

1. System operacyjny Linux

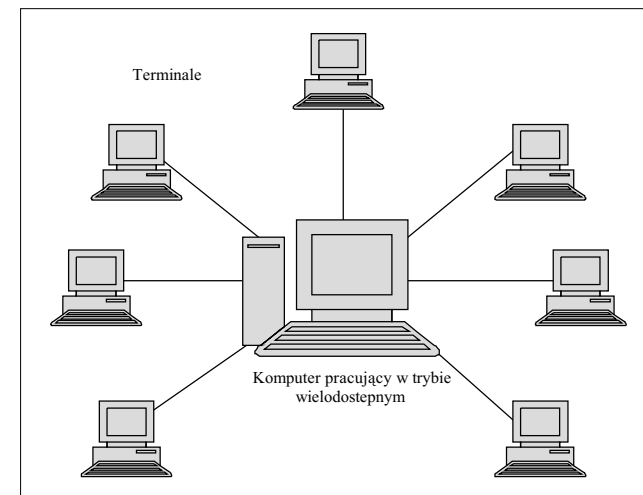
1.1. Historia systemu operacyjnego Linux

Podstawowe warunki i przyczyny powstania systemu operacyjnego Linux:

- wąskie i specjalistyczne grono użytkowników systemów uniksowych,
- szybki rozwój Internetu na początku lat dziewięćdziesiątych,
- powstanie procesora Intel 80386 umożliwiającego pracę wielozadaniową,
- napisanie przez Linusa Torvaldsa programów dla systemu operacyjnego Minix oraz jądra nowego systemu operacyjnego (opartego na Miniksie),
- ogłoszenie przez Torvaldsa na grupie dyskusyjnej comp.os.minix (sierpień 1991) prac nad bezpłatnym systemem dla komputerów PC,
- dostosowanie systemu Linux do norm POSIX pozwalające na przenoszenie oprogramowania pomiędzy Linuksem a komercyjnymi systemami uniksowymi,
- opracowanie Linuksa na podstawie licencji GNU GPL (*GNU General Public Licence*) fundacji FSF (*Free Software Foundation*),
- pojawienia się różnych *dystrybucji* Linuksa, (np. Slackware, RedHat, Debian, SuSE, Mandrake).

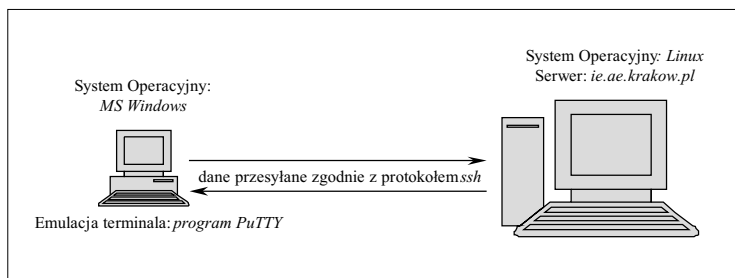
1.2. Praca zdalna z serwerem linuksowym

- *system operacyjny* (SO) to program lub zbiór programów służących do zarządzania pracą komputera,
- Linux jest systemem wielozadaniowym i wielodostępnym,
- *wielozadaniowość* to tryb pracy systemu operacyjnego, w którym użytkownik może uruchomić w tym samym czasie więcej niż jedno zadanie,
- *wielodostęp* to tryb pracy systemu operacyjnego, w którym w tym samym czasie więcej niż jeden użytkownik może pracować z systemem i korzystać z jego zasobów,
- równoczesna praca z systemem odbywa się za pomocą komputerów pełniących funkcje tzw. *terminali*,



Rys. 1. Schemat pracy z systemem wielodostępnym

- użytkownik, który chce pracować z systemem Linux, musi mieć do tego odpowiednie uprawnienia,
- weryfikacja uprawnień odbywa się poprzez sprawdzenie znajomości nazwy użytkownika (tzw. *login*) i hasła,
- użytkownik może korzystać wyłącznie z zasobów systemu, do których ma nadane prawa,
- kontrolę nad całym systemem sprawuje administrator, tzw. *root*,
- praca z serwerem odbywa się zazwyczaj w *trybie tekstowym*, jako tzw. *praca zdalna*, czyli praca na odległość za pomocą odpowiedniego programu emulującego terminal,
- praca zdalna w *trybie tekstowym* może być realizowana za pomocą protokołu *telnet* lub *ssh*,
- do pracy zdalnej z zastosowaniem protokołu *ssh* mogą być wykorzystywane różne programy, np. CRT, SSh32, PuTTY.



