

Języki i metody programowania I

Informatyka Stosowana, rok 1 semestr 1

Laboratorium 1

Zajęcia wprowadzające



**Wydział
Elektryczny**

POLITECHNIKA WARSZAWSKA

Instytut Elektrotechniki Teoretycznej i Systemów Informacyjno-Pomiarowych

Semestr 2020Z

Laboratorium 1

dr inż. Zuzanna Krawczyk i mgr inż. Aleksandra Osowska-Kurczab

Program ćwiczenia:

- Regulamin przedmiotu i zasady zaliczenia
 - Zapoznanie ze środowiskiem programistycznym
 - Pierwsza kompilacja programów
-

Ćwiczenie 1. Zapoznanie ze środowiskiem programistycznym.

Język programowania: c

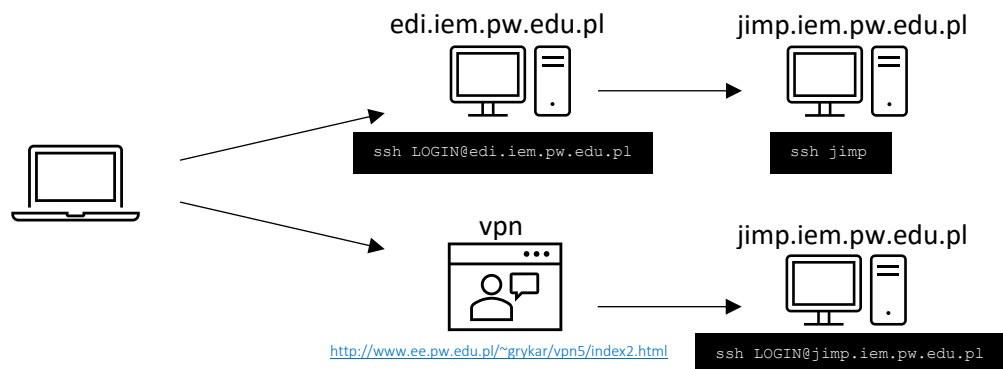
Środowisko pracy:

- Linux (domyślnie Ubuntu).
- Terminal (poruszanie się po systemie plików, kompilacja kodu itp.) + edytor `vi` (edycja plików z kodem)

Dostęp do maszyny:

Do tworzenia programów w ramach zajęć możecie Państwo korzystać ze zdalnej maszyny `jimp.iem.pw.edu.pl`. Maszyna widoczna jest tylko w sieci wydziałowej. Aby się do niej połączyć:

- „z zewnątrz”
należy najpierw zalogować się do serwera `edi.iem.pw.edu.pl` i z poziomu serwera `edi` zalogować się do maszyny `jimp`
- Poprzez VPN
połączyć się do wydziałowego vpn (szczegółowe instrukcje znajdziecie Państwo tu: <http://www.ee.pw.edu.pl/~grykar/vpn5/index2.html>) i logować bezpośrednio do serwera `jimp`.



Uwaga na maszynie `edi` nie macie Państwo możliwości kompilacji programów.

Uwierzytelnienie do serwerów `jimp` i `edi` odbywa się przy pomocy tych samych danych (**login i hasło**) jak do wirtualnego dziekanatu **ISOD**.

Do serwera łączymy się w trybie tekstowym, przy pomocy protokołu `ssh`.

```
ssh login@adres.maszyny.do.ktorej.sie.laczymy
```

np.

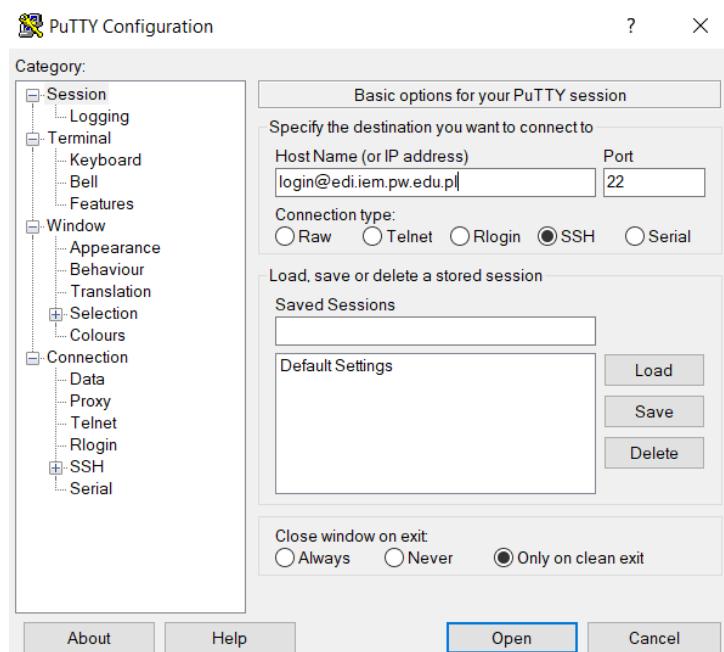
```
ssh kowalczp@edi.iem.pw.edu.pl
```

Po wciśnięciu klawisza `Enter` pojawi się miejsce do **podania hasła**. Znaki wpisywanego hasła nie są widoczne w terminalu, nie przejmujemy się tym. Po podaniu hasła należy ponownie wcisnąć `Enter`.

Korzystacie Państwo z różnych systemów operacyjnych. W zależności od systemu różni się dostęp do konsoli/ terminala przy pomocy, której będziecie mogli Państwo skorzystać z klienta `ssh`, aby połączyć się z maszyną `edi`. Poniżej kilka bardziej popularnych rozwiązań:

- **Linux/ Mac OS:** Uruchamiamy okno terminala
- **Windows 10:** Posiada wbudowanego klienta `ssh`. Uruchamiamy terminal (np. poprzez wpisanie `cmd` w okno wyszukiwania).
- **Starsze wersje systemu Windows:** nie posiadają wbudowanego klienta `ssh` możemy. Musimy zainstalować program zewnętrzny. Do najpopularniejszych należy `puTTY`: <https://www.putty.org/>.

Po uruchomieniu programu w domyślnym oknie programu (Session) w polu `host name` wpisujemy swój login@edi.iem.pw.edu.pl i klikamy „open”. Następne czynności (wpisanie hasła) wykonujemy w terminalu, który się pojawi.



