

Politechnika Łódzka
Wydział Elektrotechniki Elektroniki Informatyki i Automatyki
Instytut Informatyki Stosowanej

Laboratorium z przedmiotu
Systemy Operacyjne 1

Moduł Linux: ćwiczenie nr 1

Wprowadzenie do systemu Linux, konsola systemowa oraz obsługa katalogów i plików.

Spis treści

1.	Wprowadzenie do systemu Linux.....	3
1.1.	Historia	3
1.2.	Cechy systemu Linux	3
2.	Konsola systemowa.....	4
2.1.	Logowanie się do systemu	4
2.2.	Powłoka systemowa	4
2.3.	System pomocy	5
2.4.	Wiele sesji na komputerze lokalnym.....	6
2.5.	Szybkie wpisywanie poleceń. Klawisz tab	6
2.6.	Zadania	7
3.	Pliki i katalogi.....	7
4.	Polecenia do obsługi plików i katalogów	8
4.1.	Poruszanie się po katalogach	8
4.2.	Znaki specjalne.....	9
4.3.	Polecenia dotyczące katalogów.....	10
4.3.1.1.	Listowanie zawartości katalogów – ls.....	10
4.3.2.	Zmiana katalogu - cd	10
4.3.3.	Tworzenie katalogów - mkdir.....	11
4.3.4.	Usuwanie katalogów – rmdir.....	11
4.3.5.	Przenoszenie katalogów, zmiana nazwy katalogu - mv.....	11
4.3.6.	Gdzie jestem? - pwd	11
4.4.	Polecenia dotyczące plików	12
4.4.1.	Kopiowanie plików - cp.....	12
4.4.2.	Przenoszenie plików, zmiana nazwy - mv	12
4.4.3.	Usuwanie plików - rm.....	12
4.4.4.	Tworzenie powiązań symbolicznych - ln.....	13
4.4.5.	Tworzenie pustego pliku/aktualizacja daty - touch.....	13
4.4.6.	Tworzenie i edycja pliku.....	13
4.4.7.	Wyszukiwanie plików - find	13
4.5.	Zadania	13

1. Wprowadzenie do systemu Linux

1.1. Historia

System operacyjny Linux powstał pod koniec 1991 roku. Twórcą tego systemu był Linus Torvalds z Uniwersytetu Helsińskiego, który stworzył go w oparciu o niewielką implementację systemu Unix o nazwie MINIX. Linux po raz pierwszy został opublikowany w sierpniu 1991r w Internecie.

Udostępniona wersja 0.01 nie spotkała się jednak z większym zainteresowaniem. Dopiero po dodaniu do dystrybucji kompilatora **gcc**, interpretera poleceń **bash**, zapewnieniu zgodności ze standardem POSIX oraz wprowadzeniu mechanizmu dynamicznej wymiany pamięci Linux 0.12 wydany w styczniu 1992r zainteresował szersze grono ludzi. Przyczyniło się to do szybszego rozwoju tego systemu i sprawiło, że bardzo szybko zyskał on popularność. W roku 1994 wprowadzono do systemu Linux mechanizmy pozwalające na pracę w sieci. Od 1995 Linux mógł pracować z procesorami: Intel, Alpha Digital i Sun Sparc. Dzięki zgodności z POSIXem szybko zaadaptowano systemem graficzny X Window System (X11). W 1998 ruszyły prace nad projektem KDE, który miał na celu stworzenie pierwszego zintegrowanego środowiska graficznego dla systemu Linux.

Obecnie Linux jest najpopularniejszym systemem operacyjnym wywodzącym się z Uniksa. Powstał dla niego ogromny zbiór oprogramowania, w większości wolnego lub o otwartym kodzie. Istnieje kilka dużych środowisk graficznych, pakietów biurowych oraz niezliczona ilość narzędzi sieciowych. Prawdopodobnie jest to system, który pozwala na uruchomienie największej ilości darmowego oprogramowania. Ponadto może pracować w kilkunastu architekturach sprzętowych oraz obsługuje większą ilość urządzeń peryferyjnych niż jakikolwiek inny system operacyjny.

