasło** – należy podać hasło w formie zakodowanej, np. takie, jak utworzone poleceniem **crypt**;
 - **- nazwa_użytkownika** – nazwa logowania nowego użytkownika.
2. **passwd** – zmienia hasło użytkownika, np.:

```
sudo passwd uczen
```

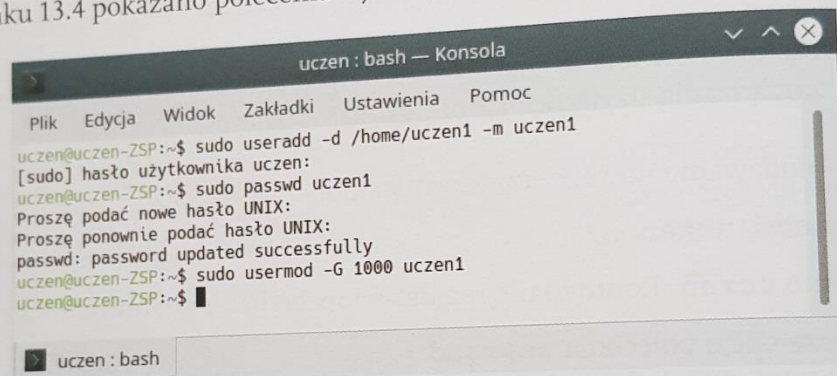
Jeżeli nie zostanie wskazana nazwa konta, zmienione hasło użytkownika konta, z którego polecenie zostało wykonane zostanie zapisane jako nowe hasło, a następnie o nowe hasło, to program zapyta nas najpierw o stare hasło, a następnie o nowe. Gdy wpisujemy dwukrotnie nowe hasło, program zweryfikuje je pod kątem wymaganej złożoności. Jeżeli hasło jest zbyt łatwe, użytkownik otrzyma stosowny komunikat, ale hasło zostanie przyjęte.

3. **usermod** – zmienia ustawienia użytkownika. Można używać takich samych opcji jak w poleceniu **useradd** oraz:
- **-L** – blokuje hasło użytkownika; opcja ta powoduje wstawienie „!” na początku zakodowanego hasła;
 - **-U** – odblokuje hasło użytkownika.
4. **userdel** – kasuje konto użytkownika. Katalog domowy nie zostanie usunięty. Jeżeli katalog domowy ma być również usunięty, trzeba użyć opcji **-r**.

Aby użytkownik mógł zalogować się do systemu, należy:

- utworzyć dla niego konto;
- wprowadzić hasło dla użytkownika.

Na rysunku 13.4 pokazano polecenia wykorzystywane do zarządzania użytkownikami.



```
uczen : bash — Konsola
Plik  Edycja  Widok  Zakładki  Ustawienia  Pomoc
uczen@uczen-ZSP:~$ sudo useradd -d /home/uczen1 -m uczen1
[sudo] hasło użytkownika uczen:
uczen@uczen-ZSP:~$ sudo passwd uczen1
Proszę podać nowe hasło UNIX:
Proszę ponownie podać hasło UNIX:
passwd: password updated successfully
uczen@uczen-ZSP:~$ sudo usermod -G 1000 uczen1
uczen@uczen-ZSP:~$
```

Rys. 13.4. Zarządzanie użytkownikami w trybie tekstowym

PRZYKŁAD 13.2

Zarządzanie użytkownikami

Należy utworzyć użytkownika **uczen1** z hasłem **123**. W tym celu wykonaj czynności przedstawione poniżej.

1. Zaloguj się na konto użytkownika **root**.
2. Utwórz konto użytkownika poleceniem:

```
sudo useradd uczen1
```

3. Zmień hasło użytkownika poleceniem:

```
sudo passwd uczen1
```

4. Zaloguj się na konto utworzonego użytkownika i sprawdź zawartość jego katalogu domowego

Aby zarządzać grupami użytkowników, trzeba mieć odpowiednie uprawnienia (domyślnie ma je **root**). Do zarządzania grupami najczęściej są używane następujące polecenia:

1. **groupadd** – dodawanie nowej grupy użytkowników, np.:

```
sudo groupadd uczniowie
```

– utworzy grupę **uczniowie**.