Rysunek 1 Okno programu PuTTY

Do transferu plików pomiędzy komputerem domowym z Windowsem, a serwerem możecie wykorzystać Państwo program WinScp.

Zachęcamy do instalacji systemu Linux, aby lepiej go poznać:

- W Windows 10 możemy zainstalować terminal Ubuntu z oficjalnego sklepu: <https://www.microsoft.com/en-us/p/ubuntu/9nblggh4msv6?activetab=pivot:overviewtab>
- Można również zainstalować Linuxa korzystając z maszyny wirtualnej: <https://www.virtualbox.org/>

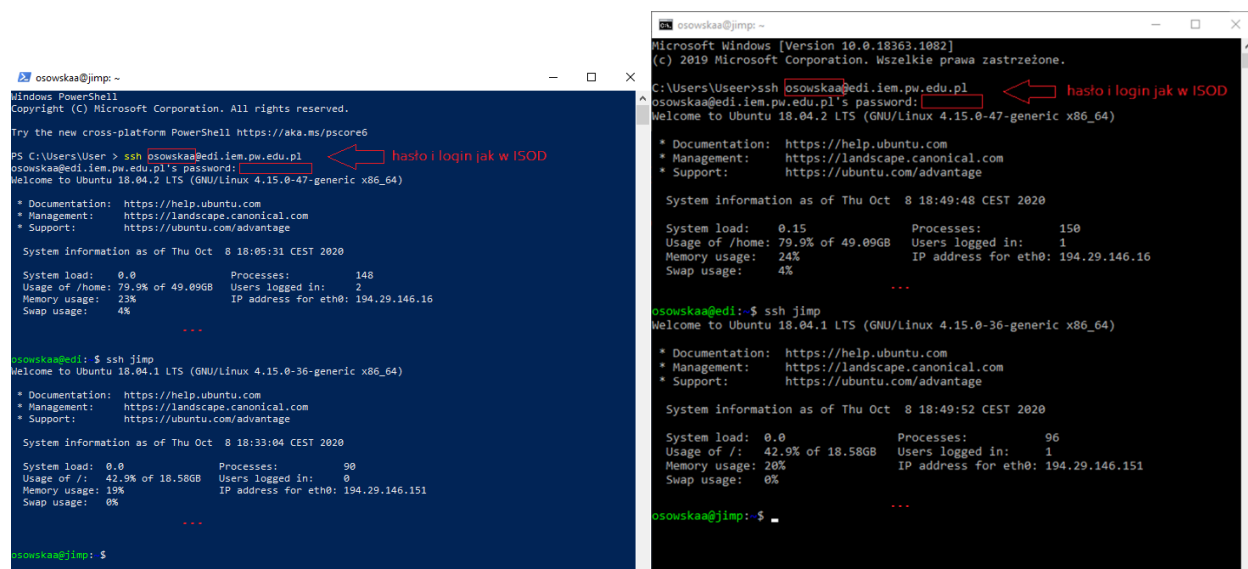
Instrukcje i materiały dydaktyczne obowiązujące na zajęciach laboratoryjnych:

<http://wikidyd.iem.pw.edu.pl/LMP.html>

Etapy logowania do maszyn w podziale na różne systemy operacyjne (bez VPN):

Windows10

1. Uruchamiamy CMD (Wiersz polecenia) lub Windows PowerShell
2. Wpisujemy komendę `ssh LOGIN@edi.iem.pw.edu.pl` (LOGIN – jak w ISOD)
3. Pojawi się prośba o wpisanie hasła. Wpisujemy hasło takie jak w ISOD, w trakcie wpisywania nie będą pojawiać się znaki – nie przejmujemy się tym
4. Maszyna przywita się z nami informacjami o systemie itp. Pojawi się znak zachęty
`osowskaa@jimp:~$`
5. Wpisujemy komendę `ssh jimp`
6. Maszyna `jimp` przywita się z nami podobnie jak `edi`. Od tego momentu możemy swobodnie działać na maszynie `jimp`.



```
osowskaa@jimp:~$ ssh osowskaa@edi.iem.pw.edu.pl
osowskaa@edi.iem.pw.edu.pl's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Oct  8 18:05:31 CEST 2020

System load:  0.0          Processes:    148
Usage of /home: 79.9% of 49.09GB   Users logged in:  2
Memory usage:  23%        IP address for eth0: 194.29.146.16
Swap usage:    4%

...

osowskaa@edi:~$ ssh jimp
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Oct  8 18:33:04 CEST 2020

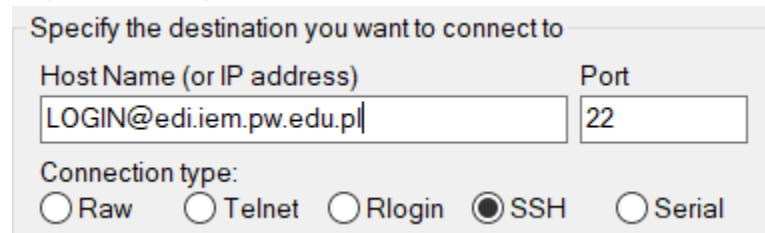
System load:  0.0          Processes:    90
Usage of /:   42.9% of 18.58GB   Users logged in:  1
Memory usage: 19%        IP address for eth0: 194.29.146.151
Swap usage:   0%

...

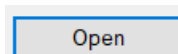
osowskaa@jimp:~$
```

Windows (dowolna wersja)