1.2. Cechy systemu Linux

System Linux posiada następujące cechy:

- wielozadaniowość i wielodostęp – wielu użytkowników w tym samym czasie może wykonywać kilka zadań na tym samym komputerze
- pamięć wirtualną – Linux może używać części dysku twardego jako pamięci wirtualnej, co zwiększa wydajność systemu poprzez umieszczanie aktywnych procesów w pamięci RAM, a tych rzadziej używanych lub nieaktywnych na dysku
- wbudowaną obsługę sieci – poprzez modem lub kartę sieciową Linux może porozumiewać się z innymi komputerami wykorzystując protokoły TCP/IP, NFS, NIS i inne
- współużytkowanie bibliotek – programy w systemie Linux nie przechowują własnych kopii standardowych bibliotek tylko wspólnie współużytkują zestaw procedur, do których mogą odwoływać się w trakcie działania. Dzięki temu otrzymujemy więcej przestrzeni dyskowej
- kompatybilność z normą POSIX – dzięki czemu Linux obsługuje wiele standardów ustalonych dla systemów Unix
- X Window – jest to środowisko graficzne
- otwarty kod źródłowy – dzięki czemu możemy aktywnie uczestniczyć w rozwoju systemu
- darmowe oprogramowanie – dostajemy wraz z systemem na płytach CD, poza tym istnieje wiele stron internetowych, na których możemy znaleźć ciekawe oprogramowanie.

2. Konsola systemowa

2.1. Logowanie się do systemu

Systemy operacyjne z rodziny Linux mogą równocześnie obsługiwać wielu użytkowników. Muszą oni posiadać konto na danym komputerze pracującym pod kontrolą tego systemu. Konto na danym komputerze może założyć administrator danego systemu. Ustala on ponadto przynależności do grup, dostęp do odpowiednich zasobów w systemie, itp. Po uruchomieniu systemu na ekranie pojawia się monit z prośbą o wprowadzenie nazwy użytkownika i odpowiedniego hasła. Po zakończeniu sesji użytkownik powinien się z systemu wylogować. Służy do tego polecenie **exit**.

2.2. Powłoka systemowa

Powłoka systemowa (*ang. shell*) jest to interfejs użytkownika, który pozwala na wprowadzanie komend oraz ich interpretację. Powłoka umożliwia również komunikację z niższymi warstwami systemu Linux. Powłoka pełni w systemie Linux funkcję zbliżoną do interpretera poleceń `command.com`, znanego z systemu DOS oraz `cmd.exe` z systemu Windows 10. W systemie Linux dostępnych jest jednak wiele różnych powłok systemowych. Powłoki tzw. pierwotne występujące w systemach Unix, a dostępne w systemie Linux to **sh** (shell Bourne'a), **ksh** (shell Koma), **csh** (C Shell - o składni zbliżonej do składni języka C). Powłoki te są dość niewygodne w użyciu i obecnie się ich praktycznie nie stosuje. Do nowszych powłok wzbogaconych o wiele nowoczesnych rozwiązań, ułatwiających pracę z systemem zaliczamy **bash** (Bourne Again Shell - wypierający sh), **tcsh** (będący znacznym rozszerzeniem csh) czy **zsh** (Z Shell - rozszerzona wersja ksh). Powłoki te pozwalają na edycję wprowadzonego już polecenia, zachowują historię wydanych poleceń i pozwalają na jej przeglądanie, sprawdzają poprawność składni polecenia oraz uzupełniają ją o brakujące elementy. Użytkownik systemu Linux może w każdej chwili zmienić powłokę na dowolnie wybraną, o ile jest ona zainstalowana w systemie. Większość nowych dystrybucji jest skonfigurowana z powłoką **bash**. Typowy znak zachęty ma postać:

```
[bieżący_użytkownik@nazwa_serwera katalog_bieżącego_użytkownika]$
```

Przykład:

```
[user@localhost/etc]$
```

Z przykładu można odczytać, iż zalogowany jest użytkownik o nazwie *user* na serwerze *localhost*. Aktualnie znajduje się on w katalogu */etc*. Znak dolara (\$) oznacza, że użytkownik nie jest administratorem systemu. Znakiem zachęty dla konta administratora jest *hash* (#).

