Wprowadzenie do systemu Linux

Damian Kurpiewski

Linia poleceń - terminal

Tekstowy interfejs systemu

Polecenia i odpowiedzi w formie tekstu

Działamy wewnątrz powłoki

Powłoka określa działanie i wygląd terminala

Aby wyświetlić obecną powłokę:

```
echo $SHELL
```

Historia poleceń

Każde wpisane polecenie jest zapisywane w **historii**

Strzałki **góra** i **dół** pozwalają przeglądać historię

Możemy edytować wcześniej wpisane polecenia i uruchomić je ponownie

Dodanie użytkownika

`sudo adduser <nazwa użytkownika>`

Np.:

`sudo adduser blackbat`

<> - określają parametr polecenia, który należy podać

sudo – uruchamia polecenie jako administrator (**root**)

Dodanie użytkownika c.d.

- Przy pierwszym wykorzystaniu polecenia **sudo** musimy podać hasło administratora

- **Uwaga**: w systemie Linux nie widać, jak wprowadzamy hasło (wskaźnik się nie przesuwa)

- Następnie podajemy hasło naszego nowego użytkownika
- I inne jego dane (jeżeli chcemy)
- Jeżeli nie chcemy podawać dodatkowych informacji, wciskamy `Enter`

Nawigacja po systemie

- Aby poznać ścieżkę, w której się znajdujemy:

pwd

pwd – ***P**rint **W**orking **D**irectory*

- Wiele poleceń zależy od tego, gdzie aktualnie jesteśmy

Nawigacja po systemie

Możemy wyświetlić zawartość obecnego katalogu:

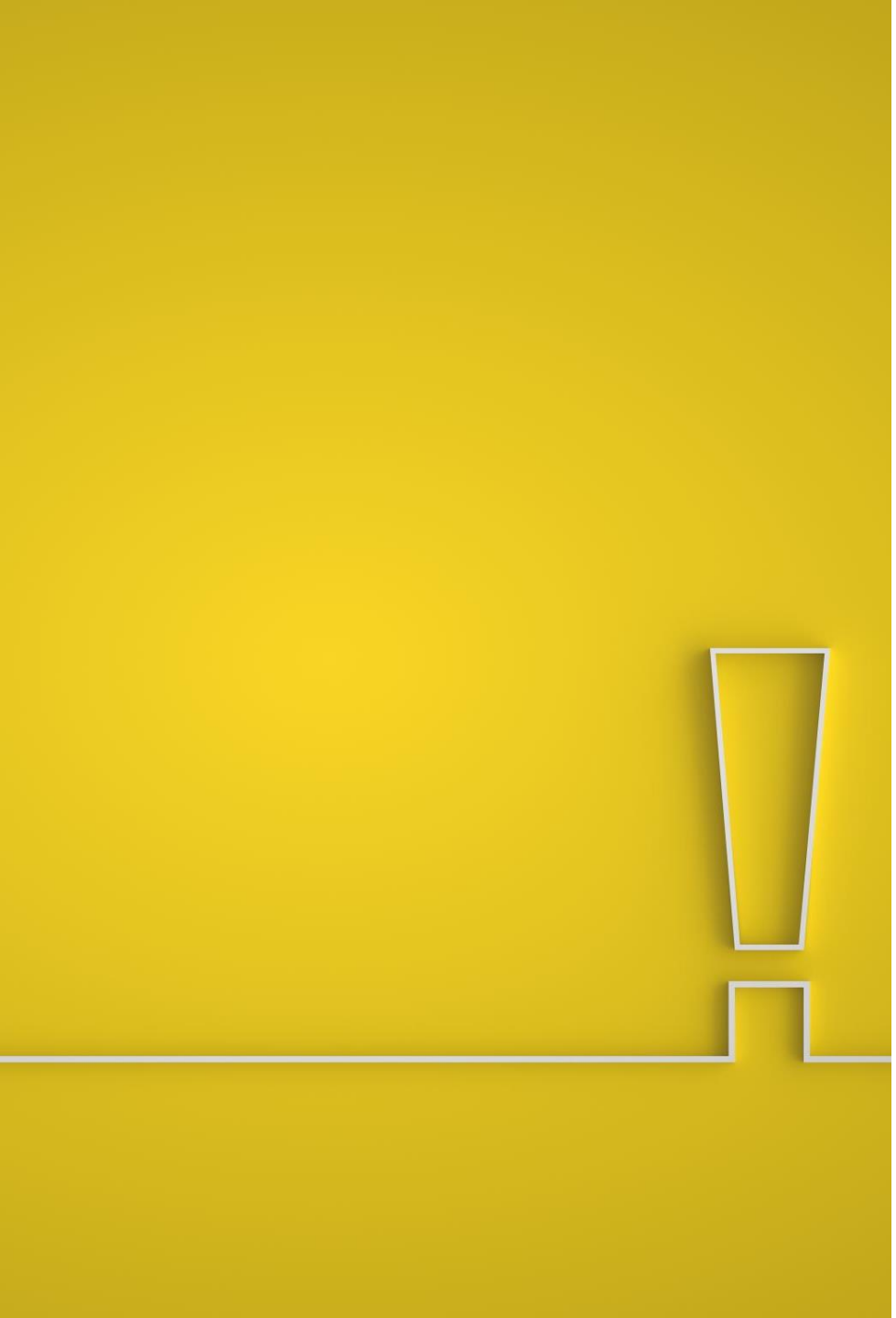
`ls`

`ls` – *list*

Polecenie możemy uruchomić z argumentami:

`ls [opcje] [lokacja]`

([`nazwa`] określa elementy opcjonalne)



`ls` - przykłady

```
ls -l
ls /etc
ls -l /etc
```

- Wszystkie opcje zaczynają się od myślnika (lub dwóch)
- Każde polecenie ma inne dostępne opcje

Ścieżki

- Mamy ścieżki **absolutne** i **względne**
- Możemy używać obu by odnosić się do plików

Ścieżki absolutne

- Korzeń systemu (root directory):

  /

  (*slash*)

- Od tego wszystko się zaczyna

- Ścieżki absolutne zawsze zaczynają się od korzenia:

  `/usr/bin`

Ścieżki względne

- Odnoszą się do miejsca, w którym się aktualnie znajdujemy
- Nie zaczynają się od /

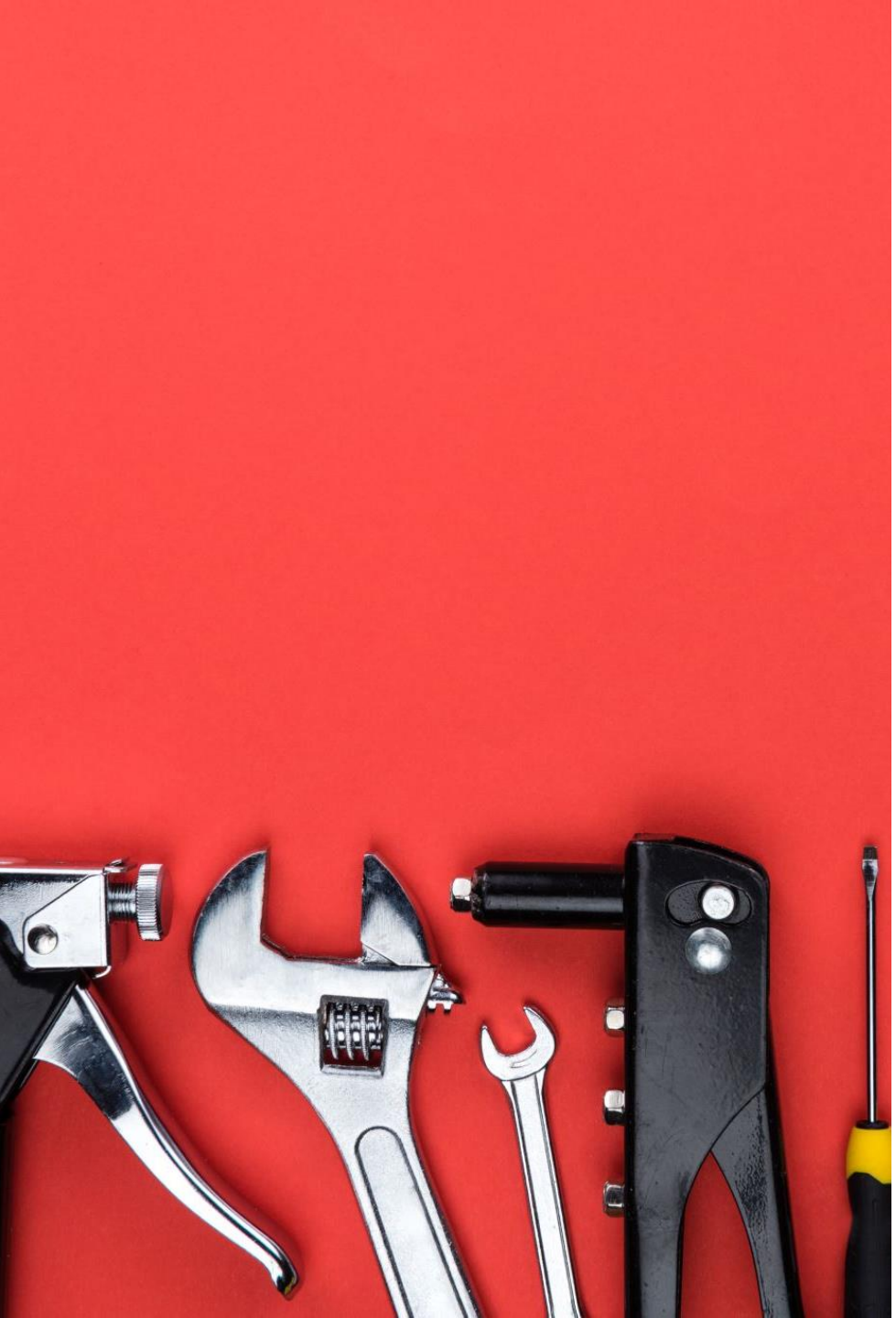
Przykład

Ścieżka względna:

```
ls Documents
```

Ścieżka absolutna:

```
ls /home/blackbat/Documents
```



Więcej o ścieżkach

- ~ (*tylda*) – skrót do naszego katalogu domowego

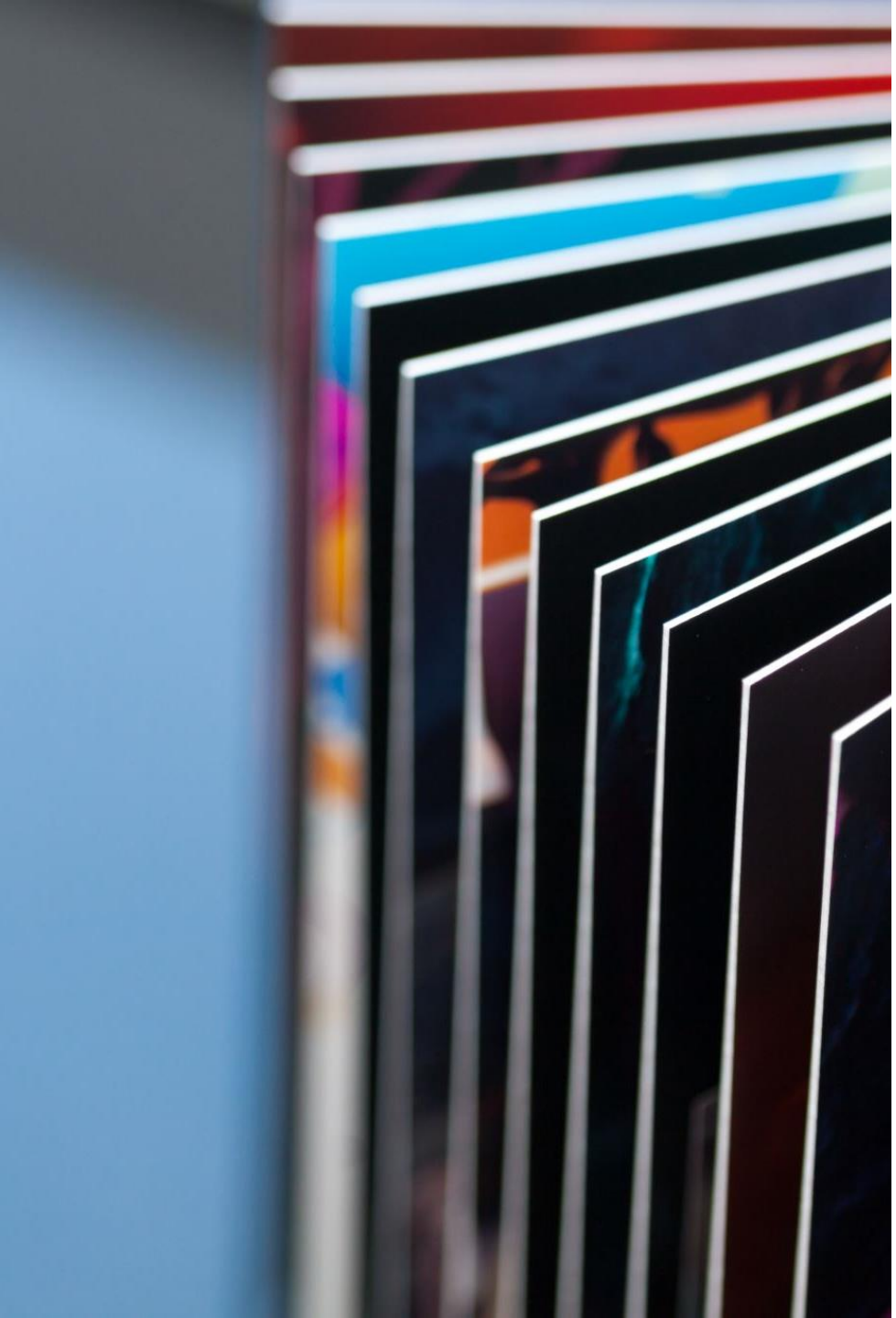
  `~/Documents`

- `.` (*kropka*) – odniesienie do obecnego katalogu

  `./Documents`

- `..` (*dwukropek*) – odniesienie do katalogu rodzica (obejmującego)

  `../../usr`



Poruszanie się po systemie

- Aby przejść do innego katalogu:

`cd <ścieżka>`

- Przykłady:

```
cd Documents
cd /
cd ~/
```

Wskazówka