Najważniejsze opcje polecenia **groupadd**:

- **-g grupa** – identyfikator grupy (GID);
 - **-p zakodowane_hasło** – należy podać hasło w formie zakodowanej, np. utworzone poleceniem **crypt**;
2. **groupmod** – zmienia ustawienia grupy. Można używać takich samych opcji, jak w poleceniu **groupadd**;
 3. **groupdel** – kasuje grupę użytkowników.

PRZYKŁAD 13.3

Zarządzanie grupami użytkowników

Aby utworzyć grupę użytkowników i przydzielić do niej użytkowników, wykonaj czynności przedstawione poniżej.

1. Zaloguj się na konto użytkownika **root**.
2. Utwórz grupę użytkowników **uczniowie**, np. poleceniem:

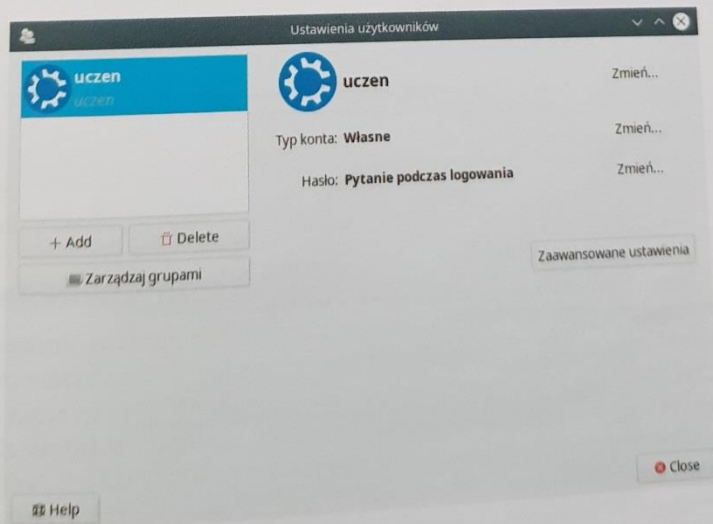
```
sudo groupadd uczniowie
```

3. Dodaj utworzone w przykładzie 13.2 konto **uczen1** do grupy **uczniowie**, np. poleceniem:

```
sudo usermod -G uczniowie uczen1
```

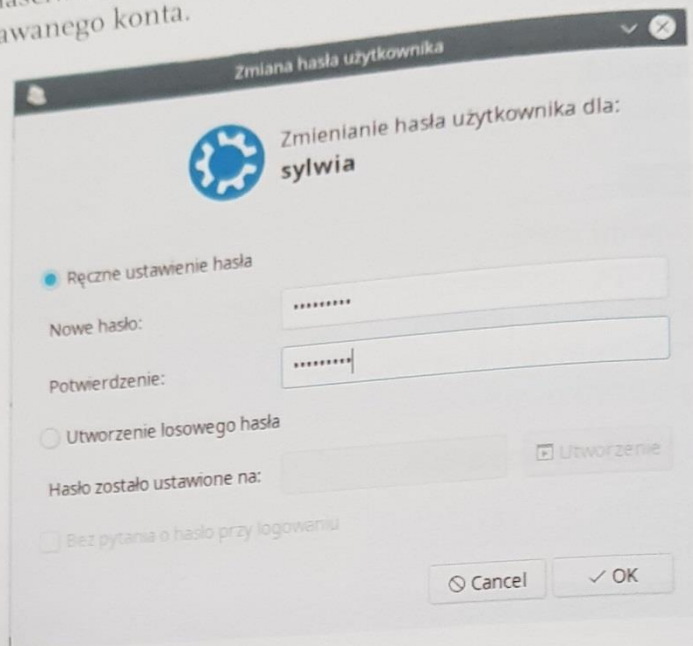
13.4. Zarządzanie grupami i użytkownikami w środowisku graficznym

W środowisku graficznym istnieje wygodne narzędzie do zarządzania użytkownikami i grupami **Users and Groups (Ustawienia użytkownika)**. Można je uruchomić, wybierając **Start/System/Programy**. Po wyborze narzędzia **Users and Groups** uruchomi się okno **Ustawienia użytkownika** (rys. 13.5). Narzędzie to pozwala jednocześnie zarządzać użytkownikami i grupami. Jeśli nie jest ono widoczne w systemie, trzeba je wgrać z **Ośrodka programów**.