Standardowo, polecenia systemowe są zbudowane z nazwy polecenia, za którą następują opcje, a dopiero za nimi argumenty:

```
polecenie opcje argumenty
```

Większość poleceń systemu Linux znajduje się w katalogu `/bin` oraz `/usr/bin`. Podstawowe cechy poleceń systemu Linux to:

- znaczna ilość dostępnych opcji dla większości poleceń. Przykładowo polecenie `ls`, służące do wyświetlania zawartości katalogu, ma przeszło 70 dostępnych opcji.
- większość poleceń wyświetla bardzo skąpe informacje wynikowe. Często więc, brak odpowiedzi po wykonaniu polecenia należy traktować jako jego poprawne wykonanie. Przykładowo polecenie `ls` zastosowane do pustego katalogu nie wyświetli żadnego komunikatu.

Opcje poleceń poprzedza się myślnikiem i muszą one być oddzielone spacją od polecenia oraz innych opcji. Powłoka systemowa pozwala jednak na łączenie kilku opcji. Stosuje się wówczas zapis, w który wprowadza się jeden myślnik, a następnie podaje się opcje. Poniższe przykłady pokazują różne sposoby stosowania opcji:

Wydanie polecenia bez opcji

```
[user@ localhost /tmp]$ ls  
plik1 plik2 plik3
```

Wydanie polecenia z opcją

```
[user@ localhost /tmp]$ ls -l  
total  
1  
  
-rw-r--r-- 1 student users 59 May 25 23:16 tekst1
```

Wydanie polecenia z wieloma opcjami

```
[user@ localhost /tmp]$ ls -la total 3  
drwxrwxrwt 2 root root 1024 May 25 23:16 .  
drwxr-xr-x 17 root root 1024 May 25 20:14 ..  
..  
  
-rw-r--r-- 1 student users 59 May 25 23:16 testowyl
```

2.3. System pomocy

System Linux posiada rozbudowany i bardzo przydatny system pomocy. Pierwszym źródłem pomocy są tzw. strony *man* (*ang. manual*). Zawierają one definicje i objaśnienia poleceń systemu wraz z opisem opcjonalnych parametrów dla specjalnych funkcji poleceń. Aby wywołać stronę *man* dla wybranego polecenia, należy wydać komendę:

```
man polecenie
```

np.:

Spowoduje to wyświetlenie opisu danego polecenia. Jeśli chcemy wyszukać potrzebną informację, należy nacisnąć klawisz /, wpisać szukaną frazę i nacisnąć ENTER. Jeśli chcielibyśmy ponowić wyszukiwanie danej frazy, wystarczy nacisnąć klawisz / i ENTER. Aby zakończyć przeglądanie manuala, należy nacisnąć klawisz *q*.

2.4. Wiele sesji na komputerze lokalnym

System Linux posiada siedem wirtualnych konsol, do których mamy dostęp za pomocą kombinacji klawiszy ALT+Fx. W zależności od konfiguracji ilość konsol może się zmieniać. W ubuntu Linux domyślnie ALT+F1 do F6 przełącza na kolejne konsole tekstowe. Konsole od 7-mej do 11-tej zarezerwowane są dla środowiska graficznego. Na konsoli 12-tej można śledzić na bieżąco log systemowy.

Na każdej z wirtualnych konsol można zalogować się jako inny użytkownik. Takie rozwiązanie pozwala na uruchomienie na każdej z konsol jakiegoś zadania.

Należy zwrócić uwagę, iż w przypadku maszyn wirtualnych, często modyfikowane jest działanie klawiszy funkcyjnych F1-F12, przez co nie zawsze przełączanie pomiędzy konsolami odbywa się według przedstawionego powyżej opisu.

2.5. Szybkie wpisywanie poleceń. Klawisz tab

Nowoczesne powłoki w systemie Linux (np. bash) mają możliwość znacznie szybszego i wygodniejszego wprowadzania poleceń, czy też ścieżek dostępu.