- Gdy zaczniesz coś wpisywać, kliknij przycisk **TAB**
- TAB uzupełnia wpisywaną komendę, ścieżkę…
- Jeżeli jest więcej niż jedna możliwość, dwukrotne wciśnięcie TAB wyświetli ich listę

Ćwiczenia

Jakie pliki i foldery znajdują się w katalogu głównym (`root`)?

Jakie pliki i foldery znajdują się w Twoim katalogu domowym?

Co mieści się pod ścieżką `/bin`?

O plikach słów kilka

- Wszystko jest plikiem: katalogi, klawiatura, monitor…
- Linux jest systemem bez rozszerzeń: nazwa pliku (np. plik.exe, plik.txt) nie definiuje jego typu
- Możemy poznać typ pliku używając polecenia:

```
file <ścieżka>
```

Pliki c.d.

- Wielkość liter ma znaczenie: `PLIK.txt` to nie `plik.txt`

- Spacje w nazwach są dozwolone, ale bywają kłopotliwe

`ls Nowy Folder` – błąd

`ls 'Nowy Folder'` – ok

`ls Nowy\ Folder` – też ok

- Symbol \ unieważnia specjalne znaczenie (działanie) kolejnego znaku

Ukryte pliki i katalogi

- Wszystkie pliki, których nazwa zaczyna się od kropki są traktowane jako ukryte

```
.ukryty_plik
```

- Aby wyświetlić ukryte pliki, możemy użyć opcji `-a` do polecenia `ls`

```
ls -a
```

Ćwiczenia

Jakie ukryte pliki znajdują się w Twoim katalogu domowym?

Jaki mają typ?

