

Podstawy Linuksa cz.2

Yakuake, Gauke

“Rozwijany” terminal

Odpowiednika w MacOS (iTerm2)

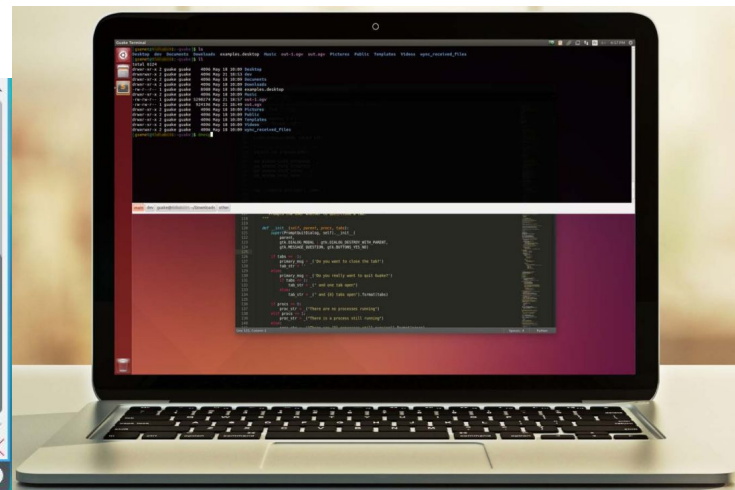
```
giurgur99@giurgur99-VirtualBox:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:af:b4:fc
        inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::640d:b998:e2f4:4003/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:153059 errors:0 dropped:0 overruns:0 frame:0
        TX packets:69635 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:216780670 (216.7 MB)  TX bytes:4853333 (4.8 MB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128  Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:1787 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1787 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:173286 (173.2 KB)  TX bytes:173286 (173.2 KB)

giurgur99@giurgur99-VirtualBox:~$
```