Podobne odpowiedzi będą wyświetlane w trakcie poruszania się po strukturze katalogów. Aby z tej możliwości skorzystać, należy w trakcie wprowadzania polecenie lub ścieżki dostępu nacisnąć klawisz tabulacji.

Działanie klawisza tab:

1. Wprowadzając jakieś polecenie możemy je dokończyć (po wpisaniu kilku początkowych liter jego nazwy) naciskając klawisz tab. Wpisane litery są traktowane jako wzorzec polecenia.
2. Jeśli wpisanemu wzorcowi odpowiada kilka poleceń, polecenie nie zostanie uzupełnione. Jednak po ponownym wciśnięciu tab (powtórny) wszystkie pasujące do wzorca polecenia zostaną wypisane na ekranie. Jeśli liczba dostępnych poleceń jest duża, system wyświetli informację o ilości poleceń odpowiadających wzorcowi oraz pytanie czy tę listę wyświetlać.
3. Użytkownik musi zawęzić liczbę dostępnych poleceń wpisując dodatkowe litery, aż system rozpozna, o jakie polecenie chodzi. Jeśli danemu wzorcowi odpowiada tylko jedno polecenie, zostanie on automatycznie dokończony.
4. Podobnie wygląda sprawa wpisywania ścieżek dostępu do katalogu. Użytkownik w trakcie wpisywania nazwy katalogu może nacisnąć klawisz tab, co spowoduje dokończenie jego nazwy lub też wyświetlenie nazw katalogów lub plików, które odpowiadają wpisanemu wzorcowi. Klawisz tab w tym przypadku dokańcza tylko nazwę katalogu lub pliku, a nie całą ścieżkę.

Przykład (Wciśnięte a niewyświetlone klawisze oznaczono: [tab] oraz [n]):

```
# m [tab][tab]
Display all 267 possibilities? (y or n) [n]
# mk[tab][tab]
mkafmmmap      mk_cmds      mke2fs      mkfs      mkfs.minix  mkindex  mknode
mkrfc2734      mktexmf      mkdep      mkfifo      mkfs.bfs    mkfs.msdo  mkinfodir  mkntfs
mkswap         mktexpk      mkfontdesc  mkfs.cramfs  mkfs.ntfs   mkisofs   mkocp
mkbundle       mktemp       mktextfm    mkfontdir    mkfs.ext2   mkfs.vfat  mklost+found  mkofm
mkbundle2      mkxmlrpc     mkdosfs     mkfontscale  mkfs.ext3   mkhtmlindex  mkmanifest
mktexfmt       mktexlsr # mkd[tab][tab]
mkdep          mkdir        mkdirhier   mkdosfs #
mkd[tab]ir
```

2.6. Zadania

1. Uruchom system Linux
2. Zapoznaj się z systemem wirtualnych konsoli. Ile konsol dostępnych jest w systemie Linux?
3. Po przełączeniu na konsolę tekstową, sprawdź w jakim katalogu się znajdujesz.
4. Wyświetl zawartość katalogu bieżącego. Zbadaj różnicę w działaniu poleceń *ls -l*, *ls -a* i *ls -la*. Czym różni się polecenie *ls* od polecenie *ls **?
5. Zapoznaj się z reakcją systemu na błędnie wprowadzone polecenie np. *ls-la*
6. Zapoznaj się z mechanizmem dokończania poleceń za pomocą klawisza **tab**
7. Korzystając z klawisza **tab** wyświetl wszystkie katalogi zaczynające się na *s*, które znajdują się w katalogu */etc*
8. Korzystając z klawisza **tab**, wyświetl wszystkie polecenia zaczynając się na literę *m*.
9. Zamknij system korzystając z polecenie *shutdown* lub *go* zrestartuj (polecenie *reboot*)
10. Sprawdź działanie kombinacji CTRL+ATL+DEL.