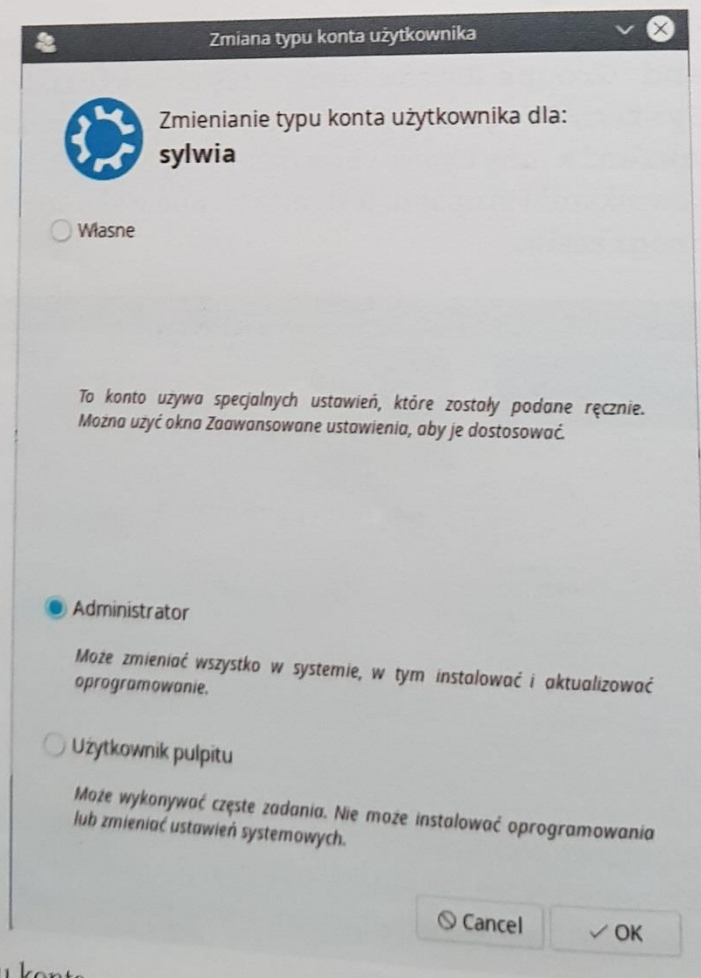
Rys. 13.5. Okno Ustawienia użytkownika

Aby dodać nowe konto, należy kliknąć przycisk **Add**, podać hasło administratora, a następnie określić właściwości tworzonego konta, nazwę i hasło. Na rys. 13.6 przedstawiono nazwę i hasło dodawanego konta.



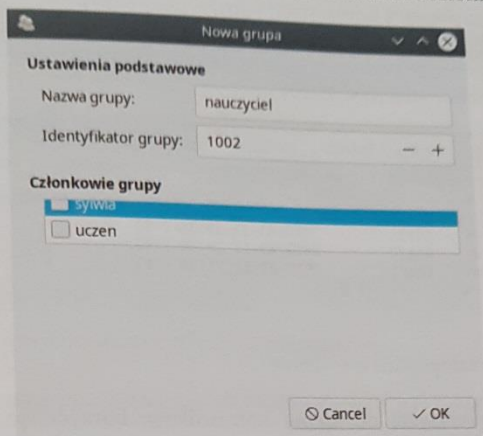
Rys. 13.6. Szczegóły dodawanego konta

Po utworzeniu konta istnieje też możliwość zmiany jego typu. W tym celu należy w polu **Typ konta** kliknąć opcję **Zmień** i wybrać odpowiedni typ (rys. 13.7).