Shell No. 2

giurgur99 : bash - Drop-Down Terminal



OhMyZSH

<https://github.com/ohmyzsh/ohmyzsh>

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

```
Using the Oh My Zsh template file and adding it to ~/.zshrc.

Time to change your default shell to zsh:
Do you want to change your default shell to zsh? [Y/n] Y
Changing your shell to /usr/bin/zsh...
Shell successfully changed to '/usr/bin/zsh'.

oh my zsh
....is now installed!

Before you scream Oh My Zsh! look over the '.zshrc' file to select plugins, themes, and options.

• Follow us on Twitter: https://twitter.com/ohmyzsh
• Join our Discord community: https://discord.gg/ohmyzsh
• Get stickers, t-shirts, coffee mugs and more: https://shop.planetargon.com/collections/oh-my-zsh

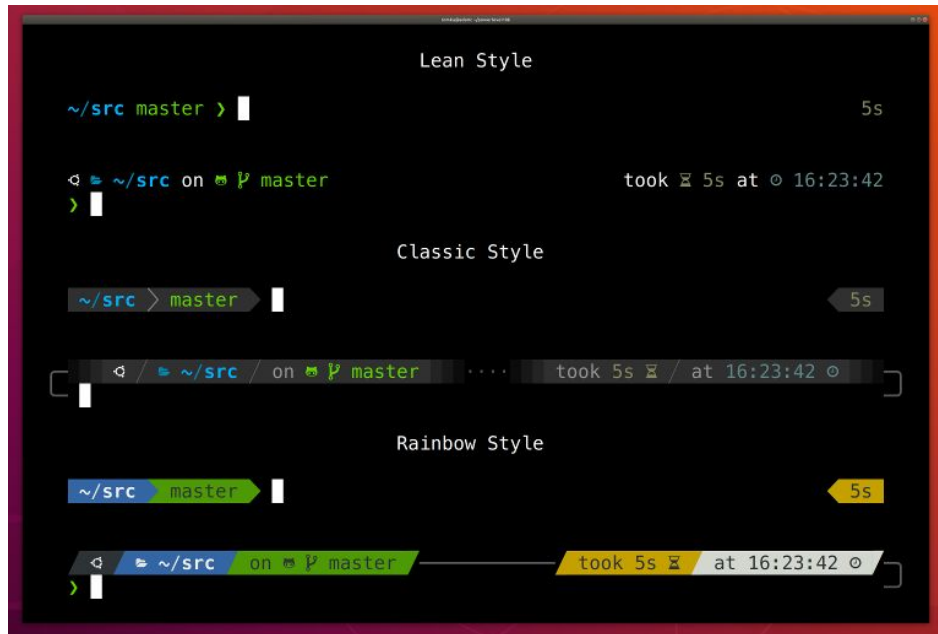
+ .oh-my-zsh git:(master)
+ .oh-my-zsh git:(master) ls -la
total 120
drwxr-xr-x 12 root root 4096 Feb 11 19:33 .
drwx----- 3 root root 4096 Feb 11 19:34 ..
drwxr-xr-x 3 root root 4096 Feb 11 19:34 cache
-rw-r--r-- 1 root root 3374 Feb 11 19:33 CODE_OF_CONDUCT.md
-rw-r--r-- 1 root root 8281 Feb 11 19:33 CONTRIBUTING.md
drwxr-xr-x 4 root root 4096 Feb 11 19:33 custom
-rw-r--r-- 1 root root 115 Feb 11 19:33 .editorconfig
drwxr-xr-x 8 root root 4096 Feb 11 19:33 .git
drwxr-xr-x 5 root root 4096 Feb 11 19:33 .github
-rw-r--r-- 1 root root 77 Feb 11 19:33 .gitignore
-rw-r--r-- 1 root root 131 Feb 11 19:33 .gitpod.Dockerfile
-rw-r--r-- 1 root root 259 Feb 11 19:33 .gitpod.yml
drwxr-xr-x 2 root root 4096 Feb 11 19:33 lib
-rw-r--r-- 1 root root 1142 Feb 11 19:33 LICENSE.txt
drwxr-xr-x 2 root root 4096 Feb 11 19:34 log
-rw-r--r-- 1 root root 5927 Feb 11 19:33 oh-my-zsh.sh
drwxr-xr-x 313 root root 12288 Feb 11 19:33 plugins
-rw-r--r-- 1 root root 13840 Feb 11 19:33 README.md
-rw-r--r-- 1 root root 1083 Feb 11 19:33 SECURITY.md
drwxr-xr-x 2 root root 4096 Feb 11 19:33 templates
drwxr-xr-x 2 root root 4096 Feb 11 19:33 themes
drwxr-xr-x 2 root root 4096 Feb 11 19:33 tools
+ .oh-my-zsh git:(master)
```

Powerlevel10k

<https://github.com/romkatv/powerlevel10k>

```
git clone --depth=1
https://github.com/romkatv/powerlevel10k.git
~/powerlevel10k
  echo 'source
~/powerlevel10k/powerlevel10k.zsh-theme'
>> ~/.zshrc
```

p10k configure



Terminal

- ~ – katalog domowy
- .. – katalog nadrzędny
- . – katalog bieżący
- | – przekierowanie wyjścia standardowego (pipe)
- > – nadpisuje dane w pliku
- >> – dołącza dane na koniec pliku

Polecenia do operowania na katalogach

- **cd** – zmienia katalog
- **cd** ~ – przechodzi do katalogu domowego
- **cd** .. – przechodzi do katalogu nadrzędnego
- **cd** /tmp – przechodzi do wskazanego katalogu
- **pwd** – pokazuje aktualny katalog
- **ls** – listuje zawartość katalogu (domyślnie bieżącego)
- **ls** -A – wyświetl wszystko (także pliki ukryte)
- **ls** -l – używa długiego formatu (long listing)
- **ls** -h – formatuje jednostki (human readable)
- **ls** -Alh /home – wszystko na raz :) (w katalogu /home)