3. Pliki i katalogi

System plików można określić jako metody i struktury danych, które są używane przez system operacyjny w celu zapisania i zorganizowania plików na danym urządzeniu. System Linux umożliwia obsługę wielu różnych systemów plików, np. FAT16, FAT32 itp. Jednak dla samego Linuksa został stworzony system *ext*, a następnie *ext2*, *ext3* oraz *ext4*.

W systemie Linux pliki są umieszczone w katalogach. Katalogi z kolei są ze sobą hierarchicznie powiązane w jedną strukturę plików. Nazwa pliku może składać się z liter, cyfr oraz niektórych znaków specjalnych, np. znaku podkreślenia. Jej długość może wynosić do 256 znaków. Należy pamiętać, że Linux rozróżnia duże i małe litery.

Kropka nie posiada specjalnego znaczenia, ponieważ Linux traktuje kropkę jak każdy inny znak. Pliki, których nazwy rozpoczynają się kropką są "ukryte". Traktowane są jak wszystkie inne pliki, z wyjątkiem tego, iż nie wyświetli ich polecenie *ls*, chyba że użytkownik zastosuje opcję *-a*. Przykładem pliku ukrytego jest plik **.profile**.

Wszystkie pliki mają jeden format fizyczny – ciąg bajtów zakończony znakiem *EOF* (*End-Of-File; Ctrl-D*). Ten hierarchiczny system plików jednolicie traktuje pliki zwykłe, katalogi oraz pliki opisujące urządzenia zewnętrzne.

Katalogi każdego użytkownika są w rzeczywistości połączone z katalogami pozostałych użytkowników. Są one ułożone w strukturę hierarchiczną drzewa, rozpoczynając się od katalogu głównego (**root**), będącego korzeniem. Wszystkie pozostałe katalogi wychodzą od tego pierwszego. Katalog główny / jest to korzeń drzewa. Niektóre katalogi są standardowymi katalogami zarezerwowanymi do użytku przez system. Katalogi systemowe zawierają pliki i programy używane do uruchomienia i utrzymywania systemu. Znaczenie poszczególnych katalogów systemowych jest następujące:

- /bin, /usr/bin** - zawierają większość komend systemowych i programy usługowe w wersji binarnej,
- /dev** - mieszczą się w nim pliki specjalne, reprezentujące urządzenia rzeczywiste (dyski twarde, elastyczne, drukarki) i pseudourządzenia (konsola systemowa, obszar swap, wirtualny terminal),
- /etc** - przechowywana jest w nim większość plików i programów umożliwiających konfigurację systemu,
- /tmp** - używany przez komendy, jak i przez użytkowników do przechowywania plików tymczasowych,
- /home** - katalog przeznaczony na katalogi domowe użytkowników systemu,
- /proc** - wirtualny system plików, zawierający informacje o systemie i uruchomionych procesach.
- /usr** - katalog zawierający zainstalowane programy (także posiada wewnątrz specyficzną strukturę)

4. Polecenia do obsługi plików i katalogów

4.1. Poruszanie się po katalogach

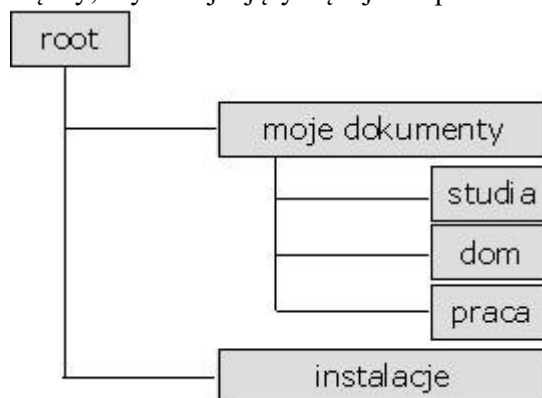
Przy rejestrowaniu nazwy użytkownika w systemie zostaje z nią związany katalog osobisty (domowy), oznaczany symbolem ~. Katalog ten staje się katalogiem bieżącym w chwili rozpoczynania przez użytkownika sesji przy terminalu. Każda nazwa pliku, którą podaje wówczas użytkownik, np. jako parametr polecenia, odnosi się do plików z katalogu bieżącego. Każdy plik można znaleźć rozpoczynając poszukiwanie od katalogu pierwotnego, tj. od korzenia drzewa katalogów.