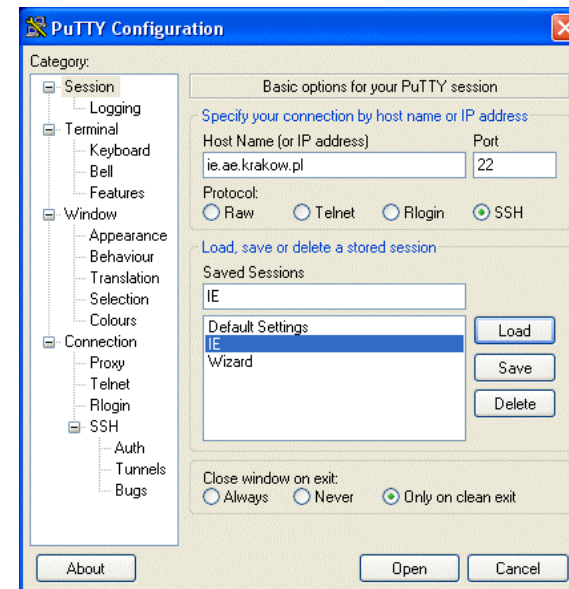
Rys. 2. Praca zdalna z serwerem linuxowym

1.3. Praca zdalna z wykorzystaniem programu PuTTY

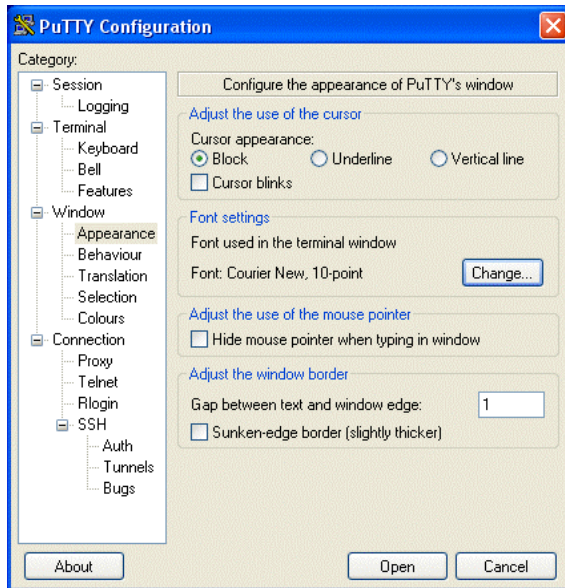
1. Uruchomienie programu PuTTY:



2. Podanie nazwy serwera (np. *ie.uek.krakow.pl*) w polu *Host Name (or IP address)* oraz wybranie protokołu *SSH*:

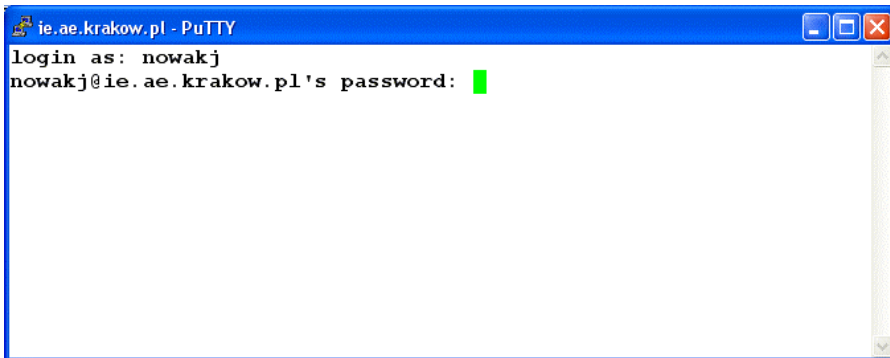


3. W razie konieczności skonfigurowanie ustawienia programu (np. wielkość i krój czcionki, kolor tła itp.) poprzez skorzystanie z kategorii *Window/Appearance*:



4. Naciśnięcie przycisku *Open*.

5. Podanie nazwy użytkownika (*login as:*) i hasła (*password:*):



Podczas pracy można wydawać polecenia i uruchamiać programy, np.:

- wyświetlić listę użytkowników zalogowanych w systemie:

```
$ finger
```

- zmienić swoje hasło:

```
$ passwd
```

po wydaniu tego polecenia użytkownik zostanie poproszony o podanie bieżącego (starego) hasła

```
Changing password for nowakj
```

```
Old password:
```

następnie zostanie poproszony o podanie nowego hasła i jego powtórne wprowadzenie (potwierdzenie).

```
New password:
```

W celu zakończenia pracy z serwerem należy wydać polecenie:

```
$ logout
```

2. System plików

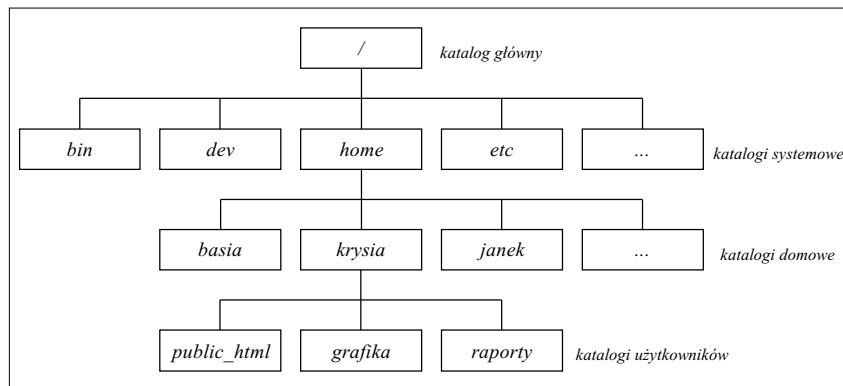
System plików jest to sposób zapisu logicznej struktury danych (tzn. plików oraz katalogów) na fizycznym nośniku, tzn.:

System plików = zbiór plików + sposób ich uporządkowania.