Kopiowanie i przenoszenie

- **cp** – kopiuje plik / katalog (z -r rekursywnie, np. katalog)
- **cp** ~/plik.txt . – kopiuje plik.txt z katalogu domowego do bieżącego katalogu
- **cp** -r ../moje-pliki /tmp – kopiuje katalog „moje-pliki” (z zawartością) z katalogu nadrzędnego do /tmp
- **cp** test.txt wazny-plik.txt – kopiuje plik test.txt z bieżącego katalogu do niego samego, ale zmieniając nazwę
- **mv** – przenosi plik / katalog (z -r rekursywnie, np. katalog)
- **mv** ~/plik.txt nowa-nazwa.txt – przenosi plik.txt z katalogu domowego do bieżącego katalogu, zmieniając jednocześnie nazwę pliku

Menadżer pakietów

- Pakiet – skompilowane źródło programu
- Większość dostępna w standardowych źródłach danej dystrybucji
- Instalacja:
 - Linuksy „typu Debianowego” (np. Ubuntu) apt / apt-get
 - Linuksy „typu Red-Hatowego” rpm
 - Np. apt-get install git
 - Aktualizacja listy pakietów: apt-get update
 - Aktualizacja pakietów: apt-get upgrade

Zarządzanie pakietami

- `apt-get remove pakiet` – usuwa pakiet
- `apt-get purge pakiet` – usuwa pakiet i wszystko powiązane, włącznie z konfiguracją
- `apt-get update && apt-get dist-upgrade && apt-get autoremove`
- `/etc/apt/sources.list` zawiera listę URLi do źródeł pakietów

Zarządzanie pakietami – dodawanie repo

Najnowszy Docker:

- Pobieramy i dodajemy oficjalny klucz GPG
- `echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]`

`https://download.docker.com/linux/ubuntu`

`$(lsb_release -cs) stable" | sudo tee`

`/etc/apt/sources.list.d/docker.list > /dev/null`

Edytory tekstu

- mcedit
- nano
- vim

```
automake [----] 0 L:[ 97+14 111/8017] *(3116/216033b)= p 112 0x70
{
    &{$self->_finish} ();
}
}

sub target_hook ($$)
{
    my ($self) = @_ ;
    if (defined $self->_target_hook)
    {
        &{$self->_target_hook} ($_);
    }
}

package Automake;

require 5.005;
use strict 'vars', 'subs';
use File::Basename;
use IO::File;

my $me = basename ($0);

## ----- ##
## Constants. ##
## ----- ##

# Parameters set by configure. Not to be changed. NOTE: assign
# VERSION as string so that eg version 0.30 will print correctly.
my $VERSION = "1.5.1a";
my $PACKAGE = "automake";
my $prefix = "/usr/local";

1:Help 2:Save 3:Mark 4:Replac 5:Copy 6:Move 7:Search 8>Delete 9:PullDn 10:Quit
```

Przydatne narzędzia

- **less / more / cat** – czytają pliki
- **tail** – także czyta plik, ale tylko koniec (domyślnie 10 linijek)
- **mc** – Midnight Commander – menadżer plików
- **grep** – wyszukuje w tekście linie zawierające ciąg znaków pasujący do podanego wyrażenia regularnego
- Np. `grep 'Ala' nazwa_pliku`

Przydatne narzędzia - sieć

- **ifconfig** – konfiguracja sieci
- **curl** – zapytania HTTP
- **wget** – pobieranie
- **ping** – wysyła 'pinga' do hosta (protokół ICMP)
- **nping**
- **awk** – język wyszukiwania i operowania na wzorcach
- **sed** – edytor strumieni tekstu: filtrowanie i modyfikowanie
- **sed** 's/unix/linux/' plik.txt
- **nping**