W każdym katalogu (oprócz root) występuje element domyślny, stanowiący nazwę katalogu nadrzędnego, oznaczony dwiema kropkami (..), oraz inny element, stanowiący nazwę katalogu, w którym on sam się znajduje, oznaczony (.). Innymi słowy, każdy katalog wskazuje na samego siebie oraz na swój katalog nadrzędny.

Położenie pliku jest określone poprzez ciąg nazw prowadzących do niego katalogów. Ciąg ten nazywany jest ścieżką dostępu. Ścieżka dostępu do obiektu określa umiejscowienie obiektu w systemie plików. Ścieżka dostępu może być określana na dwa sposoby: jako ścieżka **bezwzględna** albo **względna**.

Bezwzględna ścieżka dostępu rozpoczyna się od folderu głównego. Pierwszym znakiem ścieżki bezwzględnej jest ukośnik (/). Folder główny jest pojedynczym folderem, od którego rozpoczyna się cała struktura systemu plików.

Względna ścieżka dostępu przedstawia lokalizację pliku lub folderu względem folderu bieżącego. Aby z folderu bieżącego przejść niżej w strukturze drzewa, nie trzeba podawać pełnej ścieżki dostępu. Wystarczy wpisać ścieżkę rozpoczynając od nazwy następnego folderu. Jeśli ścieżka dostępu nie rozpoczyna się ukośnikiem, jest ścieżką względną. Przy określaniu względnych ścieżek dostępu, warto pamiętać o dwóch symbolach specjalnych czyli o (.) reprezentującym folder bieżący oraz o (..) reprezentującym folder nadrzędny, czyli znajdujący się o jeden poziom wyżej w hierarchii.



Przykładowa struktura katalogów

Jeśli katalogiem bieżącym jest katalog *praca* (rysunek powyżej), a chcemy przejść do katalogu *instalacje*, to możemy to zrobić na dwa sposoby:

1. **ścieżka względna:** `cd ../instalacje`
2. **ścieżka bezwzględna** `cd /instalacje`

Jeśli chcielibyśmy przejść z katalogu *praca* do katalogu *studia*, to składnia polecenia będzie wyglądała następująco:

1. **ścieżka względna** `cd ../studia`
2. **ścieżka bezwzględna** `cd /moje dokumenty/studia`

4.2. Znaki specjalne

Nazwy plików są najczęściej stosowanymi argumentami w poleceniach. Często można znać jedynie część nazwy pliku lub chcieć odwołać się do kilku nazw plików. Powłoka oferuje zestaw znaków specjalnych, które poszukują, dopasowują, generują listę plików. Tymi znakami specjalnymi są gwiazdka, znak zapytania i nawiasy kwadratowe. Znaki * i ? określają niepełną nazwę pliku, nawiasy kwadratowe pozwalają na określenie zestawu dopuszczalnych znaków, jakie mają być poszukiwane. Można łączyć nawiasy kwadratowe z innymi znakami specjalnymi.

- Gwiazdka * oznacza dowolny ciąg znaków, można jej używać do oznaczania nazw plików zaczynających się lub kończących dowolnym zestawem znaków.
- Znak zapytania ? oznacza dokładnie jeden dowolny znak. Można używać więcej niż jednego znaku zapytania w każdym miejscu wzorca.

- Dowolne znaki umieszczone w nawiasach kwadratowych [] oznaczają dokładnie jeden spośród wymienionych znaków.

Znaki specjalne mogą być maskowane za pomocą znaku „\”. Aby nazwa ce? odnosiła się do dokładnie jednego pliku ce?, a nie do całej grupy plików o 3-literowych nazwach rozpoczynających się znakami ce, należy za pomocą „\” zamaskować znaczenie symbolu „?” – a więc zastosować nazwę ce\?