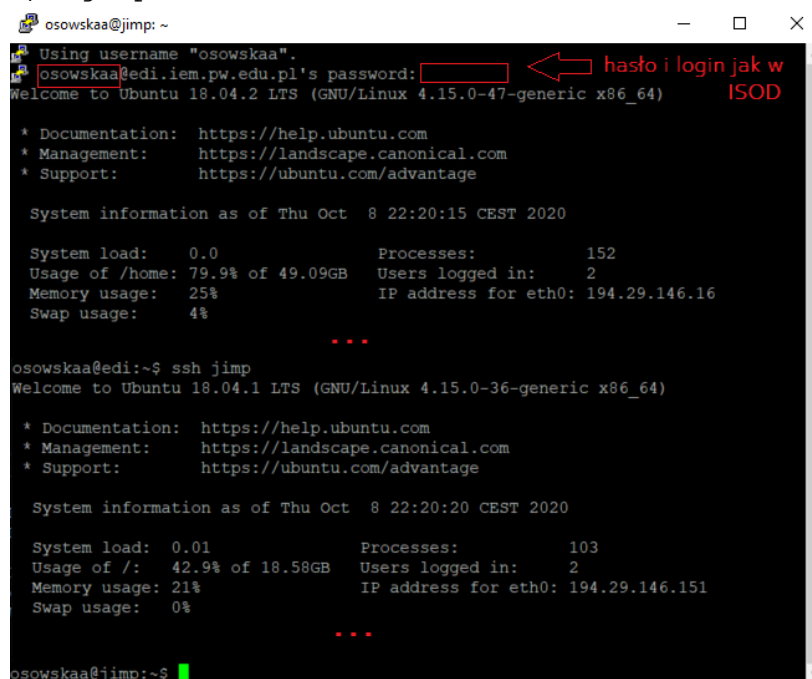
1. Pobrać i zainstalować PuTTY (<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>)
2. Uruchomić PuTTY
3. W polu **HostName (or IP address)** wpisujemy LOGIN@edi.iem.pw.edu.pl (LOGIN – jak w ISOD)



4. Klikamy przycisk Open na dole okienka



5. Pojawi się terminal, prosząc nas o hasło. Wpisujemy hasło takie jak w ISOD, w trakcie wpisywania nie będą pojawiać się znaki – nie przejmujemy się tym
 6. Maszyna przywita się z nami informacjami o systemie itp. Pojawi się znak zachęty:
- ```
osowskaa@jimp:~$
```
7. Wpisujemy komendę `ssh jimp`
  8. Maszyna jimp przywita się z nami podobnie jak edi. Od tego momentu możemy swobodnie działać na maszynie jimp.



## Linux / Mac OS

1. Uruchamiamy terminal
2. Wpisujemy komendę `ssh LOGIN@edi.iem.pw.edu.pl` (LOGIN – jak w ISOD)
3. Pojawi się prośba o wpisanie hasła. Wpisujemy hasło takie jak w ISOD, w trakcie wpisywania nie będą pojawiać się znaki – nie przejmujemy się tym
4. Maszyna przywita się z nami informacjami o systemie itp. Pojawi się znak zachęty:  

```
osowskaa@jimp:~$
```
5. Wpisujemy komendę `ssh jimp`
6. Maszyna jimp przywita się z nami podobnie jak edi. Od tego momentu możemy swobodnie działać na maszynie jimp.

```
air-zuzanna:~ noemi$ ssh krawczyk@edi.iem.pw.edu.pl
krawczyk@edi.iem.pw.edu.pl's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-47-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Thu Oct 8 23:43:28 CEST 2020

System load: 0.0 Processes: 152
Usage of /home: 79.9% of 49.09GB Users logged in: 3
Memory usage: 24% IP address for eth0: 194.29.146.16
Swap usage: 4%

krawczyk@edi:~$ ssh jimp
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-36-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Thu Oct 8 23:43:45 CEST 2020

System load: 0.0 Processes: 105
Usage of /: 43.0% of 18.58GB Users logged in: 2
Memory usage: 21% IP address for eth0: 194.29.146.151
Swap usage: 0%

krawczyk@jimp:~$
```

## Ćwiczenie 2. Command line i edytor vim.

W tym ćwiczeniu zostaną zaprezentowane podstawowe komendy terminalowe, z których będziecie Państwo regularnie korzystać podczas korzystania z maszyny wirtualnej *edi*.

### Rozpoczęcie pracy – połączenie z maszyną wirtualną

Po połączeniu z maszyną *edi* (`ssh edi.iem.pw.edu.pl`) i zalogowaniu się (login i hasło jak na ISOD) należy przejść do specjalnie skonfigurowanego środowiska *jimp* dla tego przedmiotu (`ssh jimp`). Maszyna przywita nas ekranem:

```
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-36-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Tue Oct 6 21:53:08 CEST 2020

System load: 0.09 Processes: 94
Usage of /: 42.2% of 18.58GB Users logged in: 0
Memory usage: 20% IP address for eth0: 194.29.146.151
Swap usage: 0%
```

Na dole terminala pojawia się znak zachęty `LOGIN@jimp:~$`, możemy rozpocząć pracę (LOGIN to indywidualny login do konta na ISOD).

Po ostatecznym zakończeniu pracy ze środowiskiem kończymy połączenie z maszyną wirtualną poleceniem `exit`. [Na potrzeby dalszej pracy proszę nie wykonywać tego polecenia w konsoli w tym momencie ćwiczenia].

### Podstawowe nawigowanie

Domyślnie po zalogowaniu użytkownik znajduje się w swoim folderze domowym. Obecną ścieżkę (katalog domowy użytkownika) można wyświetlić poleceniem `pwd`. Zawartość ścieżki, na której się znajduje użytkownik można wyświetlić poleceniem `ls`. Po pierwszym zalogowaniu do środowiska polecenie `ls` pokaże nam, że mamy do czynienia z pustym folderem.

Zacznijmy od założenia folderu dla tego przedmiotu. Po wykonaniu polecenia `mkdir JIMP1` i wylistowaniu zawartości folderu poleceniem `ls` widzimy, że został stworzony katalog o nazwie `JIMP1`. Możemy zmienić naszą ścieżkę i „wejść” do nowo utworzonego folderu komendą `cd JIMP1`. Gdy teraz wykonamy komendę `pwd` a następnie `ls` zauważymy, że znajdujemy się w pustym katalogu `JIMP1`.