Pliki, jako podstawowe jednostki logiczne systemu plików:

- służą do przechowywania programów (rozkazów dla procesora) i danych (np. znaków, liczb, **wierszy** tekstu),
- są przechowywane na nośnikach danych (np. dyskach twardych, dyskach optycznych, pamięciach flash),
- mają strukturę ustaloną przez ich twórcę,
- posiadają swoją nazwę oraz atrybuty (np. rozmiar, datę i czas utworzenia, prawa dostępu),

- są zarządzane przez system operacyjny w sposób zgodny z wymaganiami i możliwościami systemu plików.
- Funkcje systemu operacyjnego w zakresie zarządzania systemem plików:
 - przydział miejsca na nośniku (dysku, dyskietce) dla systemu plików oraz ewidencja zajętych i wolnych obszarów nośnika,
 - określenie sposobu uporządkowania plików i organizowanie do nich dostępu zgodnie z wymogami i możliwościami systemu plików,
 - wykonywanie operacji na plikach, np. kopiowanie, przesuwanie, kasowanie, zmiana nazwy itp.



Rys. 3. Schemat struktury katalogów w Linuksie

W systemie plików Linuksa:

- struktura katalogów jest na ogół ściśle określona, tzn. katalog:
 - `/bin` - zawiera niezbędne do funkcjonowania systemu programy,
 - `/boot` - zawiera jądro systemu i pliki startowe,

- `/dev` - zawiera urządzenia (widziane, jako pliki, np. `/dev/hda1` - dysk twardy, `/dev/fd0` - stacja dyskietek),
- `/etc` - zawiera pliki konfiguracji systemu,
- `/home` - zawiera katalogi domowe użytkowników,
- `/lib` - zawiera m.in. biblioteki systemowe i ładowalne moduły jądra,
- `/lost+found` - zawiera odzyskane pliki,
- `/mnt` - zawiera zamontowane systemy plików innych urządzeń (np. dyskietki, CD-ROM-u),
- `/proc` - zawiera tzw. pliki statusu jądra, urządzeń i procesów,
- `/root` - jest to katalog domowy administratora,
- `/sbin` - zawiera programy wykorzystywane przez administratora do zarządzania i konfigurowania systemu,
- `/tmp` - zawiera pliki tymczasowe tworzone przez różne programy podczas ich pracy,
- `/usr` - zawiera programy, biblioteki i dokumenty dostępne dla użytkowników systemu,
- `/var` - zawiera m.in. przychodzącą i wychodzącą pocztę oraz logi systemowe.

- każdy użytkownik systemu posiada swój katalog, tzw. *katalog domowy*,
- użytkownik może wykonywać operacje na plikach znajdujących się w jego katalogu domowym, jego podkatalogach oraz w innych katalogach - o ile ich właściciele dadzą mu do tego prawo,
- każdy plik posiada swoją nazwę o długości do 255 znaków,

- nazwa pliku może składać się, m.in. z liter, cyfr, znaku podkreślenia, spacji, kropek,
- nazwa pliku rozpoczynająca się od kropki oznacza *plik ukryty* (np. „.tajne.txt”),
- w nazwach rozróżniane są małe i duże litery (np. nazwy plików „wiosna.txt” i „Wiosna.txt” są różne),
- w nazwach plików i katalogów stosuje się zazwyczaj tylko małe litery alfabetu łacińskiego,
- użytkownik może poruszać się po systemie katalogów - katalog, w którym się aktualnie znajduje to *katalog bieżący*,
- niektóre katalogi mają specjalne oznaczenia:
 - znak slash (/) symbolizuje *katalog główny*,
 - kropka (.) symbolizuje *katalog bieżący*,
 - dwie kropki (..) symbolizują *katalog nadrzędny*,
 - znak tyldy (~) symbolizuje *katalog domowy użytkownika*,
- położenie pliku w systemie plików określa *ścieżka dostępu*:

/home/krysia/grafika/rysunek.jpeg

- *względna*, czyli taka która określa położenie pliku względem bieżącego katalogu, np.

./grafika/rysunek.jpeg

../grafika/rysunek.jpeg

~/grafika/rysunek.jpeg

- *wieloznaczne nazwy* (tzw. maski nazwy lub wzorce nazwy) oznaczają grupę plików lub katalogów i są tworzone za pomocą znaków uogólniających, takich jak:
 - * - oznacza dowolny ciąg znaków (w tym także ciąg pusty),
 - ? - oznacza dowolny jeden znak,
 - [znaki] lub [znak1-znak2] - oznacza jeden ze znaków wymienionych na liście lub znak z pomiędzy podanego zakresu.