Polecenia do operowania na katalogach i plikach

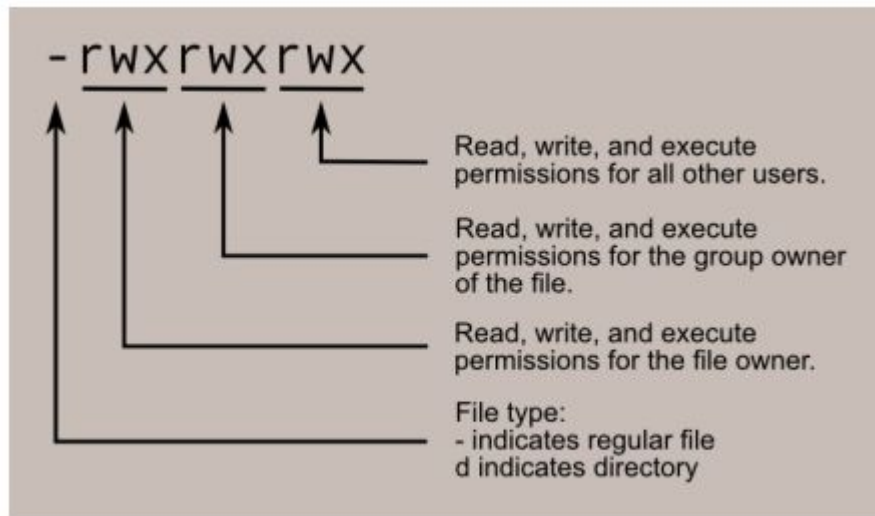
- **mkdir** – tworzy katalog
- **touch** – tworzy plik
- **rm** – usuwa plik(i) / katalog(i)
- **rm -r** – usuwa rekursywnie (łącznie z katalogami)
- **rm -rf** – usuwa rekursywnie i nie pyta
- Uwaga na **rm -rf /**

sudo

- **super user do**
- Z wielką władzą wiąże się wielka odpowiedzialność
- Operacje wykonywane z uprawnieniami innego użytkownika, zwykle administratora („roota”)
- np. `sudo ls`

Uprawnienia

- r – odczyt (read)
- w – zapis (write)
- x – wykonanie (execution)
- u – właściciel (user)
- g – grupa (group)
- o – pozostali (other)
- chmod g+w – doda grupie prawo zapisu
- chmod o-rx – zabierze uprawnienie odczytu i wykonania dla pozostałych



źródło: <https://linuxcommand.org/>

Uprawnienia liczbowo

- Potęgi dwójki:
- $r = 4$
- $w = 2$
- $x = 1$
- `chmod 700` – da wszystkie uprawnienia właścicielowi i zabierze wszystkie grupie i pozostałym

	u g o		
	754		
access	r w x	r w x	r w x
binary	4 2 1	4 2 1	4 2 1
enabled	1 1 1	1 0 1	1 0 0
result	4 2 1	4 0 1	4 0 0
total	7	5	4

Uprawnienia – po co? Kiedy?

- Zasada najmniejszej odpowiedzialności
- Nadajemy dostępy:
- Nowym użytkownikom (także tym po ssh)
- Serwisom

Dowiązania symboliczne

Skróty

- In -s plik_dowiązany skrót

Przydatne narzędzia – archiwizacja

- tar / gzip / zip / gunzip – archiwizacja i/lub kompresowanie
- Spakowanie
- .tar.gz: tar -cvf wynik.tar.gz plik_lub_katalog
- .zip: zip wynik.zip plik_lub_katalog
- Rozpakowanie
- .tar.gz: tar -xzf nazwa_pliku.tar.gz
- .gz: gunzip nazwa_pliku.gz
- .zip: unzip nazwa_pliku.zip

Metryki systemowe

- top – podstawowe źródło metryk
- htop
- lscpu – informacje o procesorze
- free -h – informacje o pamięci

<https://goinbigdata.com/understanding-output-of-free-in-ubuntu-16-04/>

- df – informacje o zamontowanych systemach plików
- du – informacje o użyciu dysku
- ss -tulpn – wyświetla połączenia sieciowe, statystyki, etc.
- netstat -tulpn – j/w (starsze)

Metryki systemowe – load average

- Np. z uptime:

23:16:49 up 10:49, 5 user, load average: 1.00, 0.40, 3.35

- Dla systemu 1-procesorowego:
- średnio 100% użycia w ostatniej minucie
- średnio 40% zużycia (60% „idle”) w ostatnich 5 minutach
- średnio 235% nadmiarowego obciążenia („overload”) w ostatnich 15 minutach; średnio 2.35 procesu czekało na swój przydział CPU

Praca z service unit

systemd

- Menedżer systemu i usług dla Linuksa („init system”)
- Używany w RHEL, CentOS, Fedora, Ubuntu, Debian, ...
- Kontrowersje: zerwanie z filozofią Uniksa, czyli jedna aplikacja spełnia dokładnie jedną funkcję

Zarządza „unitami”:

- .service – serwis / aplikacja
- .socket – socket (sieć)
- .target – służy synchronizacji poszczególnych unitów

Service unit

Składa się z 3 sekcji:

- [Unit]: informacje nie związane specyficznie z danym unitem, np. opis
- [Service]: informacje specyficzne dla danego unitu
- [Install]: informacje dotyczące instalacji unitu

Service unit – [Unit]

Opis zachowania, zależności

[Unit]

Description=The Apache HTTP Server

After=network.target httpd-init.service

Requires=avahi-daemon.socket

Zależności:

- Requires – twarda
- Wants – miękka

Service unit – [Service]

- Co ma wydarzyć się po wystartowaniu serwisu

[Service]

ExecStart=/usr/sbin/NetworkManager --no-daemon

TimeoutSec=300

- ExecStartPre / ExecStartPost – komendy do wywołania przed i po
- ExecStop / ExecReload – komendy do wywołania na zatrzymanie / przeładowanie serwisu

Service unit – [Install]

- Co ma wydarzyć się przy instalacji serwisu

[Install]

WantedBy=multi-user.target

- WantedBy / RequiredBy – ponownie miękkie i twarde zależności
- multi-user.target – potrzebny gdy system osiągnie 2 runlevel
- `ls -Alh /lib/systemd/system/runlevel*`

Service unit – instalacja

- `/usr/lib/systemd/system` – unity systemowe
- `/etc/systemd/system` – unity użytkownika
- Start: `systemctl start nasz_serwis.service`
- Stop: `systemctl stop nasz_serwis.service`
- Weryfikacja aktywności: `systemctl is-active nasz_serwis.service`

Service unit – autostart

- `systemctl enable nasz_serwis.service` - autostart
- `systemctl disable nasz_serwis.service` – wyłączenie autostartu

- Sekcja [Service]:

`Restart=on-failure`

`RestartSec=30s`

Service unit – journal

- Komponent systemd odpowiedzialny za logi ze wszystkich źródeł
- Tworzy plik binarny
- journalctl – podgląd logów
- journalctl -b – podgląd logów (tylko obecne uruchomienie)
- journalctl -u nasz_serwis.service – podgląd logów naszego serwisu

Service unit – przekierowanie logów

- Sekcja [Service]

StandardOutput=file:/var/log/nasz_serwis/log.log

StandardError=file:/var/log/nasz_serwis/log.err

Storage

GlusterFS, NFS, rook, Ceph?

GLUSTER

[Events](#)[Planet Gluster](#)[Documentation](#)[Community](#)[Install](#)

Scalable data filesystem

Gluster is a free and open source
software scalable network filesystem.

[Install Gluster](#)

Dobre praktyki (storage)

- Oddzielna partycja na dane
- Oddzielna partycja na logi
- W przypadku baz danych o ile to możliwe wybieramy dysk SSD

| Git

Git

- git init – inicjuje nowe repozytorium
- git clone – klonuje („pobiera”) repozytorium
- git add – dodaje pliki do zmiany
- git commit – „komituje” zmiany
- git push – wypycha zmiany do zdalnego repozytorium (origina)
- git pull – pobiera zmiany ze zdalnego repozytorium (origina)