Przykłady operacje na plikach z zastosowanie znaków specjalnych:

```
rm * cp /bin/a*b
$HOME/ cp ???
$HOME/ rm [Aa-f0-9]*
```

4.3. Polecenia dotyczące katalogów

4.3.1.1. Listowanie zawartości katalogów – ls

Wylistowanie bieżącego katalogu:

```
ls
```

Wylistowanie danego katalogu:

```
ls nazwa_katalogu
```

Przykładowe opcje:

- a wypisuje wszystkie pliki, razem z plikami ukrytymi,
- l podaje wszystkie dane (pełną informację) o plikach i katalogach,
- x wyświetla pliki posortowane według rozszerzeń

4.3.2. Zmiana katalogu - cd

Przejsie do katalogu o podanej nazwie:

```
cd nazwa_katalogu
```

Przejdźcie do katalogu domowego (trzy możliwości):

```
cd $HOME  
cd ~ cd
```

przejdźcie do katalogu nadrzędnego

```
cd ..
```

4.3.3. Tworzenie katalogów - mkdir

```
mkdir /usr/students
```

Przykładowe opcje

- p - pozwala utworzyć zagnieżdżoną strukturę katalogów
- m - pozwala na utworzenie katalogu z określonymi prawami dostępu

4.3.4. Usuwanie katalogów – rmdir

Usunięcie pustego katalogu:

```
rmdir /home/Kowalski
```

4.3.5. Przenoszenie katalogów, zmiana nazwy katalogu - mv

Zmiana nazwy:

```
mv old_name new_name
```

przeniesienie katalogu kat1 do katalogu kat2:

```
mv kat1 kat2
```

4.3.6. Gdzie jestem? - pwd

Wyświetlenie bieżącego katalogu:

```
pwd
```

4.4. Polecenia dotyczące plików

4.4.1. Kopiowanie plików - cp

Kopiowanie pliku:

```
cp plik_źródłowy plik_docelowy
```

Przykładowe opcje

- b - utworzenie kopii plików przed zniszczeniem ich zawartości,
- v - wyświetlenie nazwy każdego pliku w czasie kopiowania,
- r - kopiowanie całych katalogów wraz z podkatalogami,
- p - kopiowanie plików do katalogu docelowego z zachowaniem hierarchii podkatalogów

4.4.2. Przenoszenie plików, zmiana nazwy - mv

Przenoszenie:

```
mv plik_źródłowy plik_docelowy
```

Zmiana nazwy:

```
mv stara_nazwa nowa_nazwa
```

Przykładowe opcje

- f - brak pytania o potwierdzenie przed zniszczeniem zawartości plików,
- b - utworzenie kopii zapasowej nadpisywanych plików

4.4.3. Usuwanie plików - rm

Usunięcie pliku:

```
rm nazwa_pliku
```

Usunięcie katalogu z plikami (i podkatalogami)

```
rm -r nazwa_katalogu
```

Przykładowe opcje:

- f - usuwa pliki bez żadnego uprzedzenia,
- r - pozwala na usuwanie katalogu ze wszystkimi podkatalogami i plikami,
- i - żąda potwierdzenia dla każdego usuwanego pliku,
- v - wyświetla nazwy każdego kasowanego pliku.

4.4.4. Tworzenie powiązań symbolicznych - ln

Utworzenie dowiązania ("linku") symbolicznego:

`ln -s nazwa_katalogu nazwa_linku`

4.4.5. Tworzenie pustego pliku/aktualizacja daty - touch

Utworzenie pliku:

```
touch nazwa_pliku
```

4.4.6. Tworzenie i edycja pliku

Aby wypełnić plik treścią plik w najprostszy możliwy sposób, należy wydać polecenie:

```
cat > nazwa_pliku
```

(zakończenie edycji: ENTER, CTRL + D)

4.4.7. Wyszukiwanie plików - find

Polecenie find ma składnię o następującej postaci:

```
find gdzie filtry obiekt_poszukiwań
```