Tabela 1. Przykładowe wzorce nazw plików i katalogów

Wzorzec nazwy	Opis	Przykład
a*	pliki i katalogi, których nazwa zaczyna się na literę „a”	a ala a123
a???	pliki i katalogi, których nazwa ma długość czterech znaków i rozpoczyna się na literę „a”	a-la a345 abba
t[aeioy]k	pliki i katalogi, których nazwa ma długość trzech znaków, pierwszy znak to „t”, trzeci to „k” a drugi to „a”, „e”, „i”, „o” lub „y”	tak tik tok
rysunek[1-7].???	pliki i katalogi, których nazwa zaczyna się od słowa „rysunek”, po którym następuje jedna z cyfr od 1 do 7, mające trzyznakowe rozszerzenie	rysunek1.jpg rysunek3.bmp rysunek7.gif
*	wszystkie pliki i katalogi	test.doc Pliki! DOKUMENTY
*.mp?	pliki i katalogi, których nazwa składa się z dowolnego ciągu znaków, mające trzyznakowe rozszerzenie zaczynające się od „mp”	song.mp3 Moj:Film.mpg super video!.mpg

3. Operacje na plikach i katalogach

3.1. Składnia poleceń liniowych

Praca w trybie tekstowym polega na wydawaniu systemowi odpowiednich poleceń z poziomu tzw. *wiersza poleceń* (ang. *command line*).

Składania polecenia w systemie Linux:

```
polecenie [-opcje ] [ argumenty ]
```

Opcje polecenia to zazwyczaj pojedyncze litery poprzedzone znakiem minus (-). Opcje można podawać pojedynczo lub grupowo, np.

```
$ ls -a -l -t
```

```
$ ls -alt
```

```
$ ls -al -t
```

Niektóre opcje mają także postać pełnych słów, wtedy każdą z nich należy poprzedzić podwójnym znakiem minus (--), np.

```
$ ls --all --format=long
```

polecenie to jest równoważne poleceniu ls -al

Argumenty polecenia to ciągi znaków (np. nazwy plików i katalogów, nazwy użytkowników, wartości liczbowe) przekazywane do polecenia, np.

```
$ cp stary.plik nowy.plik
```

argumentami są tutaj nazwy plików

```
$ finger nowakj
```

argumentem jest tutaj nazwa użytkownika

```
$ ls /home
```

argumentem jest tutaj ścieżka dostępu

Polecenia można wydawać w jednym wierszu, ale należy je w tym celu oddzielić od siebie znakiem średnika (;), np.

```
$ who; pwd; ls /home -al
```

3.2. Wyszukiwanie informacji o poleceniach

- *apropos* (składnia: *apropos szukany_ciąg_znaków*) - program przeszukuje opisy poleceń, np.

```
$ apropos „copy files”
```

wyszukaj polecenia, w których opisie znajduje się ciąg znaków „copy files”

```
$ apropos password | less
```

wyszukaj polecenia, w których opisie znajduje się słowo „password”

- *man* (składnia: *man nazwa_programu*), *info* (składnia: *info nazwa_programu*), opcja *--help* (składnia: *nazwa_programu --help*) - wyświetla opis składni i opcji danego polecenia, np.

```
$ man mv
```

wyświetl opis polecenia mv

```
$ info cp
```

wyświetl opis polecenia cp

```
$ rm --help |less
```

wyświetl opis polecenia rm

Zakończenie wyświetlania opisu odbywa się po naciśnięciu klawisza [Q].

3.3. Podstawowe polecenia systemu Linux

- **pwd** - wyświetlanie nazwy bieżącego katalogu, np.

```
$ pwd
```

- **ls** - wyświetlenie zawartości katalogu, np.

```
$ ls
```

wyświetl zawartość bieżącego katalogu (bez plików ukrytych)

```
$ ls /bin
```

wyświetl zawartość katalogu /bin

```
$ ls *.c
```

```
$ ls -a
```

```
$ ls -l
```

```
$ ls -tl
```

- **cd** - zmiana bieżącego katalogu, np.

```
$ cd grafika
```

przejdź do katalogu „grafika” będącego w bieżącym katalogu

```
$ cd ..
```

```
$ cd ../../listy
```

```
$ cd /
```

```
$ cd /home/jacek
```

```
$ cd
```

```
$ cd ~
```

- **mkdir** - utworzenie katalogu, np.

```
$ mkdir test
```

utwórz katalog „test” w katalogu bieżącym

```
$ mkdir grafika/zdjecia
```

- **tree** - wyświetlenie struktury katalogów i plików, np.

```
$ tree
```

- **rmdir** - usunięcie pustego katalogu, np.

```
$ rmdir test
```

usuwany w ten sposób katalog „test” musi być pusty, a użytkownik musi mieć uprawnienia do jego usunięcia. Ponadto, nie można usunąć katalogu bieżącego.