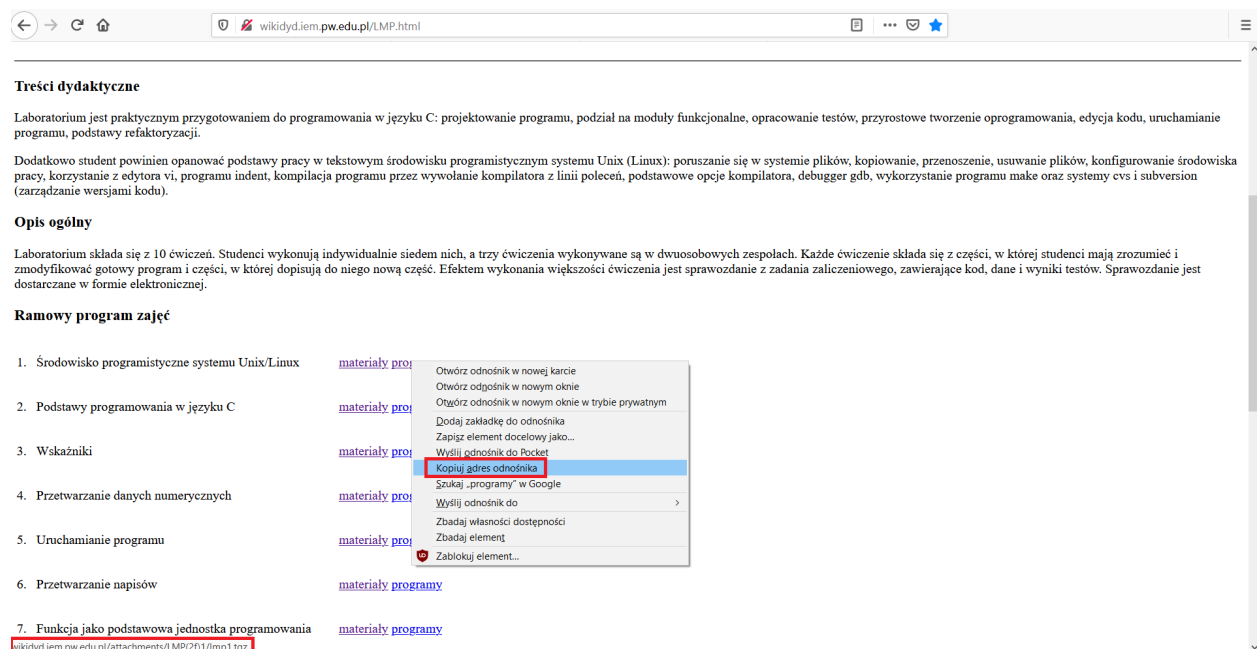
Możemy wrócić do katalogu domowego poleceniem `cd ..`. Podwójna kropka `..` oznacza ścieżkę folderu nadrzędnego (powyżej tego, w którym obecnie się znajdujemy). Pojedyncza kropka `.` oznacza aktualną ścieżkę i często stanowi domyślny argument wielu komend (np. `ls` oraz `ls .` wylistuje taką samą zawartość). Tylda `~` to oznaczenie ścieżki katalogu domowego (home usera).

W następnych krokach będziemy tworzyć pliki i kolejne foldery oraz nimi operować w środowisku. Zalecamy częste podpatrywanie zawartości tworzonych ścieżek korzystając z odpowiedniej kombinacji komend `cd`, `pwd`, `ls`.

## Tworzenie testowego środowiska

Ponownie przejdźmy do katalogu JIMP1 (`cd /home/LOGIN/JIMP1`) oraz stwórzmy sobie katalog odpowiadający pierwszym zajęciom laboratoryjnym `lab01` (`mkdir lab01`). Następnie stworzymy sobie 2 foldery pomocnicze `source` oraz `zad0` (`mkdir lab01/source lab01/zad0`). Warto zauważyć, że jednocześnie powstały 2 foldery, które określiliśmy jako ścieżkę względną od naszego obecnego folderu JIMP1 (powstały one w podkatalogu `lab01`).

Katalog `source` będzie przechowywać materiały dydaktyczne do laboratorium 1. pobrane ze strony <http://wikidyd.iem.pw.edu.pl/LMP.html>. Aby pobrać materiały bezpośrednio z internetu bez przekopiowywania skorzystamy z komendy `wget "URL"`. Będziemy potrzebować do tego adresu url pliku, który chcemy pobrać. Po wejściu na stronę z materiałami wikidyd wyświetli się nam on w lewej dolnej części ekranu po najechaniu kursorem na link [programy](#) przy laboratorium 1 (screen dla przeglądarki Mozilla Firefox). Adres możemy również skopiować do schowka systemowego (prawy przycisk myszy >> Kopiuj adres odnośnika).



Pobierzmy zatem skompresowany plik z materiałami do kolejnych ćwiczeń.

```
cd lab01/source
wget "http://wikidyd.iem.pw.edu.pl/attachments/LMP(2f)1/lmp1.tgz"
ls
```

Pobrano został plik `lmp1.tgz`. Pozostaje nam już tylko odpakowanie pobranego pliku.

```
tar -xzf lmp1.tgz
ls
```

Powstał folder `lmp1` zawierający pliki `p1.c p2.c p3.c p4.c`.

Jak inaczej możemy przekazać pliki na naszą maszynę wirtualną?

Służy do tego polecenie `scp` (secure copy), które korzysta z protokołu `ssh` i pozwala na kopiowanie plików pomiędzy hostami. Składnia komendy:



`scp ŚCIEŻKA_ŹRÓDŁOWA ŚCIEŻKA_DOCELOWA`, gdzie ścieżka serwera musi być poprzedzona adresem hosta (serwera):

`LOGIN@HOSTNAME:ŚCIEŻKA`

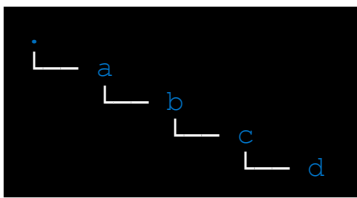
np. `kowalskij@edi.iem.pw.edu.pl:/home/kowalskij/JIMP1/lab01/source`

Dla osób korzystających z systemu Windows przydatnym może okazać się program okienkowy WINSCP, który udostępnia wygodne przesyłanie plików między hostami. Polecamy jednak zaznajamianie się z terminalem.

### Korzystanie z pomocy i podpowiedzi

Już niedługo w kolejnych semestrach zajęć przekonacie się Państwo, że umiejętne googlowanie to duża część pracy informatyka. W command line całe szczęście macie Państwo cały system pomocy, który pozwoli Wam korzystać wygodnie z narzędzi i zwolni Państwa z pamiętania wielu flag i opcji dostępnych komend. Przetestujemy to na przykładzie rekursywnego tworzenia ścieżki z pośrednimi katalogami.