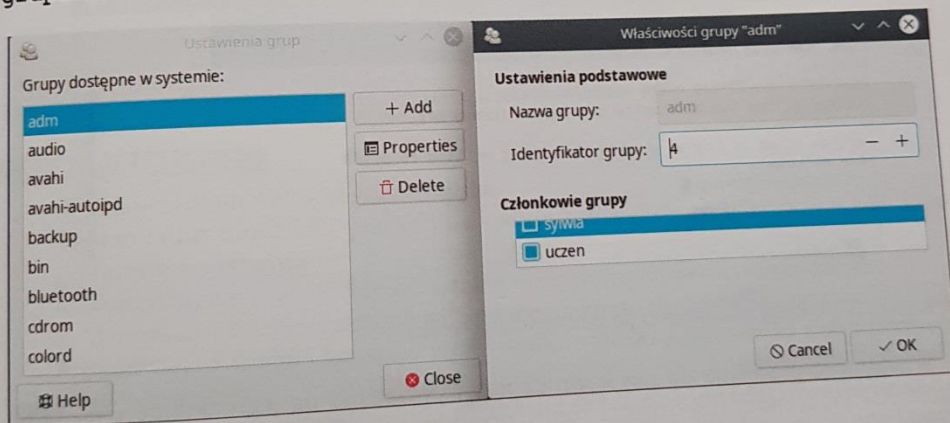
rys. 13.7. Zmiana typu konta

Aby dodać nową grupę, należy w oknie **Ustawienia użytkowników** kliknąć zakładkę **Zarządzanie grupami** i w oknie **Ustawienia grup** kliknąć przycisk **Add**. Następnie trzeba podać jej nazwę, identyfikator i ewentualnie wskazać członków (rys. 13.8).



Rys. 13.8. Tworzenie nowej grupy i dodawanie jej członków

Jeśli chcemy sprawdzić, kto jest członkiem konkretnej grupy, klikamy opcję **Zarządzaj grupami**; klikamy w nią dwukrotnie lub wybieramy zakładkę **Properties** (rys. 13.9).



Rys. 13.9. Okno Właściwości grupy

13.5. Wielodostępność

Każdy użytkownik systemu ma przydzielone zasoby (np. pamięć, czas procesora, miejsce na dysku), do których nikt oprócz niego nie ma dostępu, chyba że sam je udostępni. Użytkownik jest niezależny i odizolowany od innych. O uprawnieniach do korzystania z poszczególnych zasobów decyduje administrator systemu. Aby zalogować się do systemu, można skorzystać z klawiatury przyłączonej do komputera lub sieci komputerowej.

Po zalogowaniu uzyskuje się dostęp do konsoli. W Linuksie jest zdefiniowanych wiele **konsoli tekstowych**, na których mogą logować się różni użytkownicy. Aby zmienić konsolę, używa się kombinacji klawiszy [Ctrl] + [Alt] + [Fn], np. [Ctrl] + [Alt] + [F1], aby przełączyć się

na pierwszą konsolę. Ograniczeniem bezpłatnego środowiska graficznego jest możliwość uruchomienia go tylko jeden raz. Oznacza to, że tylko jeden użytkownik może pracować w środowisku graficznym. Środowisko graficzne jest dostępne na siódmej konsoli, którą można uruchomić za pomocą kombinacji [Ctrl] + [Alt] + [F7].

Aby sprawdzić, z jakiego konta w danym momencie się korzysta, należy użyć polecenia **whoami**. Użytkownik może podejrzec, kto w danym momencie jest zalogowany do systemu za pomocą polecenia **who** (rys. 13.10).

```

Plik  Edycja  Widok  Zakładki  Ustawienia  Pomoc
uczen@uczen-ZSP:~$ whoami
uczen
uczen@uczen-ZSP:~$ who
uczen    tty1      2019-03-11 20:36 (:0)
uczen    pts/0     2019-03-11 20:36 (:0)
uczen    pts/1     2019-03-11 21:28 (:0)
uczen@uczen-ZSP:~$ █
    
```

Rys. 13.10. Użytkownicy zalogowani w systemie

Użytkownicy mogą wysyłać do siebie komunikaty, korzystając z polecenia **writ** (rys. 13.11). Po poleceniu **write** należy wpisać nazwę użytkownika zalogowanego do systemu. Wpisywanie tekstu wiadomości można zakończyć za pomocą kombinacji klawiszy [Ctrl] + [D].