- edytor **pico** - utworzenie plików tekstowych, np.

```
$ pico zaproszenie.txt
```

uruchomi edycję pliku zaproszenie.txt w edytorze pico:



Podstawowe polecenia edytora *pico* :

- [Ctrl]+[O] - zapisanie zmian (Write Out),
- [Ctrl]+[R] - odczytanie pliku (Read File),
- [Ctrl]+[X] - zamknięcie edytora (Exit).

- *find* - wyszukiwanie plików i katalogów zgodnie z zadaniem kryterium, np.

```
$ find / -name index.html
```

znajdź w katalogu głównym i jego podkatalogach wszystkie pliki o nazwie „index.html”. Należy pamiętać, że aby móc przeszukiwać zawartość katalogów, użytkownik musi mieć do tego prawo. W przeciwnym wypadku otrzyma komunikat o braku dostępu do danego katalogu („Permission denied”).

```
$ find . -name "*.txt"
```

```
$ find /home -user kowalski -name "*.html"
```

- *cat, more, less* - wyświetlanie plików tekstowych, np.

```
$ cat zaproszenie
```

```
$ more zaproszenie
```

```
$ less zaproszenie
```

- *rm* - usuwanie plików, np.

```
$ rm zaproszenie.txt
```

usuń plik zaproszenie.txt

```
$ rm w*
```

```
$ rm ~/*
```

```
$ rm -rf moj katalog
```

- *cp* - kopiowanie plików, np.

```
$ cp list.txt zaproszenie.txt
```

kopiuj plik „list.txt” do pliku „zaproszenie.txt”

```
$ cp *.jpeg ~/grafika
```

```
$ cp list.txt ..
```

- *mv* - przenoszenie plików (zmiana nazwy):

```
$ mv list.txt zaproszenie.txt
```

zmień nazwę pliku „list.txt” na „zaproszenie.txt”

- klawisz [Tab] - uzupełnianie nazwy plików. Naciśnięcie klawisza [Tab] podczas wprowadzania nazwy pliku (jako argumentu polecenia) spowoduje uzupełnienie jego nazwy.

- klawisze kursora [↑] i [↓] - przechodzenie do wydanych wcześniej poleceń. Naciśnięcie klawiszy kursora pozwala przeglądać i wybrać wydane wcześniej polecenia.

- *history* - przeglądanie wydanych poleceń, np.

```
$ history
```

- *du* - wyświetlanie informacji o zajętości katalogów, np.

```
$ du
```

- *df* - wyświetlenie informacji o zajętości dysków, np.

```
$ df
```

- *quota* - wyświetlenie informacji o ograniczeniach użytkownika co do ilości miejsca i liczby utworzonych plików, np.

```
$ quota
```

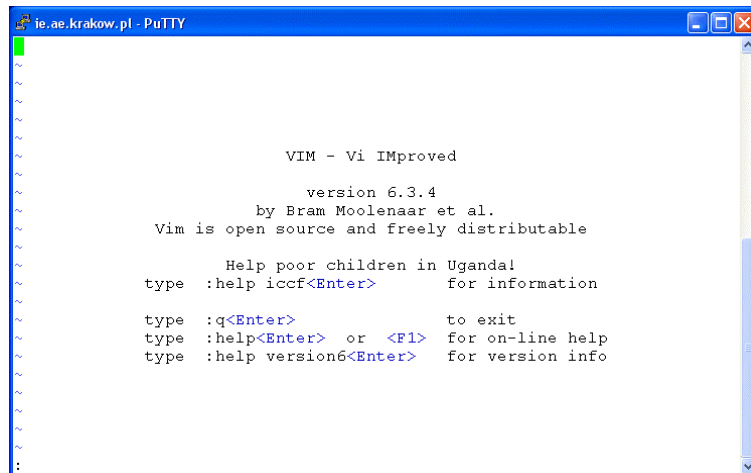

4. Edytor tekstów vi

Zalety edytora vi:

- popularność - standardowo dostępny edytor w systemach uniksowych,
- możliwość pracy z bardzo dużymi plikami tekstowymi,
- podświetlanie składni wielu różnych języków programowania oraz dodatkowe ułatwienia dla programistów (np. kontrola zamknięcia nawiasów, automatyczne wcięcia kodu programu, wsparcie dla narzędzi programistycznych, takich jak gcc, cvs, make),
- zaawansowane wyszukiwanie tekstu (interpretacja wyrażeń regularnych),
- łatwe pisanie makropoleceń.

Utworzenie (otworzenie) pliku w edytorze vi:

```
$ vi praca.txt
```



Rys. 4. Edytor tekstów vi

4.1. Tryby pracy edytora vi

Do najczęściej wykorzystywanych trybów edytora vi należy:

- normalny tryb wydawania poleceń (tzw. tryb normalny),
- tryb wprowadzania tekstu,
- tryb zastępowania tekstu,
- tryb wydawania poleceń z linii poleceń (tzw. tryb poleceń).

Normalny tryb wydawania poleceń (tryb normalny) jest to tryb, który służy do:

- poruszania się po dokumencie, tzn. przeglądanie tekstu oraz proste wyszukiwanie znaków w obrębie bieżącego wiersza (zob. Tabela 2),
- zaawansowane usuwanie fragmentów tekstu (zob. Tabela 4),
- kopiowanie fragmentów tekstu do bufora oraz wstawianie ich do dokumentu (zob. Tabela 8).