Manual

Polecenia powłoki oferują najróżniejsze możliwości

Ciężko je jednak wszystkie spamiętać

Z pomocą przychodzi **`manual`**

Możemy go traktować jako podręcznik użytkownika

Opisuje poszczególne polecenia i ich opcje

Jak poznać polecenie

- Aby wyświetlić informacje o poleceniu, użyjemy:

```
man <polecenie do wyszukania>
```

- Np.:

```
man ls
```

- Podręcznik zamykamy wciskając **q** (*quit*)

Wyszukiwanie

- Aby wyszukać informację w podręczniku:

```
man -k <hasło do wyszukania>
```

- Np.:

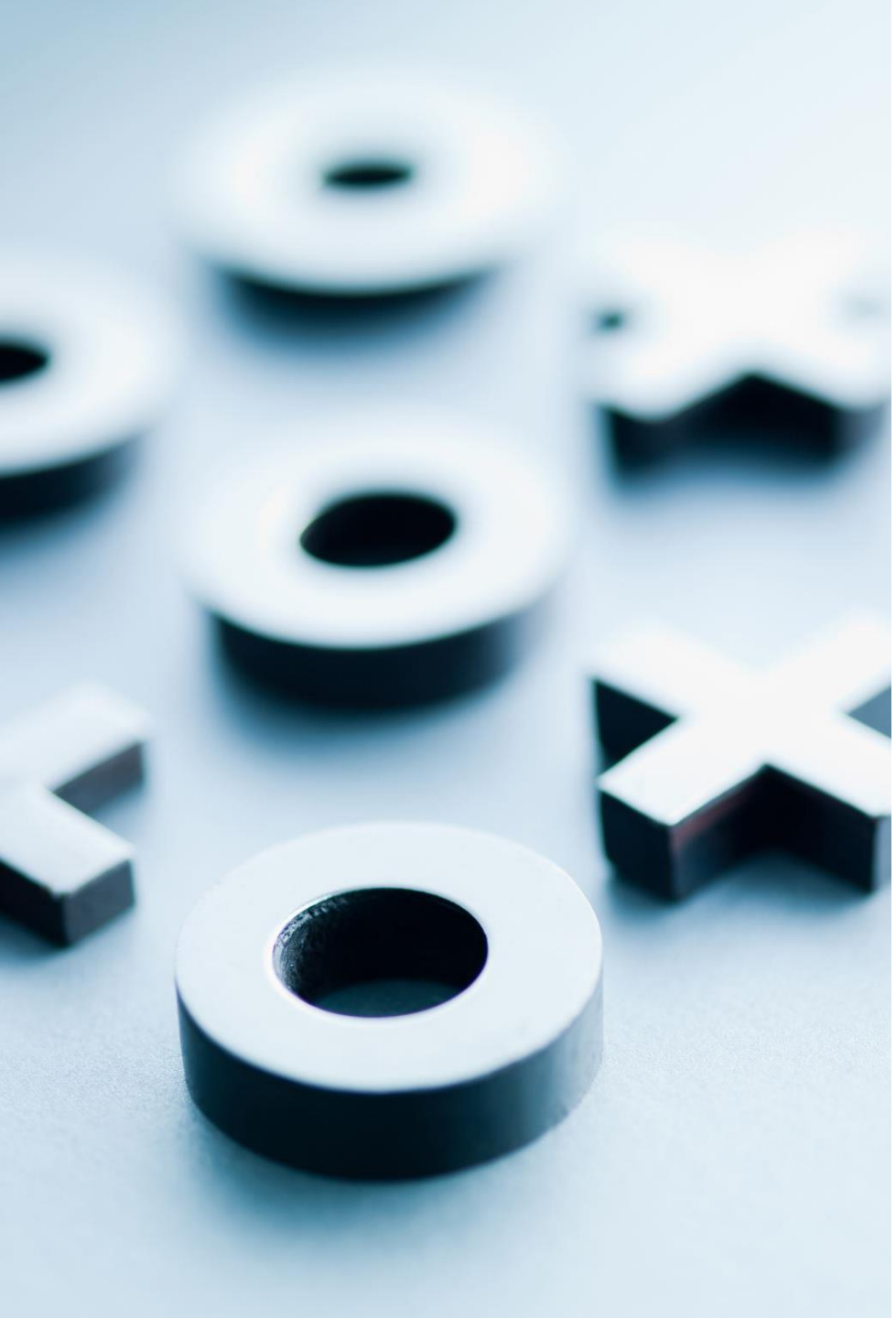
```
man -k list
```

Wyszukiwanie c.d.

- Możemy także przeszukać konkretną stronę podręcznika
- Mając otwarty manual wpisujemy / i hasło, które chcemy wyszukać
- Np.:

```
/time
```

- Jeżeli znaleziono wiele wystąpień podanego hasła, możemy je przeglądać wciskając **n**



Opcje

- Większość opcji do polecenia ma swoją krótką i długą wersję
- Np.:

  `-a` i **`--all`**

- Krótkie zaczynają się od **pojedynczego myślnika**, długie od **dwóch myślników**
- Krótkie polecenia możemy łączyć ze sobą
- Np.:

  `-l` i **`-a`** możemy połączyć w **`-la`** (albo **`-al`**)

Ćwiczenia

- Jakie opcje są dostępne dla polecenia `ls`?
- Jak posortować wynik polecenia `ls`?

Manipulacja plikami

- Utworzenie katalogu:

`mkdir [opcje] <katalog>`

- Np.:

```
mkdir Tajne
```

- Jakie opcje ma polecenie `mkdir`?

Manipulacja plikami c.d.