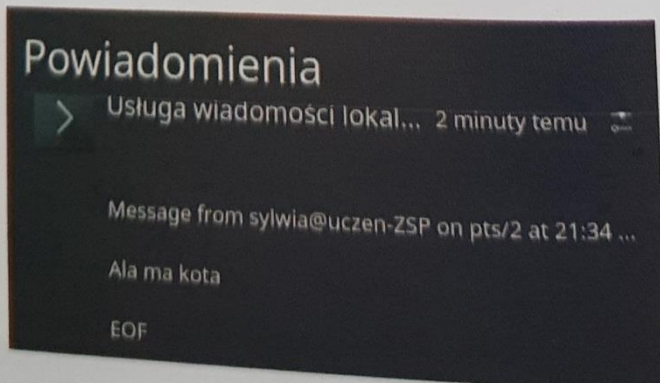
```

sylwia : bash — Konsola
Plik  Edycja  Widok  Zakładki  Ustawienia  Pomoc
See "man sudo_root" for details.
sylwia@uczen-ZSP:~$ write uczen
write: write: you have write permission turned off.

write: uczen is logged in more than once; writing to pts/0
Ala ma kota
sylwia@uczen-ZSP:~$ █
    
```

Rys. 13.11. Wysłanie wiadomości do innego użytkownika

Wiadomość przesłana do użytkownika jest wyświetlana na jego ekranie (rys. 13.12). Skrót EOF (ang. *end of file*) oznacza koniec przesyłanego tekstu.



Rys. 13.12. Wiadomość wyświetlona na ekranie odbiorcy

Administrator czasami musi wysłać wiadomość do wszystkich zalogowanych użytkowników, np. z prośbą o zakończenie pracy z powodu przerwy w pracy serwera. Do tego celu może użyć polecenia `wall` (rys. 13.13).

```

sylwia : bash — Konsola
Plik  Edycja  Widok  Zakładki  Ustawienia  Pomoc

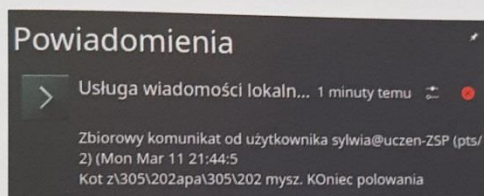
sylwia@uczen-ZSP:~$ wall
Kot złapał mysz. KOniec polowania

Zbiorowy komunikat od użytkownika sylwia@uczen-ZSP (pts/2) (Mon Mar 11 21:44:5
Kot z\305\202apa\305\202 mysz. KOniec polowania

sylwia@uczen-ZSP:~$ █
sylwia : bash
  
```

Rys. 13.13. Komunikat wysyłany do wszystkich użytkowników

Komunikat otrzymany przez wszystkich użytkowników pokazano na rys. 13.14.



Rys. 13.14. Komunikat odebrany przez wszystkich użytkowników

PRZYKŁAD 13.4

Wysyłanie komunikatów do uczestników

System Linux umożliwia przesyłanie komunikatów do innych użytkowników zalogowanych w systemie.

1. Skorzystaj z pierwszej konsoli i zaloguj się na konto użytkownika `root`.
2. Skorzystaj z drugiej konsoli i zaloguj się na konto użytkownika `uczen`.
3. Skorzystaj z drugiej konsoli i wyślij wiadomość do użytkownika `root` (użyj polecenia `write`).
4. Skorzystaj z pierwszej konsoli i przeczytaj wiadomość od użytkownika `uczen`.
5. Skorzystaj z pierwszej konsoli i wyślij do wszystkich zalogowanych użytkowników wiadomość o planowanej przerwie w pracy serwera (użyj polecenia `wall`).
6. Skorzystaj z drugiej konsoli i odczytaj wiadomość od użytkownika `root`.

13.6. System plików systemu Linux

System plików to metoda przechowywania plików, zarządzania plikami, informacjami o nich tak, aby dostęp do plików i danych w nich zgromadzonych był łatwy dla użytkownika systemu. Linux potrafi obsługiwać wiele różnych systemów plików, w tym również te najpopularniejsze:

- `ext4` – obecnie jeden z najpopularniejszych systemów plików, następca `ext3`. W wersji jądra 2.6.28 uznany za stabilny. Umożliwia obsługę woluminów do 1 eksbibajta.