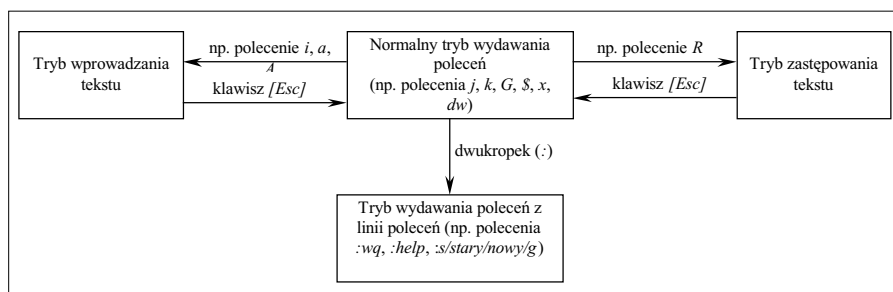
Tryb wprowadzania tekstu pozwala wprowadzać tekst do dokumentu oraz go usuwać. Przejście do trybu wprowadzania następuje po użyciu w trybie normalnym, jednego z takich poleceń, jak np. *i*, *I*, *a*, *A*, *o*, *O* (zob. Tabela 3). Powrót z trybu wprowadzania tekstu do trybu normalnego - klawisz [Esc].

Tryb zastępowania tekstu pozwala na nadpisywanie wprowadzonego tekstu nowym. Przejście do trybu zastępowania następuje po użyciu w normalnym trybie, polecenia *R* (zob. Tabela 5). Powrót z trybu zastępowania do trybu normalnego - klawisz [Esc].

Tryb wydawania poleceń z linii poleceń (tryb poleceń) pozwala na:

- zaawansowane wyszukiwanie (zob. Tabela 6) i zamianę znalezionego tekstu (zob. Tabela 7),
- operacje na plikach i wydawanie poleceń systemowych (zob. Tabela 10),
- zmianę sposobu wyświetlania tekstu (np. numerowanie wierszy tekstu) oraz konfigurowanie edytora (np. określenie zasad traktowania dużych i małych liter podczas wyszukiwania tekstu).

Przejdzie do trybu wydawania poleceń następuje po naciśnięciu w normalnym trybie dwukropka (:). Powrót z trybu wydawania poleceń do trybu normalnego następuje automatycznie po wykonaniu polecenia.



Rys. 5. Tryby pracy edytora vi

4.2. Podstawowe polecenia edytora vi

Tabela 2. Polecenia edytora vi - nawigacja po dokumencie

j	przesunięcie kursora o jeden wiersz w dół
k	przesunięcie kursora o jeden wiersz w górę
h	przesunięcie kursora o jeden znak w lewo
l	przesunięcie kursora o jeden znak w prawo
L	przesunięcie kursora na dół ekranu
H	przesunięcie kursora na górę ekranu
M	przesunięcie kursora na środek ekranu
CTRL+f	przesunięcie kursora o jeden ekran w dół
CTRL+b	przesunięcie kursora o jeden ekran w górę
CTRL+d	przesunięcie kursora o pół ekranu w dół
CTRL+u	przesunięcie kursora o pół ekranu w górę
G	przejdź na początek ostatniego wiersza
nG	powoduje przesunięcie kursora do n-tego wiersza
\$	przesunięcie kursora na koniec bieżącego wiersza
^	przesunięcie kursora na początek bieżącego wiersza
fx	przesunięcie kursora do najbliższego znaku „x” znajdującego się po prawej stronie kursora w bieżącym wierszu
Fx	przesunięcie kursora do najbliższego znaku „x” znajdującego się po lewej stronie kursora w bieżącym wierszu
;	powoduje powtórzenie ostatniego polecenia f lub F
w	przesunięcie kursora o jeden wyraz do przodu
nw	przesunięcie kursora o n wyrazów do przodu
b	przesunięcie kursora o jeden wyraz do tyłu
nb	przesunięcie kursora o n wyrazów do tyłu
e	przesunięcie kursora na koniec bieżącego wyrazu
(przesunięcie kursora o jedno zdanie do tyłu
)	przesunięcie kursora o jedno zdanie do przodu

Tabela 3. Polecenia edytora vi - wprowadzanie tekstu

i	rozpoczęcie wprowadzania tekstu od bieżącej pozycji kursora
a	rozpoczęcie wprowadzania tekstu od znaku znajdującego się za kursorem
I	rozpoczęcie wprowadzania tekstu na początku linii w której znajduje się kursor
A	rozpoczęcie wprowadzania tekstu na końcu linii w której znajduje się kursor
O	wstawienie nowego wiersza powyżej kursora i rozpoczęcie wprowadzania tekstu w nowym wierszu
o	wstawienie nowego wiersza poniżej kursora i rozpoczęcie wprowadzania tekstu w nowym wierszu