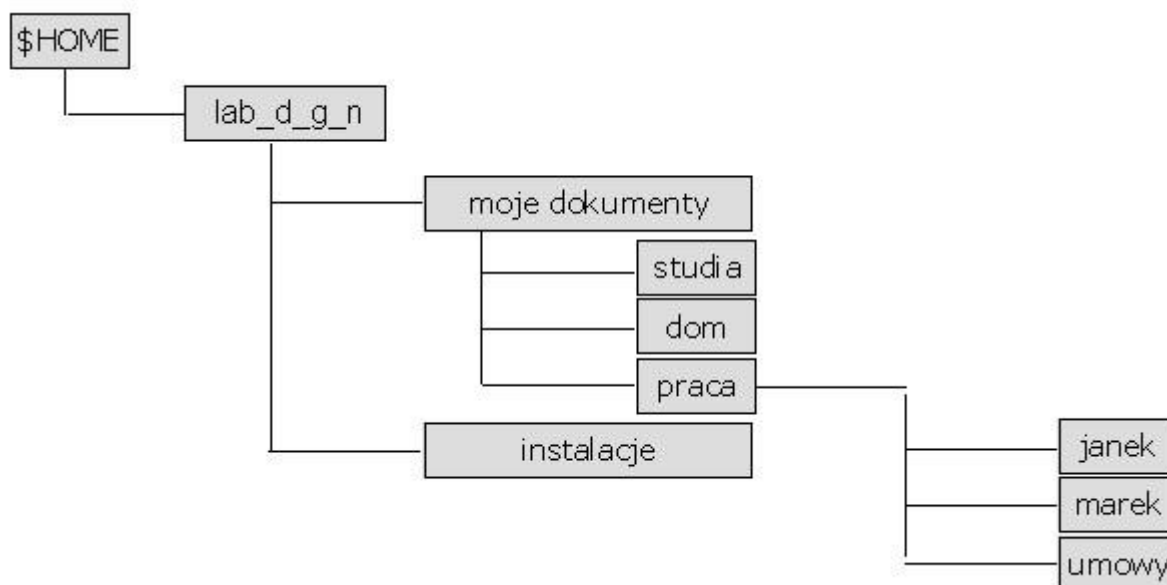
Przykład - wyszukanie w katalogu głównym (i podkatalogach) pliku o nazwie "cp":

```
find / -name cp
```

4.5. Zadania

1. Zapoznaj się z mechanizmem poruszania się po katalogach za pomocą ścieżki względnej i bezwzględnej.
2. Z katalogu domowego wylistuj jednym poleceniem zawartość katalogu */etc*

3. Utwórz następującą strukturę katalogów:



Struktura katalogów do zadania 3

Katalog proszę tworzyć według następującej konwencji:

Lab_imie

Przykład

Lab_Ala

4. skopiuj wszystkie polecenia dwuliterowe, zaczynające się od litery *d*, z katalogu */bin* do założonego wcześniej katalogu *studia*
5. zmień nazwy tych plików, aby zaczynały się od słów *stary_*
6. załóż w swoim katalogu domowym plik *info.txt*, w którym zamieść podstawowe informacje o sobie – imię, nazwisko, parę słów o sobie
7. załóż drugi plik o nazwie *email.txt*, w którym umieść swój adres e-mail
8. utwórz w katalogu *instalacje* katalog *rpm* korzystając ze ścieżki bezwzględnej
9. utwórz w katalogu *instalacje* katalog *deb* korzystając ze ścieżki względnej
10. przejdź do katalogu *deb* (w katalog *instalacje*) a następnie przejdź za pomocą jednego polecenia do katalogu *umowy* znajdującego się w katalogu *praca*.
11. przenieś plik *email.txt* do katalogu *studia*
12. załóż nowy plik, a następnie stwórz link symboliczny do niego. Co się stanie z linkiem, jeśli usuniesz plik bazowy?
13. skopiuj do katalogu *janeK* wszystkie pliki z katalogu *studia*
14. przenieś katalog *praca* wraz z podkatalogami do katalogu *instalacje* za pomocą jednego polecenia

15. usunąć cały katalog *studia*
16. usunąć katalog *praca* wraz z podkatalogami za pomocą jednego polecenia
17. usunąć katalog *instalacje*