- Usuwanie katalogu:

```
rmdir [opcje] <katalog>
```

- Np.:

```
rmdir Tajny
```

Manipulacja plikami c.d.

- Tworzenie pustego pliku:

```
touch [opcje] <nazwa pliku>
```

- Np.:

```
touch notatki.txt
```

Manipulacja plikami c.d.

- Kopiowanie plików:

  `cp [opcje] <źródło> <miejsce docelowe>`

- Np.:

  ```
  cp notatki.txt zakupy.txt
  ```

- Aby skopiować katalog użyjemy opcji **`-r`** (*recursive*)

Manipulacja plikami c.d.

- Przenoszenie plików i katalogów:

```
mv [opcje] <źródło> <miejsce docelowe>
```

- Np.:

```
mv notatki.txt Documents/notatki.txt
```

- **Wskazówka**: polecenia **mv** można użyć także do zmiany nazwy pliku

Manipulacja plikami c.d.

- Usuwanie plików:

```
rm [opcje] <plik>
```

- Np.:

```
rm notatki.txt
```

- **Uwaga**: operacji nie można cofnąć

Popularne opcje:

- `-f` – wymusza usunięcie (bez potwierdzania)
- `-r` – rekurencyjnie usuwa katalog i jego zawartość

Ćwiczenia

1. Utwórz katalog w swoim katalogu domowym
2. Wewnątrz dodaj kilka plików i katalogów
3. Zmień nazwy tych plików
4. Usuń katalog, który ma wewnątrz jakieś pliki
5. Skopiuj plik ze swojego nowego katalogu do katalogu domowego
6. Przenieś ten plik do katalogu zmieniając jego nazwę

Vi – edytor tekstów

- **Vi** jest edytorem tekstów działającym z poziomu linii poleceń
- Do jego obsługi używamy wyłącznie klawiatury
- Żeby zapisać plik, zamknąć itp. używamy odpowiednich poleceń
- **Vi** jest bardzo popularnym i rozbudowanym narzędziem

Vi

- Edytujemy plik:

`vi <nazwa pliku>`

- Jeżeli plik nie istnieje, to zostanie utworzony

- Zaczynamy w trybie edycji
- Aby przejść do trybu wprowadzania używamy **`i`** (*insert*)
- W tym trybie możemy wpisywać tekst jak w standardowym edytorze

Vi - strzałki

- Domyślnie edytor `Vi` używa klawiszy `jklh` zamiast strzałek

- Aby włączyć korzystanie ze strzałek w edytorze:
  - Przechodzimy do trybu edycji (ESC)
  - Wpisujemy polecenie:

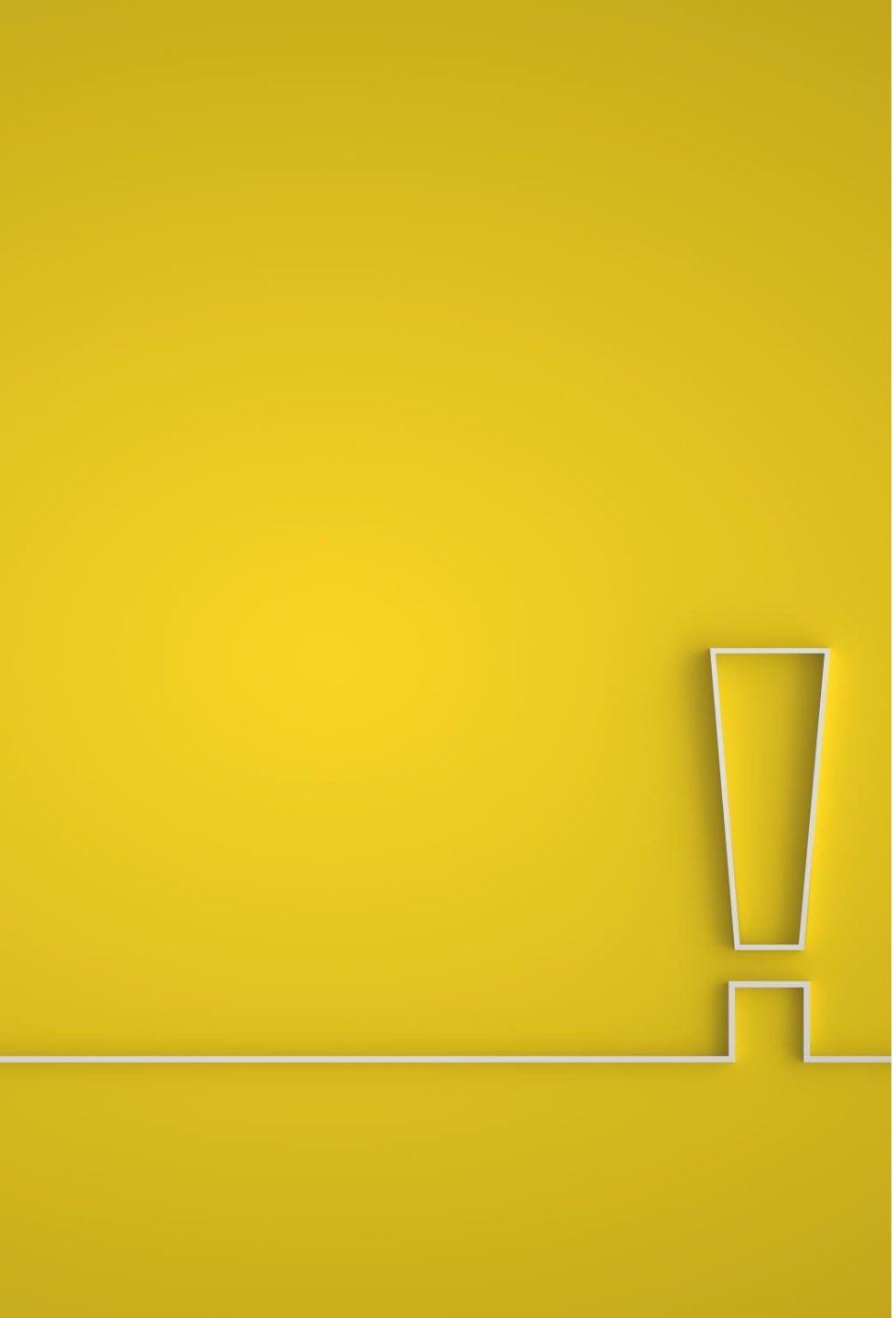
```
:set nocompatible
```

- Aby włączyć tę opcję na stałe, należy dodać wpis `set nocompatible` do pliku `~/.exrc`

`Vi` - zapisywanie

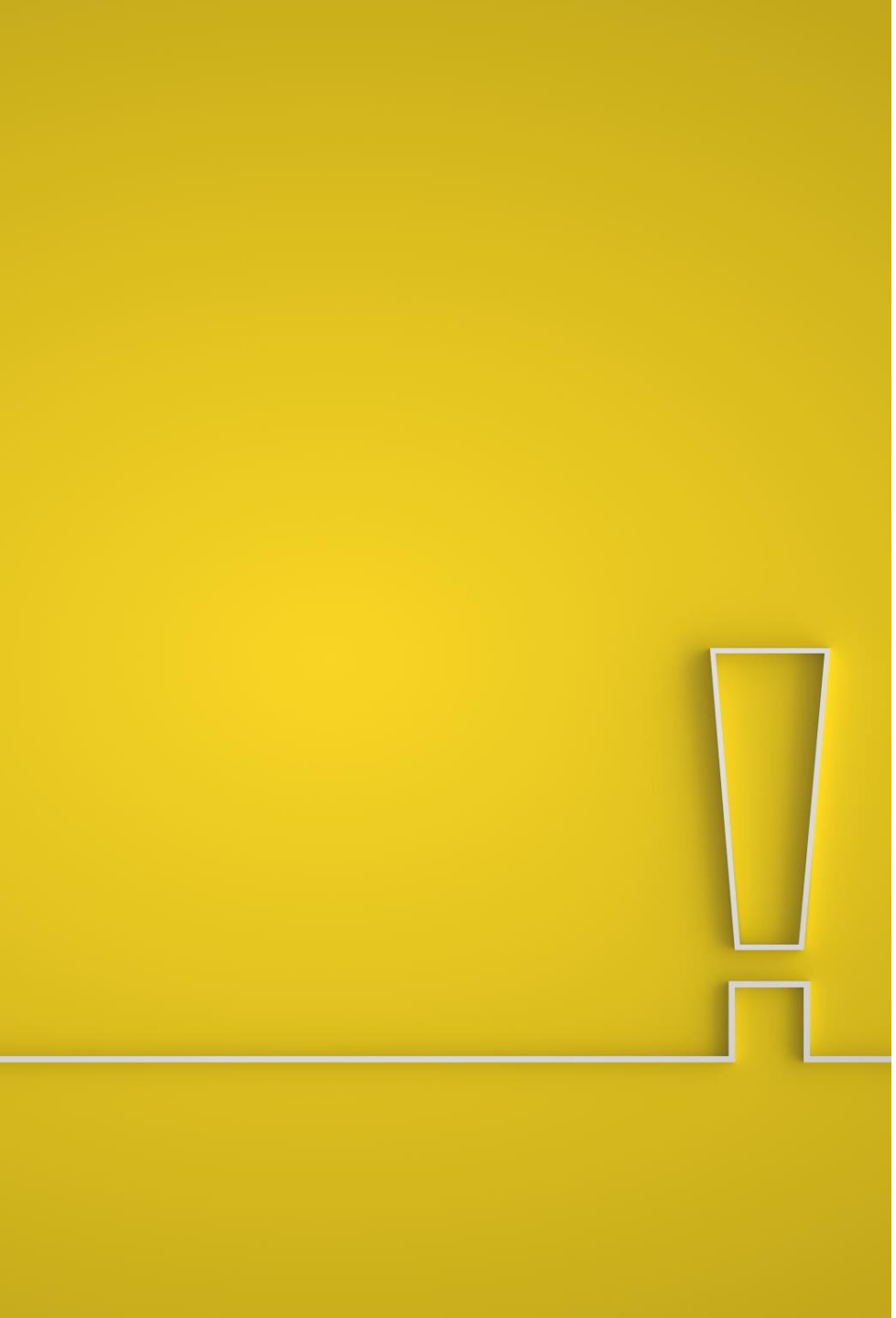
- Najpierw wychodzimy z trybu wprowadzania wciskając ESC

- Wpisujemy jedno z poniższych poleceń, aby zakończyć pracę:
    - **`ZZ`** – zapisujemy i wychodzimy z edytora
    - **`:q!`** – wychodzimy bez zapisywania
    - **`:w`** – zapisujemy, ale nie wychodzimy
    - **`:wq`** – zapisujemy i wychodzimy



`Vi` – przemieszczanie się po pliku (w trybie edycji)