Tabela 4. Polecenia edytora vi - usuwanie tekstu

x	usunięcie znaku znajdującego się pod kursorem
X	usunięcie znaku znajdującego się przed kursorem
de	usunięcie znaków znajdujących się od miejsca wskazywanego przez kursor do końca bieżącego wyrazu
dw	usunięcie znaków od miejsca wskazywanego przez kursor do początku następnego wyrazu
d^	usunięcie znaków od bieżącej pozycji kursora do początku danego wiersza
d\$	usunięcie znaków od bieżącej pozycji kursora do końca wiersza
dG	usunięcie tekstu od pozycji kursora do końca pliku
dnh	usunięcie <i>n</i> znaków poprzedzających
dnl	usunięcie <i>n</i> kolejnych znaków
D	usunięcie znaków do końca bieżącego wiersza
J	połączenie wiersza poniżej kursora z wierszem bieżącym
dd	usunięcie bieżącego wiersza
dnd	usunięcie bieżącego wiersza i <i>n-1</i> kolejnych wierszy
u	anulowanie wprowadzonych zmian (m.in. usuniętego tekstu)

Tabela 5. Polecenia edytora vi służące do zastępowania tekstu

rx	zastąpienie znaku znajdującego się pod kursorem znakiem „x”; po dokonaniu zamiany edytor pozostaje w trybie wydawania poleceń
cw	usunięcie znaków do końca bieżącego słowa i przejście do trybu wstawiania tekstu
C	usunięcie znaków do końca wiersza i przejście do trybu wprowadzania
R	włączenie trybu zastępowania

Tabela 6. Polecenia edytora vi - wyszukiwanie tekstu

/abc	wyszukanie ciągu „abc” w dół ekranu
?abc	wyszukanie ciągu „abc” w górę ekranu
n	powtórzenie ostatniego polecenia wyszukiwania ciągu znaków w dół ekranu
N	powtórzenie ostatniego polecenia wyszukiwania ciągu znaków w górę ekranu
:set ic	włączenie (:set ic) i wyłączenie (:set noic - tryb domyślny) trybu
:set noic	ignorowania wielkości znaków podczas wyszukiwania tekstu

Tabela 7. Polecenia edytora vi - zamiana tekstu

:s/stary/nowy/	znalezienie w bieżącym wierszu pierwszego wystąpienia tekstu <i>stary</i> i zastąpienie go tekstem <i>nowy</i>
:s/stary/nowy/g	znalezienie w bieżącym wierszu wszystkich wystąpień tekstu <i>stary</i> i zastąpienie ich tekstem <i>nowy</i>
:n,m s/stary/nowy/g	przeszukanie fragmentu tekstu od wiersza o numerze <i>n</i> do wiersza o numerze <i>m</i> i znalezienie w tym obszarze wszystkich wystąpień tekstu <i>stary</i> i zastąpienie ich tekstem <i>nowy</i>
:1,\$ s/stary/nowy/g	globalne zastępowanie w całym pliku
:1,\$ s/stary/nowy/gc	globalne zastępowanie w całym pliku - każda operacja zamiany wymaga akceptacji przez użytkownika

Bufor edytora

Edytor tekstu *vi* umożliwia umieszczenie (skopiowanie lub przeniesienie) fragmentu tekstu do bufora (schowka), a następnie wklejenie go w dowolne miejsce dokumentu.

W buforze umieszczany jest fragment tekstu, który został:

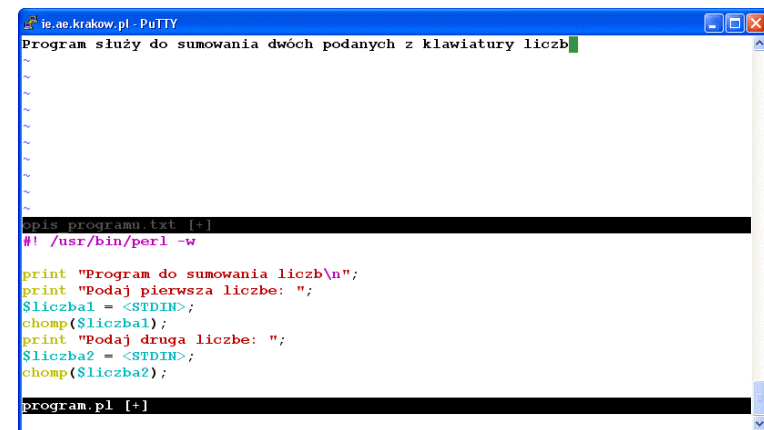
- skopiowany (zob. Tabela 8),
- usunięty (zob. Tabela 4).

Tekst z bufora może być wielokrotnie wstawiany (wklejany) do dokumentu.