Zacznijmy od przeniesienia się do folderu `cd /home/LOGIN/JIMP1/lab01/zad0`. Założmy, że chcemy stworzyć drzewko folderów wewnątrz `zad0` wyglądające w następujący sposób:



Próba wywołania komendy `mkdir a/b/c/d` zakończy się błędem:

```
mkdir: cannot create directory 'a/b/c/d': No such file or directory
```

Możemy to zrobić pojedynczym wywołaniem komendy `mkdir`, ale musimy do niej dodać specjalną komendę sterującą (flagę). Może się zdarzyć, że nie pamiętamy/nie znamy tej flagi, ale możemy podejrzeć ją we wbudowanym helpie. Możemy uruchomić go na 2 sposoby:

`man cmd` (gdzie `cmd` to komenda, którą chcemy dokładniej poznać), czyli w przypadku `mkdir` → `man mkdir` (otworzy się nowa strona pokazująca dokładną dokumentację komendy, wychodzimy klikając `q`)

```
MKDIR(1) User Commands MKDIR(1)
NAME
 mkdir - make directories

SYNOPSIS
 mkdir [OPTION]... DIRECTORY...

DESCRIPTION
 Create the DIRECTORY(ies), if they do not already exist.

 Mandatory arguments to long options are mandatory for short options
 too.

 -m, --mode=MODE
 set file mode (as in chmod), not a=rwx - umask

 -p, --parents
 no error if existing, make parent directories as needed

 -v, --verbose
 print a message for each created directory

Manual page mkdir(1) line 1 (press h for help or q to quit)
```

lub

`mkdir --help` (w terminalu wyświetli się skrócona forma dokumentacji)

```
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
 -m, --mode=MODE set file mode (as in chmod), not a=rwx - umask
 -p, --parents no error if existing, make parent directories as needed
 -v, --verbose print a message for each created directory
 -Z set SELinux security context of each created directory
 to the default type
 --context[=CTX] like -Z, or if CTX is specified then set the SELinux
 or SMACK security context to CTX
 --help display this help and exit
 --version output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report mkdir translation bugs to <http://translationproject.org/team/>
Full documentation at: <http://www.gnu.org/software/coreutils/mkdir>
or available locally via: info '(coreutils) mkdir invocation'
```

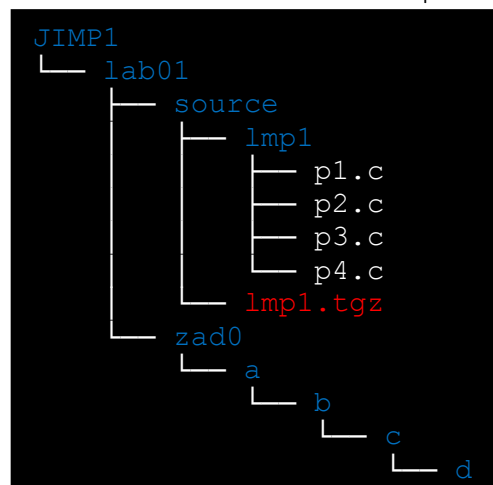
Zauważmy, że polecenie `mkdir` wywołane z flagą `-p` (lub `--parents`) pozwoli nam stworzyć pośrednie foldery w ścieżce. Ostateczne wywołanie komendy to `mkdir -p a/b/c/d`. Możemy sprawdzić czy poprawnie udało nam się wygenerować drzewko katalogów korzystając z komendy `tree`.

Inną mechaniką podpowiedzi, która jest szczególnie użyteczna przy nawigowaniu po plikach jest klawisz `Tab` (tabulacja). Przeszukuje on pliki, które mogłyby być dopełnieniem wpisywanego aktualnie tekstu, np. poniżej w folderze `KATALOG` znajdują się 3 możliwe lokalizacje rozpoczynające się literą `a`. W momencie, gdy jest tylko jedno poprawne dopełnienie to po naciśnięciu `Tab` komenda automatycznie się uzupełni.

```
LOGIN@jimp:~/KATALOG$ ls a
a/ ala alicja
```

### Tworzenie dokumentów tekstowych i zmiana ich położenia

Obecnie nasze drzewko folderów i plików powinno wyglądać następująco:



Stworzymy sobie teraz 2 testowe pliki `plik1` oraz `plik2`. Znajdujemy się w folderze `zad0`. Skorzystamy z poleceń:

```
echo "Ala ma kota" > a/b/c/d/plik1 (tworzy plik1 z tekstem „Ala ma kota”)
echo "a Kasia ma psa" >> a/b/c/d/plik1 (dopisuje do pliku1 kolejną linię „a Kasia ma psa”)
echo "Kasia ma psa" > a/b/c/plik2
```

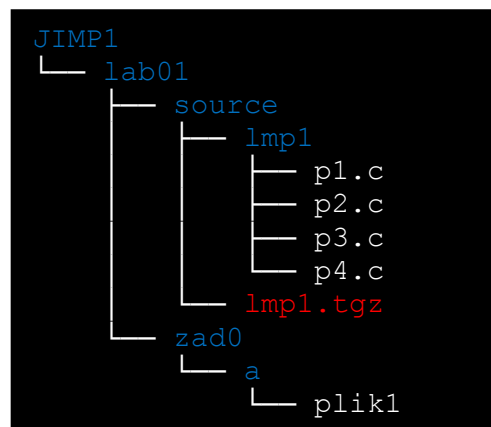
Możemy wyświetlić zawartość obydwu plików komendami `cat a/b/c/d/plik1` oraz `cat a/b/c/plik2`.

Możemy kopiować pliki komendą `cp SOURCE DESTINATION`. Przykładowo komenda `cp a/b/c/d/plik1 a/b/plik1_copy` skopiuje `plik1` z katalogu `d` do `plik1_copy` w katalogu `b`.

Możemy przenosić pliki między ścieżkami komendą `mv SOURCE DESTINATION`. Komenda `mv a/b/plik1_copy a/` przeniesie `plik1_copy` z folderu `a/b/` do folderu `a` bez zmiany nazwy tego pliku (domyślnie gdy damy pustą nazwę pliku pozostanie ona taka sama jak początkowo, tak samo działa to np. w komendzie `cp`). Zauważmy, że komendą `mv` możemy też zmieniać nazwę plików np. po wykonaniu komendy `mv a/plik1_copy a/plik1` w folderze `a` będzie znajdować się plik o nazwie `plik1`, a nie `plik1_copy`.