LazyGit

<https://github.com/jesseduffield/lazygit#ubuntu>

```
sudo add-apt-repository ppa:lazygit-team/release
```

```
sudo apt-get update
```

```
sudo apt-get install lazygit
```

```
--Status
1040 don3-materialy + master

--Files - Submodules
# bash/example_progres_bar.sh

--Staged Changes
diff --git a/bash/example_progres_bar.sh b/bash/example_progres_bar.sh
index 989f9cd..4c64c7b 100755
+++ b/bash/example_progres_bar.sh
@@ -5,18 +5,20 @@
 . progress_bar.sh

PROJECTS_TO_PULL=(
-  "sample_proj_1"
-  "sample_isa_2"
-  "sample_isa_4"
-  "sample_isa_3"
+  "sample_proj_1"
+  "sample_isa_2"
+  "sample_isa_4"
+  "sample_isa_3"
+  "sample_isa_1"
+  "sample_isa_5"
+  "sample_isa_6"
)

--Local Branches - Remotes - Tags
* master 1040

--Commits - Reflog
3752e1ec Advanced bash examples
403a957b Update README.md
74e6b3db Add files via upload
886dc569 Update README.md
88d728d4 Update README.md
3907acc1 Update README.md
4d3a48e1 Add files via upload
ee3f9fab Update README.md
45d07fab Create README.md
886dc87f Initial commit

--Command Log
You can hide/focus this panel by pressing 'O' or hide it permanently in your config with 'gui.showCommandLog: false'

Random Tip: You can append your staged changes to an older commit by pressing 'A' on that commit
Stage all files
git add -A

--Schwek
1 of 10

--
1 of 1
```

Skrypty

Pierwszy skrypt

```
#!/bin/bash
```

```
echo "Hello World"
```

- Warto pamiętać o dodaniu praw do wykonywania (+x).
- `bash nazwa_skryptu.sh`
- `./nazwa_skryptu.sh`

Sumowanie

```
#!/bin/bash
```

```
# Dodaj liczby
```

```
((sum=25+35))
```

```
# Wypisz wynik
```

```
echo $sum
```

Operacje arytmetyczne

Subtraction

```
[me@linux ~]$ expr 1 - 1
```

```
0
```

Addition

```
[me@linux ~]$ expr 1 + 1
```

```
2
```

Assign result to a variable

```
[me@linux ~]$ myvar=$(expr 1 + 1)
```

```
[me@linux ~]$ echo $myvar
```

```
2
```

Addition with a variable

```
[me@linux ~]$ expr $myvar + 1
```

```
3
```

Division

```
[me@linux ~]$ expr $myvar / 3
```

```
0
```

Multiplication

```
[me@linux ~]$ expr $myvar \* 3
```

```
6
```

Pętle for

```
#!/bin/bash  
for (( counter=10; counter>0; counter-- ))  
do  
echo -n "$counter "  
done  
printf "\n"
```

Pętlą while

```
#!/bin/bash
valid=true
count=1
while [ $valid ]
do
    echo $count
    if [ $count -eq 5 ];
    then
        break
    fi
    ((count++))
done
```

Instrukcja if / else

```
#!/bin/bash  
n=10  
if [ $n -lt 10 ];  
then  
echo "To liczba jednocyfrowa"  
elif [ $n -lt 100 ]; then  
echo "To liczba dwucyfrowa"  
else  
echo "To liczba trzy lub więcej cyfrowa"  
fi
```

Wprowadzanie danych przez użytkownika

```
#!/bin/bash
```

```
echo "Podaj imię"
```

```
read name
```

```
echo "Czesc $name!"
```

Argumenty z linii poleceń

```
#!/bin/bash
```

```
echo "Liczba argumentów : $#"
```

```
echo "pierwszy argument = $1"
```

```
echo "drugi argument = $2"
```

Funkcje

```
#!/bin/bash
```

```
pole_prostokata() {  
  area=$(( $1 * $2 ))  
  echo "Pole torea"  
}
```

```
pole_prostokata 10 20
```