Tabela 8. Polecenia edytora vi - kopiowanie tekstu

yw	skopiowanie tekstu od bieżącej pozycji kursora do końca wyrazu
yb	skopiowanie tekstu od początku wyrazu do bieżącej pozycji kursora
y\$	skopiowanie tekstu od bieżącej pozycji kursora do końca wiersza
Y	skopiowanie do bufora bieżącego wiersza
yy	Działanie jak „Y”
p	Wklejenie zawartości bufora za znakiem wskazywanym przez kursor
P	Wklejenie zawartości bufora przed znakiem wskazywanym przez kursor

Praca z kilkoma oknami



Rys. 6. Edytor tekstów vi z otwartymi dwoma oknami

Tabela 9. Polecenia edytora vi - praca z wieloma oknami

:new	otworzenie nowego okna edytora
:split lub [Ctrl+W]	podział aktualnego okna na dwie części
[Ctrl+W] j	przejdźcie do okna poniżej
[Ctrl+W] k	przejdźcie do okna powyżej
[Ctrl+W] t	przejdźcie do górnego okna
[Ctrl+W] b	przejdźcie do dolnego okna
:help	wyświetlenie w nowym oknie pomocy

Koniec pracy z edytorem

Do zapisania zmian w dokumencie i zakończenia pracy z edytorem vi służy polecenie :wq. Jeżeli użytkownik nie chce zapisywać wprowadzonych zmian musi wydać polecenie :q!. Polecenia związane z kończeniem pracy w edytorze oraz z wykonywaniem operacji na plikach zawiera poniższa tabela.

Tabela 10. Polecenia edytora vi - operacje na plikach, poleceniach systemu, kończenie pracy

:q	zakończenie pracy edytora
:q!	zakończenie pracy edytora bez zapisania zmian
:wq	zakończenie pracy edytora wraz z zapisaniem zmian
:r nazwa pliku	wstawienie zawartości pliku w bieżącym wierszu
:! polecenie	wykonanie polecenie powłoki
:r! polecenie	wstawienie wyniku działania polecenia w bieżącym wierszu

4.3. Konfigurowane edytora vi

Ustawienia edytora zapisywane są w pliku .vimrc, który musi znajdować się w katalogu domowym użytkownika.

Najczęściej zapisywane ustawienia konfiguracji:

```
syntax on
```

włączenie podświetlania składni edytowanego kodu programu

```
set number
```

numerowanie wyświetlanych wierszy dokumentu

```
set nonumber
```

wyłączenie numerowania wyświetlanych wierszy

```
set ts=4
```

ustalenie pozycji tabulacji na 4 znaki

```
set textwidth=80
```

ustawienie długości wyświetlanego wiersza na 80 znaków

```
set autoindent
```

włączenie auto-wcięcia

```
set noautoindent
```

wyłączenie auto-wcięcia

```
set incsearch
```

automatyczne wyszukiwanie ciągu znaków podczas korzystania z polecenia „/”

```
set ruler
```

wyświetlenie współrzędnych kursora (wiersz, kolumna)

```
set showmode
```

wyświetlenie informacji o trybie edytora

```
set showcmd
```

wyświetlenie wydawanego polecenia

```
set backup
```

tworzenie kopii zapasowych plików

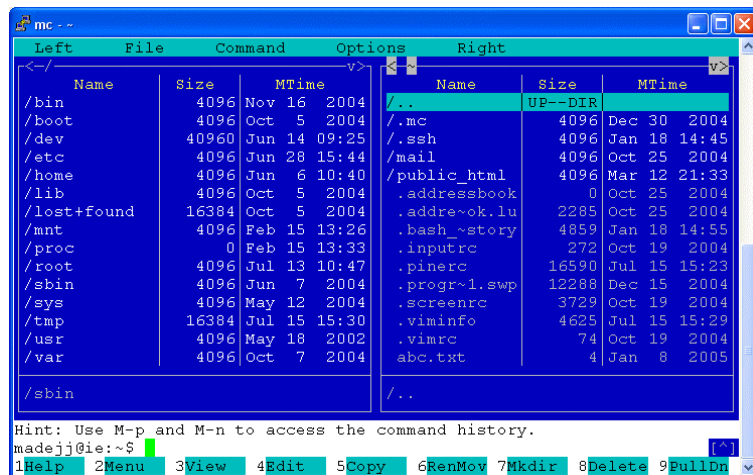
```
set backupdir=~ /backup/
```

określenie katalogu, w którym tworzone będą kopie zapasowe (w tym przypadku jest to ~/backup/)

Powyższe polecenia można wydać w trybie poleceń.

5. Program Midnight Commader

Midnight Commander jest programem do zarządzania plikami i katalogami.



Rys. 7. Midnight Commander

Uruchomienie programu Midnight Commander:

```
$ mc
```

Podstawowe skróty klawiaturowe i klawisze funkcyjne programu *mc*:

- [F1], [F2], [F3], [F4], [F5], [F6], [F7], [F8], [F9], [F10],
- [Tab], [Alt]+[?], [Alt]+[O], [Ctrl]+[O], [Ctrl]+[R], [Ctrl]+[S], [Ctrl]+[U],
- [Alt]+[+], [Alt]+[-], [Alt]+[*].

Szczegółowy opis możliwości programu i kombinacji klawiszy jest dostępny w manualu (polecenie *man mc*).