Pliki usuwamy komendą `rm PLIK`, a katalogi `rm -r KATALOG`. Przykładowo możemy usunąć `plik1` w katalogu `d` komendą `rm a/b/c/d/plik1`. Możemy usunąć też katalog `b` z całą zawartością komendą `rm -r a/b`.

Na sam koniec pozostaje nam następujące drzewko katalogów:



## Edytor tekstu vi

Będziemy korzystać z edytora tekstu vi. Na maszynie dostępna jest również wersja rozszerzona edytora (vim). W przypadku naszych zastosowań nie ma większej różnicy między obydwoimi edytorami (domyślnie proszę wybierać vi). Vi to notatnik terminalowy, dzięki któremu możemy (w miarę) wygodnie edytować pliki tekstowe. Ma niesamowicie dużo możliwości i jest to ABC każdego informatyka.

### Jak edytować plik przy pomocy vi?

`vi` – zostanie stworzony nowy plik

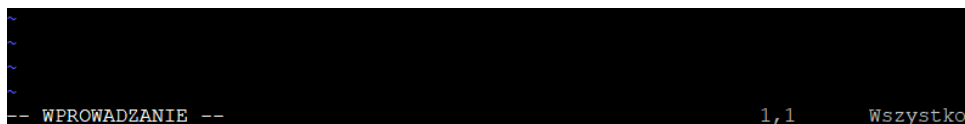
`vi PLIK` – będzie edytowany istniejący PLIK

### Tryby pracy

Vi ma kilka trybów pracy (dokładnie 6), my skupimy się na czterech głównych:

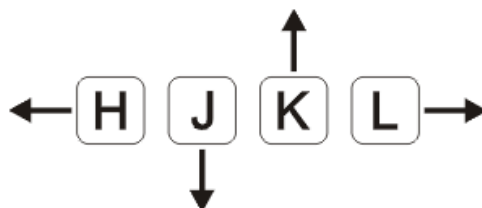
- Normal mode (uruchamiany klawiszem `Esc`) – pozwala na swobodne przeglądanie edytowanego pliku i usuwanie znaków
- Insert mode (`i`, `a` lub `s`) – pozwala na wstawianie nowych znaków.
- Visual mode (`v`) – pozwala na usuwanie, kopiowanie, wycinanie i wklejanie tekstów. W zaznaczeniu można korzystać z klawiszy kierunkowych (o tym poniżej) lub myszki.
- Cmd-line (`:`) – tryb poleceń, pozwala wpisywać proste komendy np. wyjść z vima

Tryby pracy widoczne są na samym dole terminala, gdy uruchomiony jest vi. Poniżej widoczny jest tryb WPROWADZANIE (Insert mode).



### Klawisze kierunkowe

Poruszanie się po otwartym dokumencie w vimie może odbywać się na 2 sposoby (klawiszami `H` `J` `K` `L` lub strzałkami `←` `↑` `↓` `→`). Obsługa klawiszami `H` `J` `K` `L` po jakimś czasie stanie się naturalna i bardzo szybka, więc warto ćwiczyć od początku nawigację tymi 4 klawiszami.



Wciśnięcie przycisków kierunkowych z `Shift` powoduje przeskok o całą stronę, a z `Ctrl` o jedno słowo.

## Normal mode

Dostępny zaraz po odpaleniu vima, albo po naciśnięciu klawisza `Esc`. Możemy wtedy poruszać się po dokumencie z użyciem klawiszy kierunkowych oraz usuwać znaki w dokumencie:

- Nakierowując na konkretny znak i naciskając klawisz `x`
- Nakierowując na linię do usunięcia i dwukrotnie naciskając klawisz `d` (`d` `d`)

Aby cofnąć wprowadzone zmiany wystarczy nacisnąć klawisz `u` (vimowe Ctrl+Z dostępne w trybie Normal), a żeby ponownie te zmiany wprowadzić należy wcisnąć `Ctrl+R` (vimowe Ctrl+Y dostępne w trybie Normal).

## Insert mode (wprowadzanie)

Dostępny po naciśnięciu klawisza `i` lub `a` będąc w trybie Normal.

- `i` – wstawianie tekstu na początku linii, w której znajduje się kursor
- `a` – wstawianie tekstu tuż za kursorem

Następnie nawigując się strzałkami po tekście (lub przechodząc do trybu Normal i posługując się klawiszami kierunkowymi) możemy normalnie edytować tekst.

## Visual mode

Dostępny po naciśnięciu klawisza `v` będąc w trybie Normal. Operuje głównie na 3 komendach:

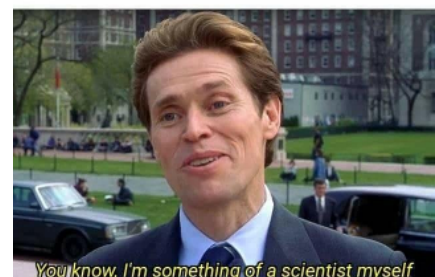
- `d` (delete) = CUT, wytnij
- `y` (yank) = COPY, kopiuje
- `p` (paste) = PASTE, wklej

## !Wychodzenie z vima! – Tryb command

Przechodzimy do trybu Normal wciskając `Esc`. Następnie zależnie od pożądanego skutku edycji pliku wpisujemy:

- `:w` – zapisz plik, ale bez wyjścia z vi
- `:wq` lub `:x` – zapisz plik i wyjdź z vi
- `:q` – wyjdź z vi (jeżeli są niezapisane zmiany to pojawi się błąd)
- `:q!` – wyjdź z vi i nie zapisuj wprowadzonych zmian

When you finally exit vim



### Proste ćwiczenie w vi

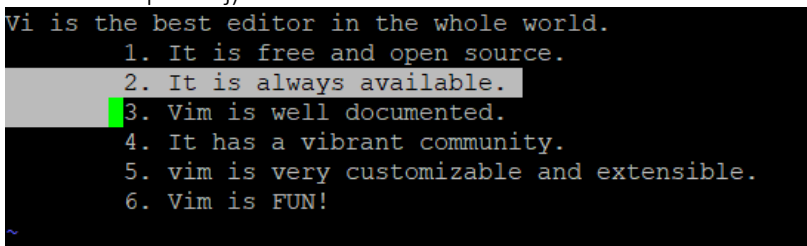
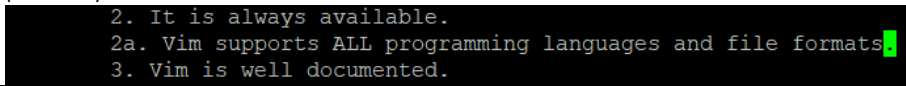
Na początku zajrzemy do pliku stworzonego w poprzedniej części ćwiczenia (plik `zad0/a/plik1`).