Ponadto oferuje możliwość rezerwowania obszaru dla nowych plików, obniżając ryzyko fragmentacji danych, zmniejszając liczbę potrzebnych metadanych i zwiększając wydajność operacji odczytu oraz zapisu (włączenie tej funkcji powoduje utratę kompatybilności z **ext3**);

- **ext3** – domyślny system plików w większości dystrybucji systemu Linux opartych na jądrze 2.4 oraz nowszych. W starszych dystrybucjach można spotkać wersję **ext2**;
- **FAT** – system plików używany w środowisku DOS/Windows;
- **NTFS** – nowsza wersja systemu plików dla Windows (niektóre starsze dystrybucje mogą mieć problemy z obsługą zapisu);
- **ISO9660** – stosowany na dyskach CD/DVD.

Podczas instalacji Linuksa jest tworzona struktura katalogów na dysku lub dyskach zainstalowanych w komputerze. Standardowe katalogi tworzone podczas instalacji:

- **/** – katalog główny;
- **/bin** – są w nim przechowywane standardowe polecenia systemu Linux;
- **/dev** – są w nim przechowywane pliki reprezentujące punkty dostępu do urządzeń systemowych, np. napędów CD/DVD;
- **/etc** – są w nim przechowywane administracyjne pliki konfiguracyjne;
- **/home** – są w nim przechowywane katalogi domowe poszczególnych użytkowników;
- **/mnt** – zapewnia odpowiednią lokalizację do montowania urządzeń, takich jak zdalne systemy plików oraz nośniki wymienne;
- **/root** – katalog domowy administratora systemu;
- **/sbin** – są w nim przechowywane polecenia służące do administrowania systemem oraz uruchamiające procesy tzw. demonów (ang. *daemon*);
- **/tmp** – są w nim przechowywane pliki tymczasowe, wykorzystywane przez różne aplikacje;
- **/usr** – są w nim przechowywane pliki dokumentacji systemu oraz większość programów użytkownika;
- **/boot** – jest w nim przechowywane jądro systemu;
- **/var** – są w nim przechowywane często zmieniające się pliki, np. logi systemowe oraz katalogi danych różnych aplikacji, w szczególności takie jak serwer FTP (katalog **/var/ftp**) czy serwer WWW (katalog **/var/www**).

System plików w Linuksie możemy traktować jak pojedyncze drzewo. Jeżeli dołączymy do systemu plików jakiś dodatkowy nośnik danych, np. dysk CD/DVD lub pamięć flash, to system plików tego urządzenia zostanie doczepiony do systemu głównego. Miejsce doczepienia jest nazywane **punktem montowania** (ang. *mount point*). Jako punkt montowania można wybrać dowolny katalog w głównym systemie plików (jeżeli w katalogu są dane, to zostaną one przesłonięte przez zawartość nośnika). Zwykle do montowania nośników używa się katalogów **/mnt** i **/media**. Aby zamontować system plików z dyskietki, należy użyć polecenia:

```
sudo mount -t msdos /dev/fd0/mnt/floppy
```

Opcja **-t** w poleceniu oznacza, że zostanie podany typ systemu plików na nośniku. Najczęściej używane to:

- **msdos** – system FAT12 (dyskietka),
- **vfat** – system FAT,
- **ntfs** – system NTFS,
- **iso9660** – CD/DVD.

Pierwszy parametr oznacza identyfikator urządzenia, np.:

- `/dev/fd0` – pierwsza stacja dyskietek,
- `/dev/cdrom` – napęd CD/DVD,
- `/dev/hda1` – pierwsza partycja na dysku ATA (kontroler podstawowy);
- `/dev/sda1` – pierwsza partycja na dysku SCSI, SATA lub pamięci flash.