- **^** - przenieś kursor na początek linii
- **$** - przenieś kursor na koniec linii
- ***<n>*G** – przenieś kursor do n-tej linii (np. 5G przenosi do 5-tej linii)
- **G** – przejdź do ostatniej linii
- **w** – przejdź do początku kolejnego wyrazu
- ***<n>*w** – przejdź do przodu o n wyrazów (np. 5w)
- **b** – przejdź do początku poprzedniego wyrazu
- ***<n>*b** – przejdź do tyłu o n wyrazów (np. 5b)



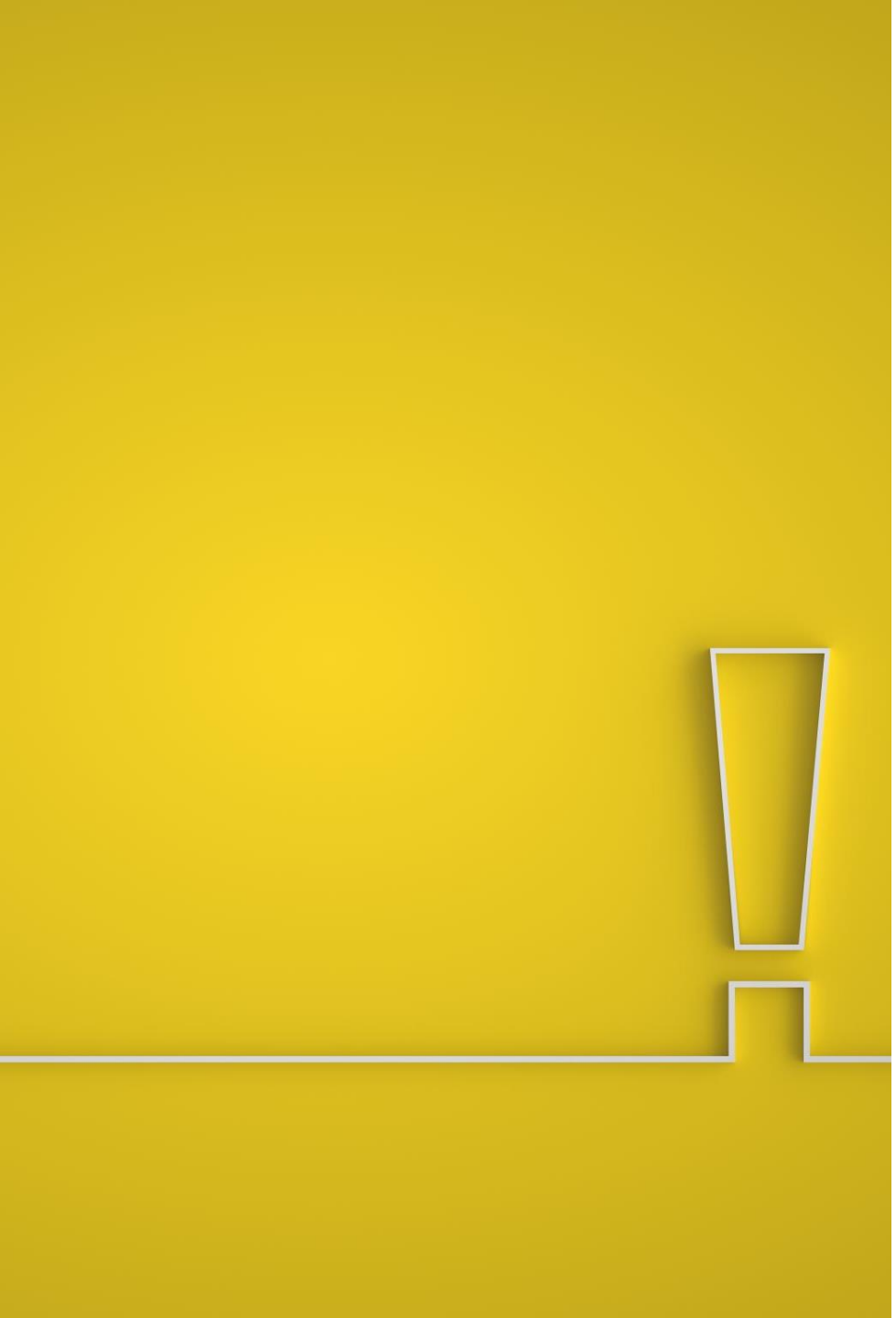
`Vi` – przemieszczanie się po pliku c.d.

- Wskazówka: w trybie edycji wpisz polecenie:

```
:set nu
```

- Sprawi to, że `Vi` będzie wyświetlać numery linii

- Ułatwi to przemieszczanie się po pliku



`Vi` – usuwanie zawartości

- W trybie edycji:
  - **x** – usuń pojedynczy znak
  - **<*n*>x** – usuń n znaków (np. 5x)
  - **dd** – usuń obecną linię
  - **d<n>** – n to polecenie przejścia; usuwa wszystko do miejsca, gdzie nas przeniesie (np. d5w usunie 5 wyrazów)

`Vi` – cofanie (w trybie edycji)

- **u** – cofnij ostatnią akcję (można wielokrotnie używać)
- **U** – cofnij wszystkie akcje w obecnej linii

`Vi` – pozostałe polecenia

- Edytor `Vi` ma wiele opcji
- W miarę używania stają się one intuicyjne
- Aby zobaczyć jak coś zrobić w `Vi`, najłatwiej jest to wygooglować
- Możemy też użyć tzw. ściągi (`Vi` cheat sheet)