Proszę wykonać następującą sekwencję „komend”:

|                          |                                                                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <code>vi a/plik1</code>  | Otwieramy vimem plik1 znajdujący się w katalogu a                                                                                     |
|                          | Znajdujemy się w trybie Normal. Możemy nauczyć się sterowania klawiszami kierunkowymi <code>H J K L</code> lub <code>← ↑ ↓ →</code> . |
| <code>d d</code>         | Usuujemy linię, w której się znajdowaliśmy kursorem.                                                                                  |
| <code>u</code>           | Cofamy usunięcie linii.                                                                                                               |
| <code>Ctrl + R</code>    | Ponawiamy usunięcie linii.                                                                                                            |
| <code>u</code>           | Ponownie cofamy usunięcie linii. Znowu mamy pełny tekst<br>„Ala ma kota<br>a Kasia ma psa”                                            |
| <code>x</code>           | Ustawiamy kursor na słowie Kasia i ponawiając przyciskanie klawisza <code>x</code> usuwamy całe słowo z linii.                        |
| <code>i</code>           | Przechodzimy do trybu Insert.                                                                                                         |
|                          | Kursorem jesteśmy ustawieni w linii „a <code>ma</code> psa” na literce m. Zaczynamy wprowadzać słowo „Michał”.                        |
| <code>Esc</code>         | Przechodzimy do trybu Normal.                                                                                                         |
| <code>:q!</code>         | Nie decydujemy się na zapis pliku, chcemy cofnąć wszystkie zmiany i wrócić do stanu pliku przed edycją.                               |
| <code>cat a/plik1</code> | Sprawdzamy czy plik zmienił się w porównaniu do początkowego<br>„Ala ma kota<br>a Kasia ma psa”.                                      |

Teraz samodzielnie stworzymy prosty plik korzystając z poznanych wcześniej komend.

|                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>vi</code>                                                                                                                       | Otwieramy vima bez wymienionego konkretnego pliku.                                                                                                                                                                                                                                                                                                                                                                              |
|                                                                                                                                       | Prawdopodobnie pojawi się ekran powitalny, który zniknie zaraz po rozpoczęciu edycji pliku. Witamy w trybie Normal.<br>Kopiujemy do schowka ( <code>Ctrl+C</code> ) poniższy tekst bez znaków „”:<br>„Vi is the best editor in the whole world. 1. It is free and open source. 2. It is always available. 3. Vim is well documented. 4. It has a vibrant COMMUNITY. 5. VIM is very customizable and extensible. 6. Vim is fun!” |
| <code>Shift + Insert</code><br><code>Środkowy klawisz myszy</code><br><code>" + p</code> lub <code>i</code> <code>ctrl+Shift+V</code> | Wklejamy tekst<br>JEŻELI UŻYWASZ PUTTY (oraz we wszystkich pozostałych warunkach)<br>JEŻELI JESTEŚ POŁĄCZONY PRZEZ SSH<br>JEŻELI DZIAŁASZ LOKALNIE WE WŁASNYM TERMINALU (nie używasz ssh)                                                                                                                                                                                                                                       |
| <code>i</code>                                                                                                                        | Wchodzimy do trybu Insert jeżeli jeszcze w nim nie jesteśmy. Dodajmy enterzy rozdzielające konkretne podpunkty oraz tabulatory przed numeracją.                                                                                                                                                                                                                                                                                 |

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Esc</b>              | Przechodzimy do trybu Normal. Przechodzimy kursorem do słowa COMMUNITY w podpunkcie 4.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>v</b>                | Zmieniamy tryb na tryb Visual.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>~</b>                | Klikamy kilkakrotnie tyldę (~) aby COMMUNITY zamieniło się w community.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|                         | <p>Powtarzamy poprzednie 3 kroki w słowie VIM w podpunkcie 5. oraz w słowie fun w podpunkcie 6.</p> <p>Na ten moment mamy tekst</p> <p>“Vi is the best editor in the whole world.</p> <ol style="list-style-type: none"> <li>1. It is free and open source.</li> <li>2. It is always available.</li> <li>3. Vim is well documented.</li> <li>4. It has a vibrant community.</li> <li>5. vim is very customizable and extensible.</li> <li>6. Vim is FUN!”</li> </ol> <p>Będąc w trybie normal przechodzimy kursorem do podpunktu 2.</p> |
| <b>0</b>                | Wciskając klawisz <b>0</b> (zero) przenosimy się na początek linii z podpunktem 2.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>v</b>                | Przechodzimy do trybu wizualnego                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                         | <p>Zaznaczamy cały drugi punkt razem z łamaniem linii klawiszami kierunkowymi (jak na screenie poniżej).</p>                                                                                                                                                                                                                                                                                                                                         |
| <b>y</b>                | Kopiujemy zaznaczony fragment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|                         | Przesuwamy się klawiszami kierunkowymi na początek linii 3.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>p</b>                | Wklejamy skopiowany tekst.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>i</b>                | <p>Wchodzimy do trybu Insert i poprawiamy nowo wklejoną linię aby zawierała poniższy tekst</p>                                                                                                                                                                                                                                                                                                                                                      |
| <b>Esc</b>              | Czytamy cały tekst i utrwalamy sobie czemu vim jest tak dobrym edytorem. Następnie wracamy do trybu Normal.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>:wq vimIsTheBest</b> | Zapisujemy powstały plik pod nazwą vimIsTheBest.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>cat vimIsTheBest</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

**BARDZO** wskazane jest przejście przez tutorial vima dostępny bezpośrednio z terminala po wpisaniu komendy **vimtutor** – w pełni po polsku i bardzo przystępnie napisany! 😊 Powoli będziecie Państwo przyzwyczajać się do pracy w tym narzędziu, należy tylko dać sobie chwilę czasu i nie zrażać się na samym początku.