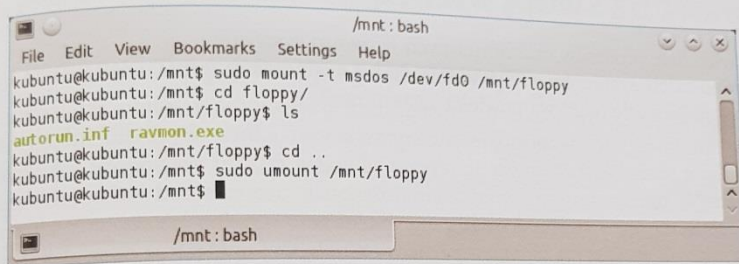
Drugi parametr określa punkt montowania. Aby odmontować system plików, należy użyć polecenia:

```
sudo umount /dev/fd0
```

Zamiast identyfikatora urządzenia można podać również punkt montowania, np.

```
sudo umount /mnt/floppy
```

Na rys. 13.15 pokazano montowanie i odmontowanie dyskietki.



```

/mnt: bash
File Edit View Bookmarks Settings Help
kubuntu@kubuntu:/mnt$ sudo mount -t msdos /dev/fd0 /mnt/floppy
kubuntu@kubuntu:/mnt$ cd floppy/
kubuntu@kubuntu:/mnt/floppy$ ls
autorun.inf  ravmon.exe
kubuntu@kubuntu:/mnt/floppy$ cd ..
kubuntu@kubuntu:/mnt$ sudo umount /mnt/floppy
kubuntu@kubuntu:/mnt$

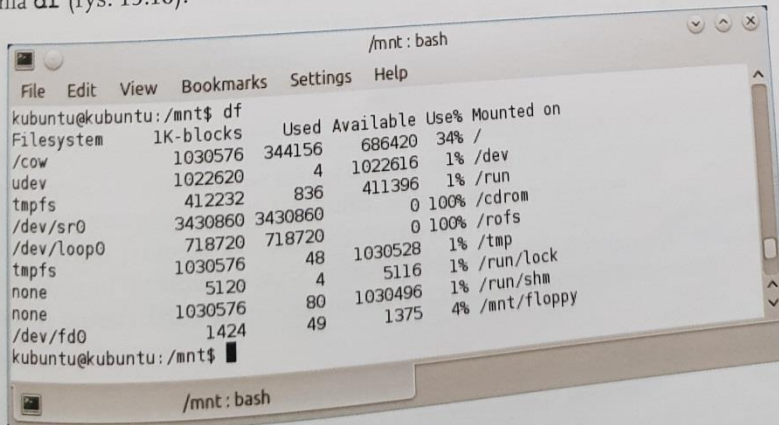
```

Rys. 13.15. Montowanie i odmontowanie dyskietki

! UWAGA

Nie można odmontować systemu plików aktualnie używanego. W powyższym przykładzie w pierwszej kolejności został zmieniony katalog na główny, a następnie możliwe było odmontowanie systemu plików.

Aby wyświetlić informacje o wszystkich zamontowanych systemach plików, należy użyć polecenia `df` (rys. 13.16).



```

/mnt: bash
File Edit View Bookmarks Settings Help
kubuntu@kubuntu:/mnt$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/cow             1030576    344156   686420   34% /
udev            1022620         4   1022616    1% /dev
tmpfs           412232         836   411396    1% /run
/dev/sr0        3430860  3430860         0 100% /cdrom
/dev/loop0      718720    718720         0 100% /rofs
tmpfs           1030576         48  1030528    1% /tmp
tmpfs           5120         4      5116    1% /run/lock
none            1030576         80  1030496    1% /run/shm
/dev/fd0        1424         49     1375    4% /mnt/floppy
kubuntu@kubuntu:/mnt$

```

Rys. 13.16. Lista zamontowanych systemów plików

PRZYKŁAD 13.5**Montowanie nośników wymiennych**

Systemy plików z nośników wymiennych są dołączane do głównego systemu plików Linuksa w punktach montowania. Standardowym punktem montowania dla nośników wymiennych jest katalog `/mnt` lub `/media`.

1. Utwórz w katalogu `/mnt` podkatalogi `cdrom`, `flash`, które posłużą jako punkty montowania nośników.
2. Zamontuj do katalogu dyskietkę, płytę z napędu CD/DVD, nośnik pamięci flash.
3. Sprawdź zawartość nośników wymiennych.
4. Wyświetl informacje o zamontowanych systemach plików.
5. Odmontuj nośniki wymienne.

 **SPRAWDŹ SWOJĄ WIEDZĘ**

1. Wyświetl listę procesów uruchomionych w twoim systemie.
2. Wymień zalety systemów wielozadaniowych i wielodostępnych.
3. Wyjaśnij, jaką rolę odgrywa powłoka systemowa.