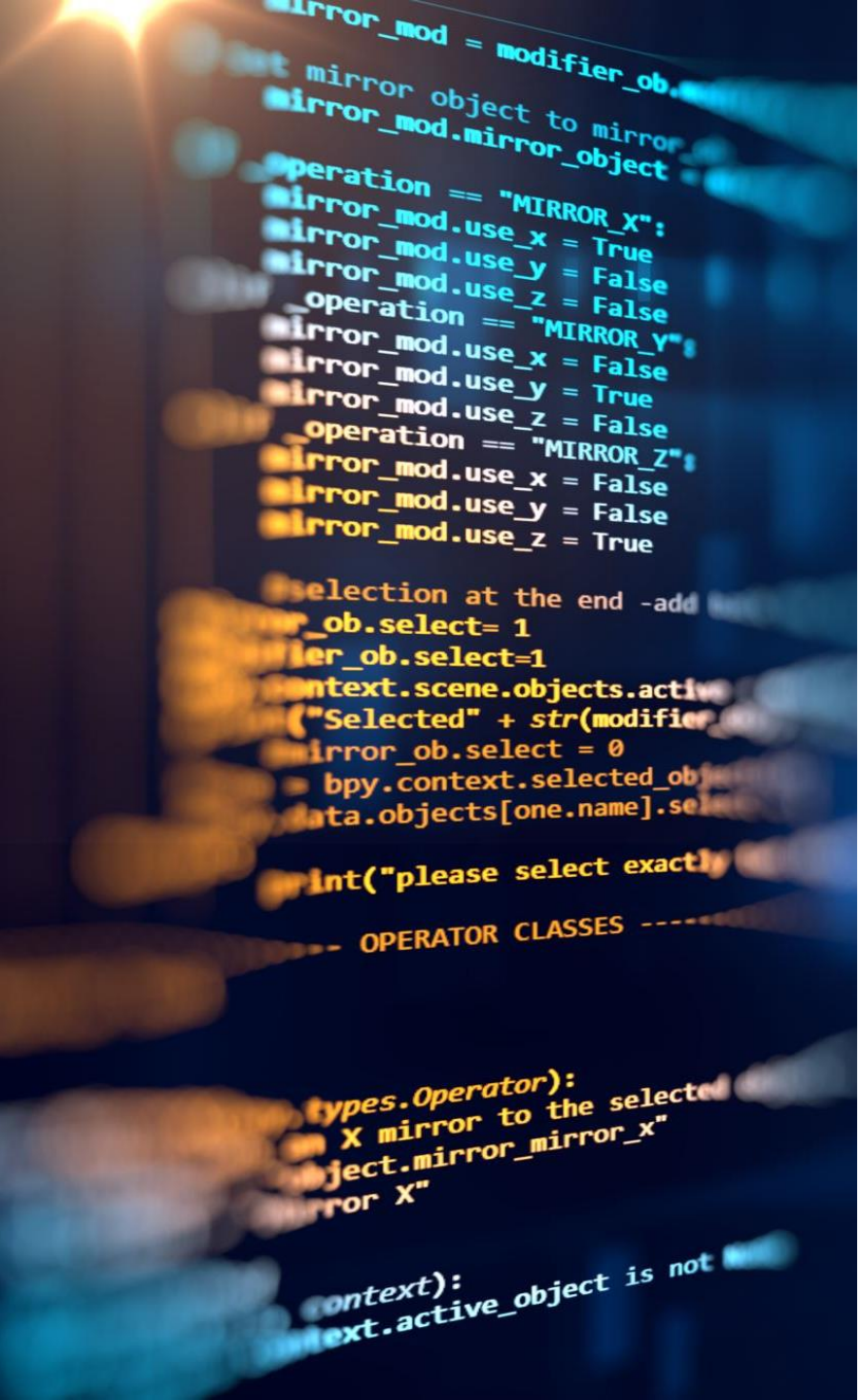
Podglądanie zawartości pliku

- Aby podejrzeć zawartość pliku bez edytowanie go, możemy użyć polecenia:

```
cat <plik>
```

- Jeżeli mamy do czynienia z większym plikiem, warto użyć:

```
less <plik>
```



Programowanie

- Z poziomu terminala możemy także pisać programy, kompilować je i uruchamiać
- Zarówno w języku C++ jak i w innych językach

Programowanie c.d.

- Utwórzmy prosty program:

```
vi hello.cpp
```

- I wypełnijmy go kodem naszego programu:

```
#include <iostream>
using namespace std;
int main() {
        cout << „Hello World!” << endl;
        return 0;
}
```

Kompilacja programu

- Po napisaniu naszego programu należy go skompilować
- W tym celu użyjemy kompilatora języka C++

```
g++ [opcje] <pliki źródłowe>
```

- Czyli:

```
g++ hello.cpp
```

Uruchamianie

- Po skończonej kompilacji (zakończonej sukcesem) utworzony zostanie plik **`a.out`**
- Jest to plik wykonywalny – nasz program (użyjmy polecenia `file`)
- Aby go uruchomić wpisujemy:

```
./a.out
```

Kompilacja programu - opcje

- Żeby nasz program zapisany został pod inną nazwą, możemy użyć opcji –o <*nazwa*>:

```
g++ -o hello hello.cpp
```

Przydatne opcje:

- `-Wall` – wyświetla wszystkie ostrzeżenia
- `-O3` – optymalizacja wynikowego programu
- `--help` – wyświetla ekran pomocy