### Co dalej?

- Zapoznać się z cheatsheetem do command line'a i vim'a – potestować inne wylistowane komendy.
- Potrenować obsługę vima korzystając z dostępnych materiałów internetowych np. <https://www.openvim.com/>, lub np. z komendy `vimtutor` w terminalu.

### Dla chętnych (pisanie skryptów w bashu):

- Napisać w bashu skrypt automatycznie tworzący serię folderów (na jednym poziomie zagnieżdżenia) na każde laboratorium z przedmiotu (jest ich 10) – lab01, lab02, lab03... . Należy skorzystać z pętli/sekwencji (1-10) i zmiennych (zmienna przechowująca prefix „lab”). Skrypty zapisujemy zazwyczaj jako pliki z rozszerzeniem .sh (np. create10dirs.sh). Zanim wywołamy skrypt komendą `./create10dirs.sh` należy zmienić uprawnienia nadane utworzonemu plikowi, aby user miał prawo do wywoływania (ang. execute) (np. komendą typu `chmod 700 create10dirs.sh` lub `chmod u+x create10dirs.sh`).
- Napisać w bashu skrypt automatycznie tworzący rekursyjne drzewko folderów 1/2/3/4/5/.../n, gdzie n to argument wywołania skryptu. Argumenty wywołania skryptu przechowywane są w zmiennych \$1 (pierwszy argument), \$2 (drugi argument) itp. Używać n do max 10 (dla czytelności). Skrypt powinien korzystać z pętli/sekwencji. Powstałe drzewko można sprawdzić poleceniem `tree`.

### Poniżej zestawienie kilku najczęściej używanych na zajęciach komend:

- `ls` – listowanie plików
- `pwd` – wypisuje bezwzględną ścieżkę do bieżącego katalogu
- `cd` – przemieszczanie się pomiędzy katalogami np. `cd sciezka/do/katalogu` – przenosi nas do katalogu podanego w ścieżce; `cd ../` - przeniesienie do katalogu nadrzędnego, `cd` – bez argumentów, przeniesienie do katalogu domowego
- `mkdir nazwa` – tworzy katalog o nazwie „nazwa”
- `rm plik` – usuwa plik
- `cat plik` – wypisuje plik
- `cp` – kopiowanie plików np. `cp plik1 plik2`, `cp plik katalog`, `cp -r katalog1 katalog2`
- `mv` – przenosi dany plik/katalog pod wskazaną lokalizację np. `mv plik1 plik2`
- `diff plik1 plik2` – wypisuje różnice pomiędzy plikami plik1 i plik2
- `chmod` – zmienia uprawnienia dostępu do pliku/katalogu np. `chmod 755 plik`
- `man polecenie` – wyświetla podręcznik użytkownika dla danego polecenia
- `touch nazwa` – tworzy pusty plik o danej nazwie lub ustawia datę modyfikacji w przypadku już istniejącego pliku
- `tar` – program do umieszczania grupy pliku w zbiorczym archiwum np.
  - rozpakowanie archiwum z kompresją zip: `tar -zxvf nazwa_archiwum.tar.gz`
  - spakowanie archiwum z kompresją zip: `tar -zcvf nazwa_archiwum.tar.gz katalog1 katalog2 ... plik1 plik2...`
- `wget` – program służący do pobierania plików z internetu. np. `wget „http://adres/pliku”`



### Ćwiczenie 3. Pierwsza kompilacja programów.

Przejdźmy do katalogu w którym znajdują się program przykładowe. Jeśli wykonaliśmy poprzednie etapy instrukcji powinien to być katalog `~/JIMP1/lab01/source/lmp1`. Skompilujemy teraz pierwszy plik źródłowy o nazwie „p1.c”. Możemy to zrobić przy pomocy następującej komendy:

```
cc -Wall -ansi -pedantic p1.c
```

Uruchomiliśmy w ten sposób kompilator języka c. Wynikiem działania kompilatora powinien być plik wykonywalny o domyślnej nazwie **a.out**. Opcje:

- -Wall – włącza więcej ostrzeżeń;
- -ansi – składnia programu musi być zgodna ze standardem ansi
- -pedantic – składnia programu musi być zgodna z ISO C

Skompilowany program uruchomimy wpisując:

```
./a.out
```

Co da nam wynik:

```
Dzien dobry!student@jimp:~/JIMP1/source/lmp1$
```

Często chcemy skompilować program do pliku wykonywalnego o konkretnej nazwie służy do tego opcja `-o np. cc -Wall -ansi -pedantic p1.c -o p1`, da nam w wyniku plik wykonywalny o nazwie `p1`, który uruchomimy wpisując `./p1`.

Następne kroki ćwiczenia:

W kolejnej części ćwiczenia zapoznajcie się Państwo z pozostałymi programami. Ich dokładniejszy opis znajdziecie na platformie wikidyd: [http://wikidyd.iem.pw.edu.pl/LMP\(2f\)1.html](http://wikidyd.iem.pw.edu.pl/LMP(2f)1.html)

Następnie napiszcie Państwo samodzielnie jeden z programów sprawdzających zrozumienie zajęć podany przez prowadzącego (Uwaga: zadanie z pierwszych zajęć nie będzie oceniane).

Co warto poćwiczyć przed kolejnymi zajęciami na wejściówkę?

- Podstawy linuxowego command line'a
  - Podstawy obsługi edytora vim
  - Kompilacja programów
  - Podstawowa składnia i metody programowania wykorzystywane w materiałach wikidy do **lab1**
  - Podstawowa składnia i metody programowania wykorzystywane w materiałach wikidy do **lab2**
- 

Dodatkowe materiały dla zainteresowanych:

- Kursy vima  
komenda `vimtutor` w terminalu  
<https://pl.wikibooks.org/wiki/Vim/Wst%C4%99p>  
<https://www.openvim.com/>  
<https://vim-adventures.com/>
- Skrypty bashowe  
<https://linuxconfig.org/bash-scripting-tutorial-for-beginners>  
<https://www.linode.com/docs/development/bash/intro-bash-shell-scripting/>
- Chmod krok po kroku:  
<https://en.wikipedia.org/wiki/Chmod>  
<https://chmodcommand.com/>
- Ogólnodostępne cheatsheet'y  
<https://cheatography.com/davechild/cheat-sheets/linux-command-line/>  
<https://vim.rtorr.com/>